

LUERS

MICRO APPLICATION

2

AMSTRAD

**PROGRAMMES BASIC
CPC 464**

<http://www.amstradeus.com>



UN LIVRE DATA BECKER

Messages d'erreur

On a parfois l'impression d'être victime d'une malédiction quand, après avoir longuement travaillé au développement d'un programme, on l'entre dans l'ordinateur et qu'on s'aperçoit que malgré des efforts répétés il se produit toujours des erreurs.

Comme le CPC n'a pas été conçu en France, les messages d'erreur sont naturellement affichés en anglais et ils sont de plus relativement brefs pour ne pas consommer trop de place en mémoire (syntax error, type mismatch...).

Les programmeurs consciencieux apprennent tous avec le temps à connaître par cœur la signification des différents messages d'erreur, mais c'est le plus souvent un long chemin semé d'embûches et de petites fautes d'interprétation qui ralentissent le développement des programmes. Comme le CPC dispose de l'instruction ON ERROR GOTO, nous pouvons la forcer à réagir à une erreur comme nous le souhaitons et il peut même corriger certaines erreurs de lui-même: si par exemple le CPC charge des données à partir de la cassette et qu'après avoir lu toutes les données d'un fichier vous tentez de lire d'autres données, il va normalement afficher le message d'erreur: 'OF' met (= rencontré la fin du fichier). Il est alors très simple de prévoir dans une routine de traitement des erreurs que le CPC devra fermer le fichier ouvert quand se produira cette erreur. Le programme pourra ensuite suivre son cours normal.

Notre programme de messages d'erreur ne supprime toutefois pas les

erreurs qui se produisent (ceci peut mener en effet à des résultats inattendus, si la suppression des erreurs n'est pas maniée avec précaution) mais il vous aide très efficacement dans la recherche des erreurs. Le mieux est que vous essayiez vous-même! Entrez en ligne 20 uniquement NEXT n. Si vous lancez le programme, vous voyez que l'erreur est immédiatement détectée et indiquée et qu'une stratégie de correction du programme vous est proposée.

Comment procéder maintenant avec les programmes que vous avez déjà écrites? Il faut d'abord numérotter les lignes de vos programmes avec l'instruction REMUM pour que les deux programmes ne se chevauchent pas (les numéros de lignes de vos programmes doivent donc être compris entre 10 et 10000). Utilisez maintenant l'instruction 'MERGE' (reportez-vous aux explications du manuel) pour fusionner les deux programmes entre eux. Vous pouvez alors lancer votre programme et le corriger à l'aide des messages d'erreur et des propositions de rectification qui vous sont faites.

Le CPC dispose d'autres instructions que ON ERROR GOTO qui peuvent vous aider dans le domaine du dépistage des erreurs: l'interrogation de la variable 'ERR' vous fournit en effet la ligne où s'est produite l'erreur et la variable 'ERR' vous fournit le numéro du message d'erreur. Vous pouvez enfin produire des erreurs avec l'instruction ERBON.

```

10 REM Messages d'erreur
11 REM Programmes Basic Pour le CPC464
12 REM Copyright 1984 DATA BECKER &
    Rainer Lueers
13 INK 0,1:INK 1,24:INK 3,1,24
14 ON ERROR GOTO 10000
20 REM Veuillez taper votre Programme
    entre les lignes 20 et 9999
    ou bien utilisez MERGE
100 WEND
9999 STOP
10000 REM Debut de la routine de
    traitement detaille des
    erreurs
10010 MODE 1:PEN 1
10020 PRINT"Erreur en ligne"ERL"!"
10030 GOSUB 11420
10040 REM UnexPected NEXT
10050 IF ERR=1 THEN PRINT "En ligne"ERL"
    figure l'instruction":PEN 3 ELSE GOTO 1
0100
10060 PRINT "'NEXT'";:PEN 1:PRINT "sans
    qu'une instruction 'FOR'           corre
    spondante ne figure auparavant dans le Pr
    ogramme."
10070 PRINT"Il se Peut également que you
    s ayez      utilise avec NEXT le nom d'u
    ne variable correspondant à une autre bo
    ucle."
10080 END
10090 REM Syntax Error
10100 IF ERR=2 THEN PRINT "La constructi
    on de la ligne"ERL ELSE GOTO 10150
10110 PRINT "est incorrecte ou un mot a
    ete mal      ecrit. On Parle de      ":PE
    N 3:PRINT "SYNTAX ERROR";:PEN 1:PRINT "
    "
10120 PRINT
10130 END
10140 REM UnexPected RETURN

```

```

10150 IF ERR=3 THEN PRINT "En ligne"ERL"
    figure l'instruction":PEN 3:PRINT "RETUR
    N";:PEN 1:PRINT ", sans qu'une instructi
    on "; ELSE GOTO 10190
10160 PRINT "'GOSUB'":PRINT"n'a pas auPara
    vant appelle un           sous-Program
    me correspondant."
10170 END
10180 REM DATA exhausted
10190 IF ERR=4 THEN PRINT "Vous voulez f
    aire lire encore d'autres":PEN 3:PRINT"D
    ATAS";:PEN 1:ELSE GOTO 10230
10200 PRINT"Alors que toutes les donnees
    en DATAS      ont déjà été lues. Il se Peu
    t également que vous avez oublié d'utili
    ser l'ins-   truction RESTORE avant de fa
    ire lire      à nouveau les donnees en DAT
    AS."
10210 END
10220 REM ImProper argument
10230 IF ERR=5 THEN PRINT "Un Parametre
    ou un ";:PEN 3:PRINT "argument";:PEN 1:P
    RINT " a été" ELSE GOTO 10270
10240 PRINT"entre de maniere incorrecte.
    En effet, vous ne Pouvez Pas utiliser
    n'importe quelles valeurs avec les ins
    tructions SOUND ou ENV par exemple. Co
    nsultez      votre manuel!"
10250 END
10260 REM Overflow
10270 IF ERR=6 THEN PRINT "Votre ordinateur
    detecte un ";:PEN 3:PRINT "OVERFLOW"
    :PEN 1:PRINT"c'est-à-dire un dépassem
    ent." ELSE GOTO 10320
10280 PRINT"En effet, vous avez essayé e
    n      ligne"ERL":PRINT "de rePrese
    nter un":PRINT"nombre qui dépasse les Po
    ssibilités de rePrésentation du CPC464
    (nombres >      1.7E-38)."
10290 PRINT "Vous avez Peut-être aussi e

```

ssaye de convertir un nombre en un autre type de nombres, sans respecter la valeur limite."

10300 END

10310 REM Memory full

10320 IF ERR=7 THEN PRINT "Memoire ou en anglais ":";:PEN 3:PRINT "MEMORY";:PEN 1:PRINT " Pleine." ELSE GOTO 10380

10330 PRINT "Il faut le faire! Vous avez écrit un Programme Basic tellement long qu'il dépassé la capacité de mémoire de l'ordinateur. Essayez de trouver où vous";

10340 PRINT "Pourriez raccourcir votre Programme. Par exemple en supprimant des lignes de REM. Vous Pouvez d'autre Part arriver très rapidement à une surcharge de la memoire si vous dimensionnez vos variables ou tableaux trop largement ou";

10350 PRINT "si vous imbriquez trop de boucles FOR...NEXT ou trop de sous-Programmes."

10360 END

10370 REM Line does not exist

10380 IF ERR=8 THEN PRINT "Vous voulez ajouter à une ligne de Programme qui n'existe pas, en anglais ":";:PEN 3:PRINT "LINE DOES NOT EXIST";:PEN 1:PRINT "." ELSE GOTO 10420

10390 PRINT "Vous avez Peut-être mis la charrue avant les bœufs ou bien vous avez déjà renommé le Programme sans tenir compte de ce même message d'erreur."

10400 END

10410 REM Subscript out of range

10420 IF ERR=9 THEN PRINT "Le ":";:PEN 3:PRINT "SUBSCRIPT";:PEN 1:PRINT "(la grandeur de la dimension choisie d'une variable) est excessive"; ELSE GOTO 10470

10430 PRINT "en ligne":PRINT ERL:PRINT "Vous avez oublié de dimensionner une variable ou vous ne l'avez pas dimensionnée assez grande."

10450 END

10460 REM Array already dimensioned

10470 IF ERR=10 THEN PRINT "Le dimensionnement avec ":";:PEN 3:PRINT "DIM";:PEN 1:PRINT " a déjà été fait quelque part dans le Programme."; ELSE GOTO 10510

10480 PRINT "Vous ne Pouvez dimensionner à nouveau en ligne":ERL;"sans avoir au Paravant":PRINT "utilise l'instruction CL ERR."

10490 END

10500 REM Division by zero

10510 IF ERR=11 THEN PRINT "La ":";:PEN 3:PRINT "DIVISION";:PEN 1:PRINT " par 0 n'est pas permise.";:PRINT "Cherchez donc comment vous avez pu en provoquer une (Vérifiez également les variables dans la ligne fautive)."; ELSE GOTO 10540

10520 END

10530 REM Invalid direct command

10540 IF ERR=12 THEN PRINT "Vous ne Pouvez entrer cette instruction directement: ce n'est pas une ":";:PEN 3:PRINT "direct";:PRINT "command";:PEN 1:PRINT "."; ELSE GOTO 10580

10550 PRINT "(Par exemple 'A\$=INKEY\$' Pour interroger le clavier). Intégrez cette instruction dans un Programme et lancez-le avec 'RUN'."

10560 END

10570 REM Type mismatch

10580 IF ERR=13 THEN PRINT "Vous avez choisi un mauvais type de variable ":";:PRINT "<";:PEN 3:PRINT "TYPE MISMATCH";:PEN 1:PRINT ">";:PRINT "Les nombres ne peuvent être lus qu'avec des variables numériques (Variables A,B,C,...)"; ELSE

GOTO 10630
 10590 PRINT "Si vous voulez lire (avec READ ou INPUT) une variable numerique OU une chaine de caracteres, vous devez employer une variable alphanumerique (A\$, B\$, C\$...).";
 10600 PRINT "Si vous faites lire un nombre par une variable alphanumerique, vous pouvez convertir à nouveau la variable-chaine en nombre, avec l'instruction VRL(...)."
 10610 END
 10620 REM String space full
 10630 IF ERR=14 THEN PRINT "Il n'y a plus de place pour de nouvelles chaines de caracteres ("; PEN 3:PRINT "STRINGS"; PEN 1:PRINT ");"; PRINT "Pour continuer d'entrer des chaines, vous pouvez:" ELSE GOTO 10670
 10640 PRINT:PRINT "1) Raccourcir le programme Basic 2) Sauvegarder les chaines existantes sur cassette et effacer avec CLEAR la zone-mémoire des chaines."; PRINT " Vous pourrez ensuite entrer de nouvelles variables."
 10650 END
 10660 REM String too long
 10670 IF ERR=15 THEN PRINT "La chaine de caracteres ("; PEN 3:PRINT "STRING"; PEN 1:PRINT ");"; PRINT "utilisee en ligne"); ERL;" a plus de" ELSE GOTO 10710
 10680 PRINT "255 caracteres, ce qui n'est pas permis! Faites donc attention lorsque vous agrandissez des chaines avec C+ ou avec les fonctions de manipulation de chaines telles que INSTR."
 10690 END
 10700 REM String expression too complex
 10710 IF ERR=16 THEN PRINT "La chaine de caracteres en ligne "; ERL PRINT "est ";

:PEN 3:PRINT "trop complexe";:PEN 1:PRINT ".":PRINT "Ne compliquez pas artificiellement le calcul d'une chaine en utilisant trop" ELSE GOTO 10750
 10720 PRINT "d'instructions de manipulation de chaines telles que 'LEFT\$', 'RIGHT\$', 'MID\$' ou 'INSTR'.
 10730 END
 10740 REM Cannot CONTinue
 10750 IF ERR=17 THEN PEN 3:PRINT "CONT"; PEN 1:PRINT "inue ist zwar eine tolle Funktion" ELSE GOTO 10800
 10760 PRINT "die eine Fortsetzung nach 'END', 'STOP' oder Programmunterbrechung durch Drücken der <ESC>-Taste fast immer ermöglicht, aber bitte VORSICHT!"
 10770 PRINT "In diesem Fall ist seit der Programmunterbrechung zu viel geschehen, als dass der Computer sich noch daran erinnert, in welcher Zeile er mit dem Programmablauf fortfahren soll."
 10780 END
 10790 REM Unknown user function
 10800 IF ERR=18 THEN PEN 3:PRINT "FN"; PEN 1:PRINT " est certainement une instruction" ELSE GOTO 10860
 10810 PRINT "Pratique pour calculer des fonctions mathématiques qui ont été au paravant définies dans le programme avec DEF FN, mais en l'occurrence, vous avez appelle avec FN une fonction en ligne"; ERL;
 10820 PRINT "qui n'avait pas encore été définie."; PRINT:PRINT "Nous vous conseillons donc de toujours regrouper les définitions en début de"
 10830 PRINT "Programme, dans ce qu'on appelle la partie-declaration du programme.
 10840 END

10850 REM RESUME missing
 10860 IF ERR=19 THEN PEN 3:PRINT "RESUME";:PEN 1:PRINT " doit être utilisé après l'ins-"; ELSE GOTO 10910
 10870 PRINT "truction ON ERROR GOTO Pour que le Programme puisse reprendre son cours au bon endroit";:PRINT "Si vous entrez simplement RESUME, l'exécution du Programme reprendra à"
 10880 PRINT "Partir de l'endroit où l'erreur est apparue";:PRINT "Si vous entrez RESUME plus un numéro de ligne, l'exécution du Programme se poursuivra à partir de cette ligne."
 10890 END
 10900 REM Unexpected RESUME
 10910 IF ERR=20 THEN PRINT "L'instruction ";:PEN 3:PRINT "RESUME";:PEN 1:PRINT " est apparue en ligne"; ERL ELSE GOTO 10960
 10920 PRINT "sans qu'une routine de traitement des erreurs n'ait été auparavant appelée dans le Programme avec ON ERROR GOTO. Il se peut que vous avez simplement oublié de terminer le Programme avec 'END'.";
 10930 PRINT "C'est pourquoi le Programme à";:PRINT "continue et exécute la routine de traitement des erreurs."
 10940 END
 10950 REM Direct command
 10960 IF ERR=21 THEN PRINT "Lors du chargement d'un Programme sur cassette, une instruction";:PRINT "(";:PEN 3:PRINT "DIRECT COMMAND";:PEN 1:PRINT ")";:PRINT "a été trouvée, qui apparaît dans le"; ELSE GOTO 11000
 10970 PRINT "Programme sans numéro de ligne. Ceci n'est pas permis."
 10980 END
 10990 REM Operand missing

11000 IF ERR=22 THEN PRINT "Certaines instructions BASIC ne peuvent être employées sans être suivies d'un minimum d";:PEN 3:PRINT "OPERANDES";:PEN 1:PRINT ".":PRINT "Par exemple l'instruction SOUND doit"; ELSE GOTO 11030
 11010 PRINT "être suivie au moins de l'indication du numéro de canal et de la fréquence";:PRINT " SOUND 1,100 Par exemple, mais pas SOUND."
 11020 REM Line too long
 11030 IF ERR=23 THEN PRINT "Ligne de Programme trop longue";:PRINT "(";:PEN 3:PRINT "too long";:PEN 1:PRINT ")"; ELSE GOTO 11070
 11040 PRINT "Une ligne de Programme ne peut avoir plus de 255 caractères. Vous pouvez corriger cette erreur par exemple en scindant cette ligne en plusieurs petites lignes."
 11050 END
 11060 REM EOF met
 11070 IF ERR=24 THEN PEN 3:PRINT "EOF";:PEN 1:PRINT " signifie END OF FILE." ELSE GOTO 11150
 11080 PRINT "Vous avez essayé par erreur en ligne";:PRINT ERL;"de changer plus de données";:PRINT "d'un fichier qu'il n'en contient. Nous pouvons vous donner les conseils suivants pour éviter cette erreur:"
 11090 PRINT "Avant la lecture des données sur cassette, utilisez le test";:PRINT " IF EOF THEN CLOSE IN:GOTO .."
 11100 PRINT "Lorsque EOF sera alors rencontré par l'ordinateur, le fichier sera fermé et le Programme se poursuivra à un autre endroit."
 11110 PRINT "Autre possibilité: placer le

nombre d'elements stockes dans un fichier dans le fichier lui-même. Lors du chargement, faites d'abord lire ce nombre, et."

```

11120 PRINT "utilisez-le dans une boucle
FOR ... NEXT."
11130 END
11140 REM File type error
11150 IF ERR=25 THEN PRINT "Le type du f
ichier (";:PEN 3:PRINT "FILE TYPE";:PEN
1:PRINT ")":PRINT "est incorrect." ELSE G
OTO 11200
11160 PRINT "Vous ne Pouvez lire des donn
ees avec l'instruction OPENIN que sur
des fichiers ASCII. Par contre,
LOAD, RUN ou MERGE ne Permettent de charg
er que des Programmes qui ont été sauve
gardés directement avec SAVE."
11170 PRINT "Ces Programmes n'ont Pas été
en effet sauvegardés lettre à lettre
mais mot BASIC (token) après mot
BASIC. Ils sont donc en code abrégé."
11180 END
11190 REM NEXT missing
11200 IF ERR=26 THEN PRINT "Une boucle "
FOR..."::PEN 3:PRINT "NEXT";:PEN 1:PRINT
" se compose de" ELSE GOTO 11250
11210 PRINT "'FOR' et 'NEXT'. Pour la bouc
le en ligne":ERL; ","
11220 PRINT "il manque visiblement un NE
XT! Vérifiez et ajoutez-le où i
l convient."
11230 END
11240 REM File already open
11250 IF ERR=27 THEN PRINT "Un fichier q
ui a déjà été ouvert avec l'instructio
n"::PEN 3:PRINT "OPEN"::PEN 1 ELSE GOTO
11300
11260 PRINT "ne doit Pas être à nouveau
ouvert en ligne":ERL;"avec l'instr
```

uction":PRINT"OPENIN";
11270 PRINT " ou OPENOUT.":PRINT"Soit vo
us avez oublié que ce fichier a déjà et
e ouvert dans une autre Partie duProgram
me, soit vous avez relance le Program
me avant d'avoir fermé les"
11280 PRINT"fichiers (avec CLOSEIN, CLOS
EOUT) qui avaient été ouverts Par le P
rogramme (avec OPENIN, OPENOUT)."
11290 REM Unknown command
11300 IF ERR=28 THEN PRINT "Une instruct
ion en ligne":ERL ELSE GOTO 11330
11310 PRINT "est n'est Pas connue de l'o
rdinateur dans cette fonction (";:PEN
3:PRINT "UNKNOWN COMMAND";:PEN 1:PRINT
")."
11320 REM WEND missing
11330 IF ERR=29 THEN PRINT "WHILE..."::P
EN 3:PRINT "WEND";:PEN 1:PRINT " constit
ue certainement" ELSE GOTO 11380
11340 PRINT "un bon moyen de réaliser de
s Programmes structures, mais vous avez
oublié jusqu'en ligne":ERL;"de Fer
mer dans"::PRINT"votre Programme une bou
cle"
11350 PRINT "WHILE...WEND avec ";:PEN 3:
PRINT "WEND";:PEN 1:PRINT "."
11360 END
11370 REM Unexpected WEND
11380 IF ERR=30 THEN PRINT "WHILE..."::P
EN 3:PRINT "WEND";:PEN 1:PRINT " constit
ue certainement" ELSE END
11390 PRINT "un bon moyen de réaliser de
s Programmes structures, mais vous avez
oublié jusqu'en ligne":ERL;"d'ouv
rir dans"::PRINT"votre Programme une bouc
le"
11400 PRINT "WHILE...WEND avec ";:PEN 3:
PRINT "WHILE";:PEN 1:PRINT "."
11410 END
11420 PEN 3
11430 REM Les messages d'erreur analysis
11440 IF ERR=1 THEN PRINT "Unexpected NE
XT";
11450 IF ERR=2 THEN PRINT "Syntax Error"

```
11460 IF ERR=3 THEN PRINT "Unexpected RE  
TURN";  
11470 IF ERR=4 THEN PRINT "DATA exhauste  
d";  
11480 IF ERR=5 THEN PRINT "Improper argu  
ment";  
11490 IF ERR=6 THEN PRINT "Overflow";  
11500 IF ERR=7 THEN PRINT "Memory full";  
11510 IF ERR=8 THEN PRINT "Line does not  
exist";  
11520 IF ERR=9 THEN PRINT "Subscript out  
of range";  
11530 IF ERR=10 THEN PRINT "Array alread  
y dimensioned";  
11540 IF ERR=11 THEN PRINT "Division by  
zero";  
11550 IF ERR=12 THEN PRINT "Invalid dire  
ct command";  
11560 IF ERR=13 THEN PRINT "Type mismatch";  
11570 IF ERR=14 THEN PRINT "String space  
full";  
11580 IF ERR=15 THEN PRINT "String too l  
ong";  
11590 IF ERR=16 THEN PRINT "String expre  
ssion too complex";  
11600 IF ERR=17 THEN PRINT "Cannot CONTI  
nue";  
11610 IF ERR=18 THEN PRINT "Unknown user  
function";  
11620 IF ERR=19 THEN PRINT "RESUME missi  
ng";  
11630 IF ERR=20 THEN PRINT "Unexpected R  
ESUME";  
11640 IF ERR=21 THEN PRINT "Direct comm  
nd";  
11650 IF ERR=22 THEN PRINT "Operand miss  
ing";  
11660 IF ERR=23 THEN PRINT "Line too lon  
g";  
11670 IF ERR=24 THEN PRINT "EOF met";  
11680 IF ERR=25 THEN PRINT "File type er  
ror";  
11690 IF ERR=26 THEN PRINT "NEXT missing"
```

Références des variables

Nous avons déjà étudié dans le programme Mémoire 4 comment les différentes numéros de ligne et la longueur de chaque ligne sont stockées en mémoire (les adresses 368 et 369 contiennent la longueur de la première ligne, les adresses 370 et 371 contiennent le numéro de la première ligne, etc...). Le programme Mémoire 4 nous a en outre fourni un moyen simple de faire rechercher une expression d'un maximum de 6 lettres dans la mémoire de l'ordinateur. L'utilité du programme Mémoire 4 était donc d'une part de nous permettre de connaître un peu mieux la mémoire de l'ordinateur et d'autre part de pouvoir rechercher une séquence alphanumérique dans toute la mémoire.

Le programme 'références des variables' part du même principe mais va encore beaucoup plus loin: il ne recherche pas seulement des séquences de caractères ou de lettres dans la mémoire, il ordonne de plus les expressions ainsi trouvées.

Comme le CPC permet de distinguer des noms de variables comportant jusqu'à 40 caractères, il faut bien sûr faire rechercher en mémoire le nom de la variable tout entier. Notre tâche (et bien sûr avant celle du CPC) est considérablement facilitée par le fait que celui-ci stocke les premiers caractères d'un nom de variable normalement mais que le dernier caractère du nom d'une variable reçoit un code spécial correspondant au code ASCII de ce caractère + 128. Si le nom de la variable ne se compose que d'un seul caractère, c'est bien sûr le premier mais aussi dernier caractère du nom de cette variable qui reçoit ce code spécial.

(code ASCII + 128). Le stockage des variables et le listage des références des variables par notre programme tiennent également compte de l'existence de différents types de variables (alphanumériques, nombres entiers, ...).

Voyons maintenant quelles sont les possibilités exactes de notre programme de références des variables:

1) Recherche des variables du programme (le programme 'références des variables' commençant en ligne 10 000, la dernière ligne de votre programme doit être au plus 9999).

2) Si le nom de variable trouvé figure au premier rang dans l'ordre alphabétique des variables de votre programme, le programme recherche dans la mémoire programme dans quelles lignes de votre programme cette variable apparaît. C'est ainsi qu'apparaît une liste des références de vos variables qui peut vous être d'un très grand secours dans votre travail de débogage (recherche des erreurs) de vos programmes BASIC.

Vous pouvez essayer notre programme immédiatement en entrant en lignes 10 à 50 les affectations de variables suivantes:

```
10 a$="CPC464"  
20 b$="Ordinateur"  
30 CPC=164  
40 DATAKEEPER:=464  
50 PROGRAMMEBASIC%:=464
```

Vous pouvez lancer maintenant la routine de références des

variables avec RUN 10 000 et vous verrez peu de temps après sur l'écran non seulement la liste des variables utilisées mais en outre les numéros de ligne où ces variables apparaissent.

Une fois que la totalité du programme a été examinée, le CPC indique combien d'octets votre programme occupe en mémoire, sans tenir compte bien sûr du programme de références des variables qu'il vaut mieux que vous supprimiez lorsque votre programme sera complètement au point. Il vous suffira pour cela d'utiliser l'instruction: DELETE 10000-.

Un dernier conseil:

Plus votre programme est long et plus votre CPC mettra de temps à rechercher toutes les variables que vous avez utilisées. Nous vous recommandons donc de faire preuve d'un peu de patience.

Les noms des variables doivent en effet être tout d'abord triés (non seulement d'après l'ordre alphabétique mais aussi en distinguant les majuscules et les minuscules), puis le numéro de la ligne ou des lignes où elles apparaissent doit ou doivent être recherchés et il faut enfin que l'ordinateur affiche toutes ces données à l'écran.

```

10000 REM Liste et references des
variables
10010 REM Programmes Basic Pour CPC464
10020 REM Copyright 1984 DATA BECKER &
Rainer Lueers
10030 CLEAR DIM as(1000)
10040 REM Determiner la Plus Petite et
la Plus Grande variable
Possibles
10050 marque$=STRING$(40,"z")-marque1$=S
TRING$(40,"")
10060 MODE 1
10070 z=368:a$="###":b$="#####"
10080 REM Calcul de la longueur et du
numero de la ligne
10090 a=PEEK(z)+256*PEEK(z+1):b=PEEK(z+2
)+256*PEEK(z+3)
10100 REM Le calcul ne concerne que la
Partie du Programme figurant
avant la ligne 10000
10110 IF b<10000 THEN GOTO 10150
10120 GOSUB 10360
10130 PRINT "Taille du Programme:
",USING b$:z=368:PRINT " Octets":PRINT
END
10140 REM Debut de la recherche des
variables en memoire
10150 FOR z1=z+4 TO z+a:IF ((PEEK(z1)=13
OR (PEEK(z1)>1 AND PEEK(z1)<5)) AND PEEK
(z1+2)=0 THEN GOSUB 10200
10160 IF z+4>z+a THEN NEXT z1
10170 z=z+a
10180 GOTO 10090
10190 REM Quel type de variables?
    13 = nombre normal
    3 = chaine de caracteres
    2 = nombre entier avec "%"
    4 = nombre normal avec ","
10200 IF PEEK(z1)=13 THEN zz=13
10210 IF PEEK(z1)=3 THEN zz=3
10220 IF PEEK(z1)=2 THEN zz=2

```

```

10230 IF PEEK(z1)=4 THEN zz=4
10240 REM Examiner longueur du nom
de la variable
10250 IF PEEK(z1+3)>128 THEN zz$=zz$+CHR
$(PEEK(z1+3)-128): ELSE zz$=zz$+CHR$(PE
EK(z1+3)):z1=z1+1:GOTO 10250
10260 REM Quel type de variables?
    13 = nombre normal
    3 = chaine de caracteres
    2 = nombre entier avec "%"
    4 = nombre normal avec ","
10270 IF zz=13 THEN zz$=zz$
10280 IF zz=3 THEN zz$=zz$+"$"
10290 IF zz=2 THEN zz$=zz$+"%"
10300 IF zz=4 THEN zz$=zz$+""
10310 REM Les variables sont mises
dans l'ordre alphabetique
avant d'etre listees
10320 IF zz$<marque$ AND zz$>marque1$ TH
EN marque$=zz$
10330 IF flag=1 AND zz$=a$(compteur) THE
N PRINT b;
10340 zz$="":RETURN
10350 REM Ranger le nom de la variable
et le numero de la ligne ou
il apparait
10360 IF flag=1 THEN flag=0:PRINT:GOTO 1
0079 ELSE flag=1:IF marque$=STRING$(40,"z
") THEN RETURN ELSE PRINT marque$:compteur
=compteur+1:a$(compteur)=marque$:mar
que1$=marque$:marque$=STRING$(40,"z")
10370 GOTO 10079

```

Calendrier

Tous ceux qui ont l'habitude de travailler avec des ordinateurs rencontrent fréquemment des programmes de calendrier sous telle ou telle forme. Il s'agit aussi bien de programmes de "calendrier" isolés comme dans le cas présent ou bien de programmes plus important comportant un sous-programme de calendrier (par exemple les programmes d'agenda dont vous trouverez également un listing dans le présent ouvrage).

Il s'agit de toute façon toujours d'additionner les jours, les mois et les années pour calculer comme résultat final quel jour de la semaine sera la date voulue.

Le programme que nous vous proposons comprend trois parties qui résultent de l'entrée avec INPUT de l'année, du mois et du jour calendriaire voulus.

L'année entrée doit être comprise entre 1901 et 2000 et les années bissextiles sont calculées automatiquement (ligne 180).

Pour le mois, le nom du mois et le nombre de jours sont déterminés en fonction du numéro de mois entré. Si l'année est une année bissextile ($x=1$) le 29ème jour du mois de février est pris en compte (ligne 590).

La ligne 750 permet enfin le branchement du programme sur les différents jours de la semaine et l'affichage correspondant (le ... de l'année ... est/était un ...) et si vous le souhaitez vous

pouvez faire calculer le jour de la semaine d'une autre date.

```

10 REM Calendrier
20 REM Programmes Basic Pour le CPC464
30 REM CoPyright 1984 DATA BECKER &
      Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effet=2:n
  ormal=1
50 MODE 1
60 PRINT
70 PRINT
80 PRINT
90 PRINT " Determination du jour de la
  semaine           de 1901 a 2000"
100 PRINT:PRINT
110 REM Entrée de l'année et calcul
  (Pour les années bissextiles)
120 INPUT "Année ";j
130 IF j<1901 OR j>2000 THEN GOSUB 990:G
  OTO 50
140 IF INT(j/4)=j/4 THEN z=1 ELSE z=0
150 j1=2
160 IF j=1901 THEN GOTO 230
170 FOR n=1902 TO j
180 j1=j1+1
190 IF INT((n-1)/4)=(n-1)/4 THEN j1=j1+1
200 IF j1>6 THEN j1=j1-7
210 NEXT n
220 REM Entrée du mois et calcul
  (nom du mois et nombre de jours)
230 INPUT "Mois ";m
240 IF m<1 OR m>12 THEN GOSUB 990:GOTO 5
  0
250 ON m GOSUB 270,290,310,330,350,370,3
  90,410,430,450,470,490
260 GOTO 510
270 m$="Janvier"
280 RETURN
290 m$="Février"
300 RETURN
310 m$="Mars"
320 RETURN

```

```

330 m$="Avril"
340 RETURN
350 m$="Mai"
360 RETURN
370 m$="Juin"
380 RETURN
390 m$="Juillet"
400 RETURN
410 m$="Août"
420 RETURN
430 m$="Septembre"
440 RETURN
450 m$="Octobre"
460 RETURN
470 m$="Novembre"
480 RETURN
490 m$="Décembre"
500 RETURN
510 RESTORE:FOR n=1 TO m
520 READ m1
530 NEXT n
540 DATA 31,28,31,30,31,30
550 DATA 31,31,30,31,30,31
560 IF z=1 AND m=2 THEN m1=29
570 REM Entrée du jour (numéro) et calcu
  l du jour de la semaine
580 INPUT "Jour ";t
590 IF t=29 AND m=2 AND z<>1 THEN GOSUB
  990:GOTO 50
600 IF t<1 OR t>m1 THEN GOSUB 990:GOTO 5
  0
610 t1=0
620 RESTORE
630 IF m=1 THEN GOTO 690
640 FOR n=2 TO m
650 READ m1
660 t1=t1+m1
670 NEXT n
680 IF z=1 AND m>2 THEN t1=t1+1
690 t1=t1+t+j1

```

```

700 t1=t1-INT(t1/7)*7
710 IF t1=0 THEN t1=7
720 PRINT
730 PRINT "      Le";t;"m$;" de l'annee";j
740 PRINT "      est/etait un ";
750 ON t1 GOSUB 840,860,880,900,920,940,
960
760 PRINT
770 PRINT
780 INPUT " Voulez-vous calculer "
         un autre jour de la semaine
    (/N);f$
790 f$=UPPER$(f$)
800 IF f$="N" THEN END
810 RESTORE
820 CLS
830 GOTO 60
840 PRINT"Dimanche"
850 RETURN
860 PRINT"Lundi"
870 RETURN
880 PRINT"Mercredi"
890 RETURN
900 PRINT"Mercredi"
910 RETURN
920 PRINT"Jeudi"
930 RETURN
940 PRINT"Vendredi"
950 RETURN
960 PRINT"Samedi"
970 RETURN
980 END
990 PEN effet:PRINT:PRINT TAB(12) "Entre
e incorrecte!"
1000 PEN normal:GOSUB 1010:RETURN
1010 PRINT:PRINT TAB(7) "<APPuyez sur un
e touche S.V.P.>"
1020 f$=INKEY$:IF f$="" THEN GOTO 1020
1030 RETURN

```

Fichier de disques

Ce programme ne méritera son nom que si vous le tapez mot pour mot, instruction pour instruction, exactement comme il figure dans notre recueil de programmes.

Mais comme le produit final, moyennant cette précaution, sera une petite banque de données aux performances pas du tout négligeables, il est aussi possible de modifier certaines instructions PRINT, de façon par exemple à ce que PRINT"Entrez les disques" devienne PRINT"Entrez les adresses".

Vous pourriez ainsi vous constituer très facilement, en plus de votre programme de fichier de disques, un programme de fichier d'adresses qu'il faudrait bien sûr que vous sauvegardiez séparément.

Nous reconnaissons toutefois que le programme de fichier de disques pâtit malgré tout de certaines contraintes qui ne pourraient être valablement éliminées que si nous ne travaillions pas sur cassette mais sur disquette.

La première limite du programme tient bien sûr à la capacité mémoire de notre ordinateur: si nous utilisons pleinement une fiche (trois fois 40 caractères), nous occupons déjà 120 octets en mémoire. 100 fiches représentent donc 12000 octets. C'est pourquoi nous avons dû limiter à 300 le nombre maximum de fiches différentes.

Si vous êtes certain que vous n'utiliserez pas autant de caractères par fiche, vous pouvez modifier le programme de façon à renoncer le nombre limite de fiches. Vous pouvez par exemple modifier ainsi la ligne 100:

100 IF z<1 OR z>400...

Le programme "fichier de disques" vous offre les possibilités suivantes:

1. Vous pouvez constituer des fiches de trois champs (nom du disque, chanteur, numéro).
2. Vous pouvez charger des fichiers à partir d'une cassette.
3. Vous pouvez sauvegarder des fichiers sur cassette.
4. Vous pouvez faire rechercher un disque en mémoire (d'après le nom du disque, du chanteur ou le numéro). Pour cette fonction, l'écran affiche les données trouvées qui correspondent à ce que vous cherchez jusqu'à ce que l'écran soit plein. Vous appuyez alors sur ENTREE pour que la recherche se poursuive. Ceci vous permet donc de faire rechercher par exemple dans votre fichier tous les disques d'un chanteur que vous possédez.
5. Vous pouvez faire éditer la liste complète de vos disques sur l'écran ou sur une imprimante (pour l'affichage à l'écran, les données sont affichées par groupe de 5 et vous devez appuyer sur la touche ENTER pour passer à la suite).

6. Vous pouvez modifier les données entrées. Comme la routine d'entrée (point 1) vous permet déjà de corriger les données que vous avez entrées, il est peu probable que vous utiliserez cette fonction pour corriger le nom des chanteurs ou des disques figurant dans votre fichier. Nous avons en fait surtout pensé ici au cas où vous transformeriez ce programme en programme de fichier d'adresses. Il arrive en effet fréquemment qu'on ait à modifier les données d'un fichier d'adresses à la suite de changements d'adresses. Il vous suffira alors de sélectionner le point 6 du programme. Vous pourrez ensuite laisser inchangées certaines informations en appuyant simplement sur ENTREE et en modifier d'autres en entrant les nouvelles informations correspondantes.

Que vous utilisez ce programme pour vos disques ou que vous le transformiez en programme de fichier d'adresses, nous vous souhaitons en tout cas de bons moments et nous espérons qu'il vous permettra de mettre un peu d'ordre dans vos affaires.

```

10 REM Fichier de disques
20 REM Programmes Basic Pour le CPC464
30 REM Copyright 1984 DATA BECKER &
        Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effet=2:n
ormal=1
50 m=1:MODE 1
60 REM Preparatifs avant dimensionnement
        des variables
70 PAPER 1:PEN 0:PRINT "Fichier de disques":PAPER 0:PEN 1
80 PRINT
90 INPUT "Combien de disques voulez-vous
        enregistrer (maximum 300) "/z
100 IF z<1 OR z>300 THEN GOSUB 2300:GOTO
    50
110 fr=FRE(0):DIM P$(z),s$(z),n$(z)
120 PRINT:PRINT "Le seul dimensionnement
        a" ;z:PRINT "disques necessite";fr-FRE(0)
        ;"cases memoire."
130 PRINT:PRINT:PRINT "Restent donc Par
        disque au maximum":PRINT:PRINT "
        ";INT(FRE(0)/z):PRINT:PRINT "cases n
        emoire disponibles.":PRINT:PRINT:PRINT
140 INPUT "Est-ce suffisant (sinon il fa
        ut changer le dimensionnement des disque
        s a entrer)= (< N) ";f$:
150 f$=LEFT$(UPPER$(f$),1)
160 IF f$="N" THEN CLEAR:GOTO 50
170 REM Affichage du menu Principal
180 CLS
190 LOCATE 6,8
200 PAPER 1:PEN 0:PRINT "Fichier de dis
    ques":PAPER 0:PEN 1
210 LOCATE 16,19
220 PRINT "Menu"
230 LOCATE 8,13
240 PRINT "1.Entrer disques"
250 LOCATE 8,14:PRINT "2.Charger disques
    "

```

```

260 LOCATE 8,15:PRINT "3.Sauvegarder dis
    ques"
270 LOCATE 8,16:PRINT "4.Rechercher disq
    ues"
280 LOCATE 8,17:PRINT "5.Sortir disques"
290 LOCATE 8,18:PRINT "6.Modifier disque
    s"
300 LOCATE 6,21
310 INPUT "Votre choix (1/2/3/4/5/6) ";f
    $
320 IF f$="1" OR f$="6" THEN GOSUB 2300:
    GOTO 180
330 ON VAL(f$) GOSUB 370,680,940,1220,17
    50,2020
340 f$="":f1$="":f2$="":n1=0
350 GOTO 180
360 REM Sous-Programme d'entree des
        donnees
370 FOR n=m TO z
380 CLS
390 PAPER 1:PEN 0:PRINT "1.Entrer des di
    sques":PAPER 0:PEN 1
400 PRINT
410 GOSUB 2360
420 PRINT
430 f1$="":PRINT "Nom du disque"
440 INPUT P$(n):IF LEN(P$(n))>40 THEN P$(
    n)=LEFT$(P$(n),40)
450 IF P$(n)="" THEN m=n:RETURN
460 PRINT "Chanteur, Groupe:"
470 INPUT s$(n):IF LEN(s$(n))>40 THEN s$(
    n)=LEFT$(s$(n),40)
480 PRINT "Numero/Code:"
490 INPUT n$(n):IF LEN(n$(n))>40 THEN n$(
    n)=LEFT$(n$(n),40)
500 REM Premier affichage des donnees
        entrees avec Premiere
        Possibilite de modification
510 PRINT
520 PRINT "Voici encore une fois les don
    nees      entrees:"

```

```

530 PRINT "Numero d'enregistrement";n;"s
ur 1 à";z
540 PRINT
550 PAPER 1:PEN 0:PRINT "Nom du disque:"
:PAPER 0:PEN 1
560 PRINT P$(n)
570 PAPER 1:PEN 0:PRINT "Chanteur, Group
e:";PAPER 0:PEN 1
580 PRINT s$(n)
590 PAPER 1:PEN 0:PRINT "Numero/Code:";P
APER 0:PEN 1
600 PRINT n$(n)
610 PRINT
620 INPUT "Connect ( /N)";f1$
630 f1$=UPPER$(f1$)
640 IF f1$="N" THEN PRINT:GOTO 430
650 NEXT n
660 RETURN
670 REM Sous-Programme de chargement de
    fichiers sur cassette
680 CLS
690 PAPER 1:PEN 0:PRINT "2.Charger des d
isques";PAPER 0:PEN 1
700 PRINT
710 GOSUB 2360
720 PRINT
730 PRINT "En chargeant un fichier, vous
detruisez les données pouvant etre en me
moire!"
740 PRINT
750 INPUT "Voulez-vous charger des donne
es (O/ )";f1$
760 f1$=LEFT$(UPPER$(f1$),1)
770 IF f1$="O" THEN GOTO 780 ELSE RETURN
780 PRINT
790 PRINT "Nom du fichier"
800 INPUT dn$
810 dn$="!"+dn$
820 OPENIN dn$
830 INPUT #9,n

```

```

840 PRINT
850 PRINT "Le fichier a charger ";dn$
860 PRINT "se compose de";m-1;"enregis
trements"
870 FOR n=1 TO m-1
880 INPUT #9,P$(n)
892 INPUT #9,s$(n)
894 INPUT #9,n$(n)
890 PRINT n:PRINT P$(n):PRINT s$(n):PRIN
T n$(n)
900 NEXT n
910 CLOSEIN
920 RETURN
930 REM Sous-Programme de sauvegarde de
    fichiers sur cassette
940 CLS
950 PAPER 1:PEN 0:PRINT "3.Sauvegarder d
es disques";PAPER 0:PEN 1
960 PRINT
970 GOSUB 2360
980 PRINT
990 INPUT "Voulez-vous sauvegarder des d
onnees (O/ ) ";f1$
1000 f1$=UPPER$(f1$)
1010 IF f1$="O" THEN GOTO 1020 ELSE RETU
RN
1020 PRINT
1030 IF dn$<>"" THEN PRINT "Faut-il le s
auvegarder sous le nom";PRINT dn$:INPUT
    " (O/ ) ";f2$:f2$=LEFT$(UPPER$(f2$),1)
    IF f2$="O" THEN GOTO 1070
1040 PRINT "Nom du fichier ";
1050 INPUT dn$
1060 dn$="!"+dn$
1070 OPENOUT dn$
1080 PRINT #9,n
1090 PRINT
1100 PRINT "Le fichier a sauvegarder ";d
n$
1105 PRINT "se compose de";m-1;"enregis
trements"

```

```

1110 FOR n=1 TO m-1
1120 PRINT #9,P$(n)
1122 PRINT #9,s$(n)
1124 PRINT #9,n$(n)
1130 PRINT n
1140 PRINT P$(n)
1150 PRINT s$(n)
1160 PRINT n$(n)
1170 PRINT
1180 NEXT n
1190 CLOSEOUT
1200 RETURN
1210 REM Sous-Programme de recherche de
    disques, chanteurs ou numeros
1220 CLS
1230 PAPER 1: PEN 0: PRINT "4. Recherche de
    disques": PAPER 0: PEN 1
1240 PRINT
1250 GOSUB 2360
1260 PRINT
1270 REM Affichage d'un sous-menu:
    Recherche d'apres:
        1) Nom du disque
        2) Chanteur, Groupe
        3) Numero/Code
1280 PRINT "Recherche d'apres quel critere?"
1290 PRINT
1300 PRINT "1. Nom du disque"
1310 PRINT "2. Chanteur, Groupe"
1320 PRINT "3. Numero/Code"
1330 PRINT
1340 INPUT "Votre choix (1/2/3) ",f1$
1350 IF f1$<"1" OR f1$>"3" THEN RETURN
1360 ON VAL(f1$) GOSUB 1390,1510,1620
1370 f1$="" : f2$="" : n1=0: GOTO 1220
1380 REM Recherche d'apres le nom du
    disque
1390 CLS
1400 PRINT "Nom du disque"

```

```

1410 INPUT P$: IF LEN(P$)>40 THEN P$=LEFT
    $(P$,40)
1420 IF P$="" THEN RETURN
1430 FOR n=1 TO m-1
1440 IF P$=P$(n) THEN PRINT n:PRINT P$(n)
    :PRINT s$(n):PRINT n$(n):n1=n1+1
1450 IF n1=5 THEN n1=0:GOSUB 2330
1460 NEXT n
1470 PRINT
1480 GOSUB 2330
1490 RETURN
1500 REM Recherche d'apres chanteur,
    Groupe
1510 CLS
1520 PRINT "Chanteur, Groupe"
1530 INPUT s$: IF LEN(s$)>40 THEN s$=LEFT
    $(s$,40)
1540 IF s$="" THEN RETURN
1550 FOR n=1 TO m-1
1560 IF s$=s$(n) THEN PRINT n:PRINT P$(n)
    :PRINT s$(n):PRINT n$(n):n1=n1+1
1570 IF n1=5 THEN n1=0:GOSUB 2330
1580 NEXT n
1590 PRINT
1600 GOSUB 2330
1610 RETURN
1620 CLS
1630 REM Recherche d'apres numero/code
1640 PRINT "Numero/Code"
1650 INPUT n$:
1660 IF n$="" THEN RETURN
1670 FOR n=1 TO m-1
1680 IF n$=n$(n) THEN PRINT n:PRINT P$(n)
    :PRINT s$(n):PRINT n$(n):n1=n1+1
1690 IF n1=5 THEN n1=0:GOSUB 2330
1700 NEXT n
1710 PRINT
1720 GOSUB 2330
1730 RETURN
1740 REM Sous-Programme de sortie des
    donnees sauvegardees sur
    ecran ou imprimante

```

```

1750 CLS
1760 PAPER 1: PEN 0: PRINT "5. Sortie disques": PAPER 0: PEN 1
1770 PRINT
1780 GOSUB 2360
1790 PRINT
1800 PRINT "Sortie sur ecran ou sur imprimante?"
1810 INPUT "(E/I)": f2$
1820 IF f2$="" THEN RETURN
1830 f2$=LEFT$(UPPER$(f2$),1)
1840 IF f2$="I" THEN GOTO 1950
1850 REM Sortie des donnees sur le
    Moniteur
1860 CLS: FOR n=1 TO m-1
1870 n1=n1+1
1880 IF n1=5 THEN n1=0: GOSUB 2330: CLS
1890 PRINT n: PRINT P$(n): PRINT s$(n): PRINT
    n$(n)
1900 NEXT n
1910 PRINT
1920 GOSUB 2330
1930 RETURN
1940 REM Sortie des donnees sur
    imprimante
1950 FOR n=1 TO m-1
1960 PRINT #8,n: PRINT #8,P$(n): PRINT #8,
    s$(n): PRINT #8,n$(n): PRINT #8
1970 NEXT n
1980 PRINT
1990 GOSUB 2330
2000 RETURN
2010 REM Sous-Programme de modification
    des donnees en memoire
2020 CLS
2030 PAPER 1: PEN 0: PRINT "6. Modification
    des disques": PAPER 0: PEN 1
2040 PRINT
2050 GOSUB 2360
2060 PRINT

```

```

2070 INPUT "Numero du disque ": n1
2080 IF n1>m-1 OR n1<0 THEN GOSUB 2300: G
    OTO 2070
2090 IF n1=0 THEN RETURN
2100 PRINT
2110 PRINT n1: PRINT P$(n1): PRINT s$(n1):
    PRINT n$(n1)
2120 PRINT
2130 PRINT "Faut-il changer (Non=<ENTER>
    )?"
2140 PRINT P$(n1)
2150 INPUT d$: IF d$<>"" THEN P$(n1)=d$
2160 PRINT s$(n1)
2170 INPUT d$: IF d$<>"" THEN s$(n1)=d$
2180 PRINT n$(n1)
2190 INPUT d$: IF d$<>"" THEN n$(n1)=d$
2200 CLS
2210 PRINT "Entree pour numero de disque
    ": n1
2220 PRINT P$(n1)
2230 PRINT s$(n1)
2240 PRINT n$(n1)
2250 PRINT
2260 GOSUB 2330
2270 n1=0
2280 GOTO 2020
2290 REM Sous-Programme Pour message
    quand entree incorrecte
2300 PRINT: PEN effet: PRINT TAB(12): "Entr
    ee incorrecte!": PEN normal: GOSUB 2330: RE
    TURN
2310 a$=INKEY$: IF a$="" THEN GOTO 2310 E
    LSE RETURN
2320 REM Sous-Programme Pour geler
    l'affichage a l'ecran
2330 PRINT: PRINT TAB(7): "<APPuyez sur un
    e touche S.V.P.>"
2340 a$=INKEY$: IF a$="" THEN GOTO 2340 E
    LSE RETURN
2350 REM Sous-Programme general
2360 PRINT "Retour au Menu avec <ENTER>,
    sans entrer aucun mot"
2370 RETURN

```

Tableau sportif

Ce programme peut vous réservé de très bons moments et vous faire en outre gagner beaucoup de temps si vous vous intéressez au sport.

Votre ordinateur va en effet vous aider à gérer les nombreux événements réguliers qui émaillent une année sportive. Nous avons choisi comme exemple le football qui a donc servi de référence pour notre programme.

au début de la routine d'entrée, vous pouvez choisir de gérer jusqu'à 20 clubs. Le CPC vous demande ensuite si les noms des clubs qui figurent déjà en lignes DATA (clubs de 1ère division) correspondent bien aux clubs en compétition cette saison ou si vous voulez entrer vous-même une liste de jusqu'à 20 clubs. Le fonctionnement de ce programme est très simple: il vous permet d'entrer les résultats de chaque journée dès qu'ils sont connus et de voir alors le tableau qui se dégage de ces résultats. Vous pouvez consulter les résultats de toutes les journées que vous avez déjà entrées et bien sûr vous pouvez également sauvegarder les résultats sur cassette.

Vous pouvez entrer la liste des clubs et le calendrier en mode direct ou les modifier dans les lignes de DATA. Vous pourrez ensuite recharger à tous moments toutes les données que vous avez déjà sauvegardées sur cassette, en choisissant l'option charger un tableau en début de programme. Nous vous souhaitons un grand championnat!

```

10 REM Tableau sPortif
20 REM Programmes Basic Pour le CPC 464
30 REM CoPyright 1984 DATA BECKER &
    Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effet=2:n
ormal=1:PEN 1:PAPER 0
50 MODE 1
54 INPUT "Charger un tableau (0/ )";f$
56 MODE 1:f$=UPPER$(f$):IF LEFT$(f$,1)="0"
    THEN GOSUB 7000:GOTO 530
60 REM Choix du nombre de clubs
    (de 4 a 20)
70 INPUT "Combien de clubs (4/6/8/10/12/
    14/16/18 ou 20) ";f$
80 REM Liste des variables qui sont
    dimensionnee avec leurs
    significations respectives.
90 REM a$( )=nom de club
    b( , )=rencontre ayant eu lieu
    c( , )=resultat
    e( , )=total buts
    f( , )=total buts encaissee
100 REM g( , )=total matchs gagnes
    h( , )=total matchs nuls
    i( , )=total matchs Perdus
    k( , , )=rencontres jumelees
    z1 bis z5=Pour routine de tri
110 f=VAL(f$):IF f/2<>INT(f/2) THEN PRIN
T:GOSUB 1290:GOTO 50
120 IF f<4 OR f>20 THEN GOSUB 1290:GOTO
    50
125 GOSUB 130:GOTO 140
130 DIM a$(f),b(f*2,f/2),c(f,f*2-2,2),e(
    f),f(f),g(f),h(f),i(f),k(f-1,f/2,2),z1(f
    +1),z2(f+1),z3(f+1),z4(f+1),z5(f+1):RETU
    RN
140 REM Entrée directe de noms de clubs
    ou entrée des noms de clubs
    a Partir des lignes de DATA
150 INPUT "Noms des lignes de Programme
    (0/ )";f$

```

```

160 f$=UPPER$(f$):IF LEFT$(f$,1)="0" THE
    N GOTO 230
170 FOR n=1 TO f
180 PRINT "Club":n:INPUT ":";f$
190 IF f$="" THEN GOTO 180
200 IF LEN(f$)>>10 THEN a$(n)=LEFT$(f$,10
    ) ELSE a$(n)=f$
210 NEXT n
220 GOTO 350
230 ON f GOSUB 250,250,250,250,250,260,2
    50,270,250,280,250,290,250,300,250,310,2
    50,320,250,330
240 GOTO 350
250 RESTORE 6350:FOR n=1 TO f:READ a$(n)
    :NEXT n:RETURN
260 RESTORE 6320:FOR n=1 TO f:READ a$(n)
    :NEXT n:RETURN
270 RESTORE 6290:FOR n=1 TO f:READ a$(n)
    :NEXT n:RETURN
280 RESTORE 6260:FOR n=1 TO f:READ a$(n)
    :NEXT n:RETURN
290 RESTORE 6230:FOR n=1 TO f:READ a$(n)
    :NEXT n:RETURN
300 RESTORE 6200:FOR n=1 TO f:READ a$(n)
    :NEXT n:RETURN
310 RESTORE 6170:FOR n=1 TO f:READ a$(n)
    :NEXT n:RETURN
320 RESTORE 6130:FOR n=1 TO f:READ a$(n)
    :NEXT n:RETURN
330 RESTORE 6080:FOR n=1 TO f:READ a$(n)
    :NEXT n:RETURN
340 REM Le calendrier en lignes de DATA
    (système habituel) Peut être
    utilisé, mais il est également
    possible d'entrer son propre
    calendrier
350 INPUT "Calendrier en lignes DATA (0/
    )";f$:f$=UPPER$(f$):IF LEFT$(f$,1)="0"
    THEN GOTO 420

```

```

360 PRINT:PRINT:PRINT "Veuillez entrer l
e calendrier avec des chiffres (Exemple
: 1<virgule>2<ENTER>) 0 Pour renouveler
derniere entree"
370 FOR n=1 TO f-1:PRINT:PRINT:PRINT "Jo
urnee":n:PRINT
380 FOR m=1 TO f/2
390 PRINT:PRINT "Match":m:INPUT a,b:IF a
>f OR b>f OR a=b THEN GOTO 390 ELSE IF a
=0 OR b=0 THEN m=m-1:GOTO 390 ELSE k(n,m
,1)=a:k(n,m,2)=b:a=0:b=0:PRINT a$(k(n,m
,1));" - ";a$(k(n,m,2))
400 NEXT m,n
410 GOTO 530
420 ON f GOSUB 440,440,440,440,440,450,4
40,460,440,470,440,480,440,490,440,500,4
40,510,440,520
430 GOTO 530
440 RESTORE 5000:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
450 RESTORE 5040:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
460 RESTORE 5100:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
470 RESTORE 5180:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
480 RESTORE 5280:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
490 RESTORE 5400:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
500 RESTORE 5540:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
510 RESTORE 5700:FOR n=1 TO f-1:FOR m=1

```

```

TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
520 RESTORE 5880:FOR n=1 TO f-1:FOR m=1
TO f/2:READ k(n,m,1),k(n,m,2):NEXT m,n:R
ETURN
530 MODE 1: PEN effet:LOCATE 7,7:PRINT "T
ableau des clubs"::PEN normal
540 LOCATE 7,10
550 PRINT "1. Entrer journee"
560 LOCATE 7,11
570 PRINT "2. Afficher tableau actuel"
580 LOCATE 7,12
590 PRINT "3. Examiner Journee"
592 LOCATE 7,13
594 PRINT "4. Sauvegarder tableau actuel"
600 LOCATE 7,15
610 INPUT "Votre choix (1 a 4)":f$
620 IF VAL(f$)<1 OR VAL(f$)>4 THEN GOSUB
1270:GOSUB 1290:GOTO 420
630 ON VAL(f$) GOSUB 660,900,1010,8000
640 GOTO 530
650 REM Entrer Journee
660 MODE 1
670 INPUT "Quelle journee":f$:IF VAL(f$)
<1 OR VAL(f$)>f$*2-2 THEN GOSUB 1290:GOT
O 670
680 IF VAL(f$)>f-1 THEN GOTO 740
690 FOR n=1 TO f/2:PRINT "Match":n
700 IF b(VAL(f$),n)>0 THEN PRINT CHR$(7
);"Erreur! Resultat connu"::PRINT a$(k(V
AL(f$),n,1));" - ";a$(k(VAL(f$),n,2));":";
c(k(VAL(f$),n,1),VAL(f$),1);"-";c(k(VAL(
f$),n,1),VAL(f$),2):GOTO 730
710 PRINT a$(k(VAL(f$),n,1));" - ";a$(k(
VAL(f$),n,2));:INPUT a,b:IF a=99 THEN PR
INT "annule":b(VAL(f$),n)=0:GOTO 730 ELS
E GOSUB 790
720 b(VAL(f$),n)=1:c(k(VAL(f$),n,1),VAL(
f$),1)=a:c(k(VAL(f$),n,1),VAL(f$),2)=b:c(
k(VAL(f$),n,2),VAL(f$),1)=b:c(k(VAL(f$)
,n,2),VAL(f$),2)=a

```

```

730 NEXT n:RETURN
740 FOR n=1 TO f/2:PRINT "Match":n
750 IF b<VAL(f$),n)>>0 THEN PRINT CHR$(7)
    ));"Erreur! Resultat connu":PRINT a$(k(V
    RL(f$)-f+1,n,2)):;"-";a$(k(VRL(f$)-f+1,n,
    1));;"";c(k(VRL(f$)-f+1,n,1),VRL(f$),1);
    "-");c(k(VRL(f$)-f+1,n,1),VRL(f$),2):GOTO
    780
760 PRINT a$(k(VRL(f$)-f+1,n,2));;"-";a
    $(k(VRL(f$)-f+1,n,1));:INPUT a,b:IF a=99
    THEN PRINT "annule":b(VAL(f$),n)=0:GOTO
    780 ELSE GOSUB 840
770 b(VAL(f$),n)=1:c(k(VRL(f$)-f+1,n,1),
    VRL(f$),1)=a:c(k(VRL(f$)-f+1,n,1),VRL(f$),
    2)=b:c(k(VRL(f$)-f+1,n,2),VRL(f$),1)=b
    :c(k(VRL(f$)-f+1,n,2),VRL(f$),2)=a
780 NEXT n:RETURN
790 e(k(VRL(f$),n,1))=e(k(VRL(f$),n,1))+
    a:f(k(VRL(f$),n,1))=f(k(VRL(f$),n,1))+b:
    e(k(VRL(f$),n,2))=e(k(VRL(f$),n,2))+b:f(
    k(VRL(f$),n,2))=f(k(VRL(f$),n,2))+a
800 IF a=b THEN h(k(VRL(f$),n,1))=h(k(VA
    L(f$),n,1))+1:h(k(VRL(f$),n,2))=h(k(VRL(
    f$),n,2))+1
810 IF a>b THEN 900(k(VRL(f$),n,1))=900(k(VA
    L(f$),n,1))+2:i(k(VRL(f$),n,2))=i(k(VRL(
    f$),n,2))+2
820 IF a<b THEN i(k(VRL(f$),n,1))=i(k(VA
    L(f$),n,1))+2:900(k(VRL(f$),n,2))=900(k(VRL(
    f$),n,2))+2
830 RETURN
840 e(k(VRL(f$)-f+1,n,1))=e(k(VRL(f$)-f+
    1,n,1))+a:f(k(VRL(f$)-f+1,n,1))=f(k(VRL(
    f$)-f+1,n,1))+b:e(k(VRL(f$)-f+1,n,2))=e(
    k(VRL(f$)-f+1,n,2))+b:f(k(VRL(f$)-f+1,n,
    2))=f(k(VRL(f$)-f+1,n,2))+a
850 IF a=b THEN h(k(VRL(f$)-f+1,n,1))=h(
    k(VRL(f$)-f+1,n,1))+1:h(k(VRL(f$)-f+1,n,
    2))=h(k(VRL(f$)-f+1,n,2))+1
860 IF a>b THEN 900(k(VRL(f$)-f+1,n,1))=900(
    k(VRL(f$)-f+1,n,1))+2:i(k(VRL(f$)-f+1,n,
    2))=i(k(VRL(f$)-f+1,n,2))+2

```

```

870 IF a<b THEN i(k(VRL(f$)-f+1,n,1))=i(
    k(VRL(f$)-f+1,n,1))+2:900(k(VRL(f$)-f+1,n,
    2))=900(k(VRL(f$)-f+1,n,2))+2
880 RETURN
890 REM Afficher le tableau actuel
900 MODE 2:GOSUB 1180
910 PRINT"Pl. Club Matchs + ="
    - Goal average <> Points Pl.
    "
920 PRINT STRING$(71,"="):FOR n=1 TO f
930 PRINT n;TAB(5);a$(z5(n));TAB(18);(9(
    z5(n))+h(z5(n))*2)+i(z5(n))/2;TAB(24);
    9(z5(n))/2;TAB(29);h(z5(n));TAB(34);i(z5(
    n))/2;TAB(40);e(z5(n));TAB(45)):";f(z5(
    n));
940 PRINT TAB(51):e(z5(n))-f(z5(n));TAB(
    59);9(z5(n))+h(z5(n));TAB(68):n
950 NEXT n
960 PRINT STRING$(71,"=")
970 PRINT TAB(21);"(Appuyez sur une touc
    he S.V.P.)"
980 a$=INKEY$:IF a$="" THEN GOTO 980
990 RETURN
1000 REM Examiner journee
1010 CLS:PRINT:INPUT "Toute la saison (0
    / ) ";f$:f$=UPPER$(f$):IF LEFT$(f$,1)="0
    " THEN GOTO 1100
1020 PRINT "Journee ( 1 a ";f$2-2):INPUT
    " ";f$:IF VAL(f$)<1 OR VAL(f$)>f$2-2 TH
    EN GOSUB 1290:GOTO 1020 ELSE IF VAL(f$)>
    f-1 THEN GOTO 1060
1030 n=VAL(f$):CLS:PRINT "Journee":n:PRI
    NT:PRINT:PRINT:FOR m=1 TO f/2:PRINT m;TR
    B(5);a$(k(n,m,1));TAB(15);a$(k(n,m,2));
    1040 IF b(n,m)=0 THEN PRINT TAB(33);"-"
    ;"-":ELSE PRINT TAB(32);c(k(n,m,1),n,1);
    TAB(36);";";TAB(37);c(k(n,m,1),n,2);
    1050 NEXT m:PRINT:PRINT:GOSUB 1270:GOTO
    1170

```

```

1060 n=VAL(f$):CLS:PRINT "Journee":n:n=n-f-1:PRINT:PRINT:PRINT:FOR m=1 TO f/2:PRINT m:TAB(5):a$(k(n,m,2)):TAB(14):a$(k(n,m,1)):TAB(23):
1070 IF b(n,m)=1 THEN PRINT c(k(n,m,1),n,2):TAB(27);"-";TAB(28):c(k(n,m,1),n,1):ELSE PRINT "- - -":
1080 IF b(n+f-1,m)=0 THEN PRINT TAB(33);"- - -":ELSE PRINT TAB(32):c(k(n,m,1),n+f-1,1):TAB(36);"-";TAB(37):c(k(n,m,1),n+f-1,2):
1090 NEXT m:PRINT:PRINT:GOSUB 1270:GOTO 1170
1100 FOR n=1 TO f-1:CLS:PRINT "Journee":n:PRINT:PRINT:PRINT:FOR m=1 TO f/2:PRINT m:TAB(5):a$(k(n,m,1)):TAB(15):a$(k(n,m,2)):
1110 IF b(n,m)=0 THEN PRINT TAB(33);"- - -":ELSE PRINT TAB(32):c(k(n,m,1),n,1):TAB(36);"-";TAB(37):c(k(n,m,1),n,2):
1120 NEXT m:PRINT:PRINT:GOSUB 1270:NEXT n
1130 FOR n=1 TO f-1:CLS:PRINT "Spielta9":n+f-1:PRINT:PRINT:PRINT:FOR m=1 TO f/2:PRINT m:TAB(5):a$(k(n,m,2)):TAB(14):a$(k(n,m,1)):TAB(23):
1140 IF b(n,m)=0 THEN PRINT "- - -":ELSE PRINT c(k(n,m,1),n,2):TAB(27);"-";TAB(28):c(k(n,m,1),n,1):
1150 IF b(n+f-1,m)=0 THEN PRINT TAB(33);"- - -":ELSE PRINT TAB(32):c(k(n,m,1),n+f-1,1):TAB(36);"-";TAB(37):c(k(n,m,1),n+f-1,2):
1160 NEXT m:PRINT:PRINT:GOSUB 1270:NEXT 1170 RETURN
1180 REM Sous-routine de tri
1190 FOR n=1 TO f:z1(n)=g(n)+h(n):z2(n)=e(n)-f(n):z3(n)=e(n):z4(n)=-1:NEXT n
1200 FOR n=1 TO f
1210 FOR m=1 TO f

```

```

1220 IF z1(m)>z4(n) THEN z4(n)=z1(m):a=m:GOTO 1250
1230 IF z1(m)=z4(n) AND z2(m)>z2(a) THEN z4(n)=z1(m):a=m:GOTO 1250
1240 IF z1(m)=z4(n) AND z2(m)=z2(a) AND z3(m)>z3(a) THEN z4(n)=z1(m):a=m
1250 NEXT m:z1(a)=-1:z5(n)=a:a=f+1:NEXT n
1260 RETURN
1270 PRINT:PRINT TAB(7);"(APPuyez sur une touche S.V.P.)"
1280 a$=INKEY$:IF a$="" THEN GOTO 1280 ELSE RETURN
1290 PEN effet:PRINT TAB(12);"Entree incorrecte!":PEN normal:GOSUB 1270:RETURN
5000 REM Datas Pour 4 clubs
5010 DATA 1,4,2,3
5020 DATA 4,3,1,2
5030 DATA 2,4,3,1
5040 REM Datas Pour 6 clubs
5050 DATA 1,6,2,5,3,4
5060 DATA 6,4,5,3,1,2
5070 DATA 2,6,3,1,4,5
5080 DATA 6,5,1,4,2,3
5090 DATA 3,6,4,2,5,1
5100 REM Datas Pour 8 clubs
5110 DATA 1,8,2,7,3,6,4,5
5120 DATA 8,5,6,4,7,3,1,2
5130 DATA 2,8,3,1,4,7,5,6
5140 DATA 8,6,7,5,1,4,2,3
5150 DATA 3,8,4,2,5,1,6,7
5160 DATA 8,7,1,6,2,5,3,4
5170 DATA 4,8,5,3,6,2,7,1
5180 REM Datas Pour 10 clubs
5190 DATA 1,10,2,9,3,8,4,7,5,6
5200 DATA 10,6,7,5,8,4,9,3,1,2
5210 DATA 2,10,3,1,4,9,5,8,6,7
5220 DATA 10,7,8,6,9,5,1,4,2,3
5230 DATA 3,10,4,2,5,1,6,9,7,8
5240 DATA 10,8,9,7,1,6,2,5,3,4

```

5250 DATA 4,10,5,3,6,2,7,1,8,9
5260 DATA 10,9,1,8,2,7,3,6,4,5
5270 DATA 5,10,6,4,7,3,8,2,9,1
5280 REM Datas Pour 12 clubs
5290 DATA 1,12,2,11,3,10,4,9,5,8,6,7
5300 DATA 12,7,8,6,9,5,10,4,11,3,1,2
5310 DATA 2,12,3,1,4,11,5,10,6,9,7,8
5320 DATA 12,8,9,7,10,6,11,5,1,4,2,3
5330 DATA 3,12,4,2,5,1,6,11,7,10,8,9
5340 DATA 12,9,10,8,11,7,1,6,2,5,3,4
5350 DATA 4,12,5,3,6,2,7,1,8,11,9,10
5360 DATA 12,10,11,9,1,8,2,7,3,6,4,5
5370 DATA 5,12,6,4,7,3,8,2,9,1,10,11
5380 DATA 12,11,1,10,2,9,3,8,4,7,5,6
5390 DATA 6,12,7,5,8,4,9,3,10,2,11,1
5400 REM Datas Pour 14 clubs
5410 DATA 1,14,2,13,3,12,4,11,5,10,6,9,7,
,8
5420 DATA 14,8,9,7,10,6,11,5,12,4,13,3,1
,2
5430 DATA 2,14,3,1,4,13,5,12,6,11,7,10,8
,9
5440 DATA 14,9,10,8,11,7,12,6,13,5,1,4,2
,3
5450 DATA 3,14,4,2,5,1,6,13,7,12,8,11,9,
10
5460 DATA 14,10,11,9,12,8,13,7,1,6,2,5,3
,4
5470 DATA 4,14,5,3,6,2,7,1,8,13,9,12,10,
11
5480 DATA 14,11,12,10,13,9,1,8,2,7,3,6,4
,5
5490 DATA 5,14,6,4,7,3,8,2,9,1,10,13,11,
12
5500 DATA 14,12,13,11,1,10,2,9,3,8,4,7,5
,6
5510 DATA 6,14,7,5,8,4,9,3,10,2,11,1,12,
13
5520 DATA 14,13,1,12,2,11,3,10,4,9,5,8,6
,7
5530 DATA 7,14,8,6,9,5,10,4,11,3,12,2,13
,1
5540 REM Datas Pour 16 clubs

5550 DATA 1,16,2,15,3,14,4,13,5,12,6,11,
7,10,8,9
5560 DATA 16,9,10,8,11,7,12,6,13,5,14,4,
15,3,1,2
5570 DATA 2,16,3,1,4,15,5,14,6,13,7,12,8
,11,9,10
5580 DATA 16,10,11,9,12,8,13,7,14,6,15,5
,1,4,2,3
5590 DATA 3,16,4,2,5,1,6,15,7,14,8,13,9,
12,10,11
5600 DATA 16,11,12,10,13,9,14,8,15,7,1,6
,2,5,3,4
5610 DATA 4,16,5,3,6,2,7,1,8,15,9,14,10,
13,11,12
5620 DATA 16,12,13,11,14,10,15,9,1,8,2,7
,3,6,4,5
5630 DATA 5,16,6,4,7,3,8,2,9,1,10,15,11,
14,12,13
5640 DATA 16,13,14,12,15,11,1,10,2,9,3,8
,4,7,5,6
5650 DATA 6,16,7,5,8,4,9,3,10,2,11,1,12,
15,13,14
5660 DATA 16,14,15,13,1,12,2,11,3,10,4,9
,5,8,6,7
5670 DATA 7,16,8,6,9,5,10,4,11,3,12,2,13
,1,14,15
5680 DATA 16,15,1,14,2,13,3,12,4,11,5,10
,6,9,7,8
5690 DATA 8,16,9,7,10,6,11,5,12,4,13,3,1
4,2,15,1
5700 REM Datas Pour 18 clubs
5710 DATA 1,18,2,17,3,16,4,15,5,14,6,13,
7,12,8,11,9,10
5720 DATA 18,19,10,11,9,12,8,13,7,14,6,15,5
,16,4,17,3,1,2
5730 DATA 2,18,3,1,4,17,5,16,6,15,7,14,8
,13,9,12,10,11
5740 DATA 18,11,12,10,13,9,14,8,15,7,16,
6,17,5,1,4,2,3
5750 DATA 3,18,4,2,5,1,6,17,7,16,8,15,9,
14,10,13,11,12

5760 DATA 18,12,13,11,14,10,15,9,16,8,17,
,7,1,6,2,5,3,4
5770 DATA 4,18,5,3,6,2,7,1,8,17,9,16,10,
15,11,14,12,13
5780 DATA 18,13,14,12,15,11,16,10,17,9,1
,8,2,7,3,6,4,5
5790 DATA 5,18,6,4,7,3,8,2,9,1,10,17,11,
16,12,15,13,14
5800 DATA 18,14,15,13,16,12,17,11,1,10,2
,9,3,8,4,7,5,6
5810 DATA 6,18,7,5,8,4,9,3,10,2,11,1,12,
17,13,16,14,15
5820 DATA 18,15,16,14,17,13,1,12,2,11,3,
10,4,9,5,8,6,7
5830 DATA 7,18,8,6,9,5,10,4,11,3,12,2,13
,1,14,17,15,16
5840 DATA 18,16,17,15,1,14,2,13,3,12,4,1
1,5,10,6,9,7,8
5850 DATA 8,18,9,7,10,6,11,5,12,4,13,3,1
4,2,15,1,16,17
5860 DATA 18,17,1,16,2,15,3,14,4,13,5,12
,6,11,7,10,8,9
5870 DATA 9,18,10,8,11,7,12,6,13,5,14,4,
15,3,16,2,17,1
5880 REM Datas Pour 20 clubs
5890 DATA 1,29,2,19,3,18,4,17,5,16,6,15,
7,14,8,13,9,12,10,11
5900 DATA 20,11,12,10,13,9,14,8,15,7,16,
6,17,5,18,4,19,3,1,2
5910 DATA 2,29,3,1,4,19,5,18,6,17,7,16,8
,15,9,14,10,13,11,12
5920 DATA 20,12,13,11,14,10,15,9,16,8,17
,7,18,6,19,5,1,4,2,3
5930 DATA 3,29,4,2,5,1,6,19,7,18,8,17,9,
16,10,15,11,14,12,13
5940 DATA 20,13,14,12,15,11,16,10,17,9,1
8,8,19,7,1,6,2,5,3,4
5950 DATA 4,29,5,3,6,2,7,1,8,19,9,18,10,
17,11,16,12,15,13,14
5960 DATA 20,14,15,13,16,12,17,11,18,10,
19,9,1,8,2,7,3,6,4,5

5970 DATA 5,20,6,4,7,3,8,2,9,1,10,19,11,
18,12,17,13,16,14,15
5980 DATA 20,15,16,14,17,13,18,12,19,11,
1,10,2,9,3,8,4,7,5,6
5990 DATA 6,20,7,5,8,4,9,3,10,2,11,1,12,
19,13,18,14,17,15,16
6000 DATA 20,16,17,15,18,14,19,13,1,12,2
,11,3,10,4,9,5,8,6,7
6010 DATA 7,20,8,6,9,5,10,4,11,3,12,2,13
,1,14,19,15,18,16,17
6020 DATA 20,17,18,16,19,15,1,14,2,13,3,
12,4,11,5,10,6,9,7,8
6030 DATA 8,20,9,7,10,6,11,5,12,4,13,3,1
4,2,15,1,16,19,17,18
6040 DATA 20,18,19,17,1,16,2,15,3,14,4,1
3,5,12,6,11,7,10,8,9
6050 DATA 9,20,19,8,11,7,12,6,13,5,14,4,
15,3,16,2,17,1,18,19
6060 DATA 20,19,1,18,2,17,3,16,4,15,5,14
,6,13,7,12,8,11,9,10
6070 DATA 10,20,11,9,12,8,13,7,14,6,15,5
,16,4,17,3,18,2,19,1
6080 REM Datas Pour le championnat de
France 1984/85 1ere division
Exemple avec 20 clubs
6090 DATA Nantes,Toulon,Brest,Lille,Paris
SG,Nancy,Bordeaux,Toulouse,Bastia,Mont
aubin,Tours,Laval,Rouen,Strasbourg,Metz,RC
Paris,Lens,Rouen,Marseille,Sochaux
6170 REM Datas Pour 16 clubs
6180 DATA A,B,C,D,E,F,G,H
6190 DATA I,J,K,L,M,N,O,P
6200 REM Datas Pour 14 clubs
6210 DATA A,B,C,D,E,F,G
6220 DATA H,I,J,K,L,M,N
6230 REM Datas Pour 12 clubs
6240 DATA A,B,C,D,E,F
6250 DATA G,H,I,J,K,L
6260 REM Datas Pour 10 clubs
6270 DATA A,B,C,D,E

```

6280 DATA F,G,H,I,J
6290 REM Datas Pour 8 clubs
6300 DATA A,B,C,D
6310 DATA E,F,G,H
6320 REM Datas Pour 6 clubs
6330 DATA A,B,C
6340 REM Datas Pour 4 clubs
6350 REM Datas fuer 4 Vereine
6360 DATA A,B
6370 DATA C,D
7000 MODE 1:INPUT"Nom du tableau";nom$
7010 OPENIN MID$(nom$,1,LEN(nom$)):INPUT
#9,f:GOSUB 130
7020 FOR m=0 TO f:INPUT#9,a$(m),e(m),f(m)
>,g(m),h(m),i(m):NEXT m
7030 FOR m=0 TO f*2:FOR n=0 TO f/2:INPUT
#9,b(m,n):NEXT n,m
7040 FOR m=0 TO f:FOR n=0 TO f*2-2:FOR o
=0 TO 2:INPUT#9,c(m,n,o):NEXT o,n,m
7050 FOR m=0 TO f-1:FOR n=0 TO f/2:FOR o
=0 TO 2:INPUT#9,k(m,n,o):NEXT o,n,m
7900 CLOSEIN:RETURN
8000 MODE 1:INPUT"Nom du tableau";nom$
8010 OPENOUT nom$:WRITE#9,f
8020 FOR m=0 TO f:WRITE#9,a$(m),e(m),f(m)
>,g(m),h(m),i(m):NEXT m
8030 FOR m=0 TO f*2:FOR n=0 TO f/2:WRITE
#9,b(m,n):NEXT n,m
8040 FOR m=0 TO f:FOR n=0 TO f*2-2:FOR o
=0 TO 2:WRITE#9,c(m,n,o):NEXT o,n,m
8050 FOR m=0 TO f-1:FOR n=0 TO f/2:FOR o
=0 TO 2:WRITE#9,k(m,n,o):NEXT o,n,m
8900 CLOSEOUT:RETURN

```

CAMEROUN

Il est à craindre, comme c'est malheureusement le cas pour la plupart des ordinateurs familiaux, que l'essentiel des logiciels disponibles pour le CPC 464 ne soit dans une première période composé que de jeux. Nous avons essayé de montrer dans cette collection de programmes qu'on peut également faire des choses plus sérieuses avec un ordinateur et aussi qu'en peut essayer de comprendre un peu mieux comment il marche.

Mais nous n'avons rien contre les jeux et c'est pourquoi nous vous fournissons également dans ce recueil un jeu complet de Cameroun.

Nous avons ici utilisé les capacités du BASIC du CPC. C'est ainsi que nous avons créé quatre fenêtres différentes qui permettent un affichage très simple des entrées, directement avec les touches fléchées.

Nous espérons que vous connaissez les règles de ce jeu (chaque doit essayer avec 5 dés de remplir les 12 conditions prescrites). Si ce n'est pas le cas, consultez un livre de règles de jeux ou essayez simplement de déduire les règles en jouant plusieurs fois. Disons simplement que vous avez 12 tours et que vous devez essayer avec 3 lancers de dés chaque fois d'avoir finalement le plus de points possibles selon le barème du jeu.

Le CPC ne lance pas simplement les dés pour vous, il place en outre les dés dans un ordre croissant. Vous pouvez sélectionner avec la touche COPY et les touches curseur les dés que vous voulez

lancer à nouveau. Une fois que vous avez lancé les dés trois fois, sélectionnez avec les touches curseur et la touche ENTREE la ligne du tableau dans laquelle vous voulez finalement placer votre résultat (si vous avez beaucoup de 6, dans la ligne des 6). Vous verrez d'autre part apparaître chaque fois le nombre global de points.

Voici comment s'effectue le décompte des points:

de 1 à 6: seuls les dés correspondant au chiffre choisi sont comptés

Brelan: si vous avez un brelan (trois fois le même chiffre), tous les dés sont additionnés

Carré: si vous avez un carré (quatre fois le même chiffre), tous les dés sont additionnés

Quinton: (cinq fois le même chiffre) 50 points

petite Suite: (au moins 3 dés se suivant) 30 points

grande Suite: (au moins 4 dés se suivant) 40 points

Full House: (un brelan et une paire, Ex: 33355) 25 points

Auussez-vous bien!

```
10 REM Cameroun
20 REM Programmes Basic Pour le CPC464
30 REM Copyright 1984 DATA BECKER &
      Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,24
50 REM Preparatifs
60 MODE 1
70 INPUT "Nombre de joueurs (1 à 4) ";joueurs
80 IF joueurs<1 OR joueurs>4 THEN GOSUB
2350:GOTO 60
90 DIM jeu(joueurs,12),jeu1(joueurs,12)
100 FOR n=1 TO joueurs
110 PRINT:PRINT "Joueur";n;""
120 INPUT joueur$(n)
130 IF LEN(joueur$(n))>3 THEN joueur$(n)=LEFT$(joueur$(n),3)
140 IF LEN(joueur$(n))=2 THEN joueur$(n)=joueur$(n)+" "
150 IF LEN(joueur$(n))=1 THEN joueur$(n)=joueur$(n)+" "
160 IF LEN(joueur$(n))=0 THEN GOSUB 2350
:RUN
170 NEXT n
180 MODE 1
190 REM Fenetre Pour le jet des des
200 WINDOW #1,8,32,1,6
210 REM Fenetre Pour tableau resultats
220 WINDOW #2,2,40,7,22
230 REM Fenetre Pour informations
240 WINDOW #3,1,40,24,24
250 REM Fenetre Pour choix Par rapport
      au tableau
260 WINDOW #4,1,1,9,21
270 REM Fixation des couleurs
280 PAPER #4,2:PEN #4,0:CLS #4:PAPER #3,
      2:PEN #3,0:CLS #4:CLS #3:PAPER #1,2:PEN
      #1,0:CLS #1:PAPER #2,0:PEN #2,1:CLS #2:L
      OCATE #4,1,13:PRINT #4,CHR$(143)
290 REM Construction du tableau
```

```

300 PRINT#2,"Nom du Joueur -> ";:FOR n=1 TO joueurs:PRINT#2,joueur$(n);":":NEXT n:PRINT#2:PRINT#2:PRINT#2,"1 -----":PRINT#2,"2 -----":PRINT#2,"3 -----":PRINT#2,"4 -----":PRINT#2,"5 -----":PRINT#2,"6 -----"
310 PRINT#2,"Brelan -----":PRINT#2,"Carré -----":PRINT#2,"Quinton -----":PRINT#2,"Petite Suite -":PRINT#2,"Grand e Suite -":PRINT#2,"Full House ---":PRINT#2,STRING$(36,"="):PRINT#2,"Points -----"
320 REM Lecture de la forme des des en lignes de DATA
330 RESTORE:FOR n=1 TO 6:FOR m=1 TO 3
340 READ a$(n,m):NEXT m,n
350 REM Debut du Programme Principal: Jeu en 12 tours
360 FOR ronde=1 TO 12
370 FOR rondel=1 TO joueurs
380 CLS #3
390 LOCATE #3,8,1:PRINT #3,"C'est le tour de "joueur$(ronde1)". Tour":ronde
400 REM Produire 5 nombres aleatoires entre 1 et 6
410 FOR n=1 TO 5
420 RANDOMIZE TIME
430 a=INT(10*RND(TIME))
440 IF a<1 OR a>6 THEN GOTO 420
450 nombre(n)=a
460 REM RePrésentation des des correspondants
470 FOR m=2 TO 4
480 LOCATE #1,n*5-3,m
490 PRINT #1,a$(a,m-1)
500 NEXT m:NEXT n
510 REM Saut à la routine de tri des des
520 GOSUB 950
530 REM Chaque joueur a droit à trois lancers par tour

```

```

540 FOR comPtEUR=1 TO 2
550 REM Saut à la routine de modification
560 GOSUB 820
570 CLS #1
580 FOR n=1 TO 5
590 REM Produire 5 nombres aleatoires entre 1 et 6
600 IF wa(n)=1 THEN RANDOMIZE TIME:a=INT(10*RND(TIME)):IF a<1 OR a>6 THEN GOTO 600 ELSE nombre(n)=a ELSE a=nombre(n):GOTO 620
610 REM RePrésentation des des correspondants
620 wa(n)=0
630 FOR m=2 TO 4
640 LOCATE #1,n*5-3,m
650 PRINT #1,a$(a,m-1)
660 NEXT m
670 NEXT n
680 REM Saut à la routine de tri des des
690 GOSUB 950
700 NEXT comPtEUR:comPtEUR=0
710 REM Que faut-il faire du résultat ?
720 GOSUB 1420
730 NEXT rondel,ronde
740 REM Fin du jeu et question: Une autre Partie ?
750 a$=INKEY$
760 IF a$="" THEN GOTO 750
770 CLS
780 INPUT "Une autre Partie ( /N) ";a$
790 a$=UPPER$(a$)
800 IF LEFT$(a$,1)="N" THEN END ELSE RUN
810 REM Routine de modification
820 LOCATE #1,3,5:PRINT #1,CHR$(244)
830 a$=INKEY$
840 IF a$="" THEN GOTO 830
850 REM Touche avec flèche 'gauche' enfoncée

```

```

860 IF a$=CHR$(242) AND POS(#1)>4 THEN LOCATE #1,POS(#1)-1,5:PRINT #1," ";:LOCATE #1,POS(#1)-6,5:PRINT #1,CHR$(244);:GOTO 830
870 REM Touche avec flèche 'droite' enfoncée
880 IF a$=CHR$(243) AND POS(#1)<24 THEN LOCATE #1,POS(#1)-1,5:PRINT #1," ";:LOCATE #1,POS(#1)+4,5:PRINT #1,CHR$(244);:GOTO 830
890 REM Touche COPY enfoncée
    (effet: inversion affichage)
900 IF a$<>CHR$(224) THEN GOTO 920 ELSE
IF wa((POS(#1)+1)/5)=1 THEN wa((POS(#1)+1)/5)=0:PEN #1,2:LOCATE #1,POS(#1)-1,1:PRINT #1," ";:PEN #1,0:LOCATE #1,POS(#1),5 ELSE wa((POS(#1)+1)/5)=1:LOCATE #1,POS(#1)-1,1:PRINT #1,CHR$(245);:LOCATE #1,POS(#1),5
910 REM Touche <ENTER> enfoncée
920 IF a$=CHR$(13) THEN RETURN
930 GOTO 830
940 REM Routines de tri des des
950 FOR nn=1 TO 5
960 z2(nn)=7
970 NEXT nn
980 FOR nn=1 TO 5
990 FOR mm=1 TO 5
1000 IF nombre(mm)<=z2(nn) THEN z2(nn)=nombre(mm):z1(nn)=wa(mm):marque=mm
1010 NEXT mm
1020 nombre(marque)=7
1030 NEXT nn
1040 FOR nn=1 TO 5
1050 nombre(nn)=z2(nn)
1060 wa(nn)=z1(nn)
1070 NEXT nn
1080 CLS #1
1090 FOR nn=1 TO 5
1100 FOR mm=2 TO 4

```

```

1110 LOCATE #1,nn*5-3,mm:PRINT #1,a$(nombre(nn)),mm-1)
1120 NEXT mm
1130 NEXT nn
1140 LOCATE #1,1,6:PRINT #1,STRING$(8,CHR$(143));" Lancer";compteur+1;STRING$(7,CHR$(143));
1150 RETURN
1160 REM Forme des différentes faces de de
1170 REM Face de de 1
1180 DATA " "
1190 DATA " 0 "
1200 DATA " "
1210 REM Face de de 2
1220 DATA " 0"
1230 DATA " "
1240 DATA "0 "
1250 REM Face de de 3
1260 DATA " 0"
1270 DATA " 0 "
1280 DATA "0 "
1290 REM Face de de 4
1300 DATA "0 0"
1310 DATA " "
1320 DATA "0 0"
1330 REM Face de de 5
1340 DATA "0 0"
1350 DATA " 0 "
1360 DATA "0 0"
1370 REM Face de de 6
1380 DATA "000"
1390 DATA " "
1400 DATA "000"
1410 REM Routine d'entrée Pour les valeurs du tableau
1420 CLS #4
1430 LOCATE #4,1,13
1440 PRINT #4,CHR$(143);
1450 LOCATE #4,1,1
1460 PRINT #4,CHR$(243);

```

```

1470 a$=INKEY$
1480 IF a$="" THEN GOTO 1470
1490 REM Touche avec fleche 'bas'
    enfoncée
1500 IF a$=CHR$(241) AND VPOS(#4)<13 THEN
    LOCATE #4,1,VPOS(#4)-1 ELSE GOTO 1550
1510 PRINT #4," "
1520 LOCATE #4,1,VPOS(#4)
1530 PRINT #4,CHR$(243);
1540 REM Touche avec fleche 'haut'
    enfoncée
1550 IF a$=CHR$(240) AND VPOS(#4)>2 THEN
    LOCATE #4,1,VPOS(#4)-1 ELSE GOTO 1600
1560 PRINT #4," "
1570 LOCATE #4,1,VPOS(#4)-2
1580 PRINT #4,CHR$(243);
1590 REM Touche <ENTER> enfoncée,
    Si valeur Pas encore
    enregistree, elle doit être
    rangée et le nombre de Points
    est calculé avec un
    sous-Programme
1600 IF a$=CHR$(13) THEN IF jeu1(ronde1,
    VPOS(#4)-1)>1 THEN jeu1(ronde1,VPOS(#4)
    -1)=1:GOSUB 1820 ELSE GOTO 1470 ELSE GOT
    O 1470
1610 jeu(ronde1,ronde)=resultat
1620 CLS #3
1630 LOCATE #3,9,1
1640 PRINT #3,"Joueur ";joueur$(ronde1);
    ":";resultat;"Points"
1650 REM Incription du resultat dans
    le tableau
1660 LOCATE #2,14+ronde1*5,VPOS(#4)+1
1670 PRINT #2,resultat;
1680 LOCATE #2,14+ronde1*5,16
1690 FOR n=1 TO 12
1700 resultatfin(ronde1)=resultatfin(ron
    de1)+jeu(ronde1,n)
1710 NEXT n
1720 PRINT #2,resultatfin(ronde1);

```

```

1730 resultatfin(ronde1)=0
1740 a$=INKEY$
1750 IF a$="" THEN GOTO 1740
1760 CLS #1
1770 CLS #4
1780 LOCATE #4,1,13
1790 PRINT #4,CHR$(143);
1800 CLS #3
1810 RETURN
1820 resultat=0
1830 REM Sous-Programmes de calcul du
    nombre de Points obtenus
1840 ON VPOS(#4)-1 GOSUB 1870,1920,1970,
    2020,2070,2120,2170,2200,2230,2260,2300,
    2330
1850 RETURN
1860 REM Compter les 1
    (Addition de tous les 1)
1870 FOR n=1 TO 5
1880 IF nombre(n)=1 THEN resultat=result
    at+1
1890 NEXT n
1900 RETURN
1910 REM Compter les 2
    (Addition de tous les 2)
1920 FOR n=1 TO 5
1930 IF nombre(n)=2 THEN resultat=result
    at+2
1940 NEXT n
1950 RETURN
1960 REM Compter les 3
    (Addition de tous les 3)
1970 FOR n=1 TO 5
1980 IF nombre(n)=3 THEN resultat=result
    at+3
1990 NEXT n
2000 RETURN
2010 REM Compter les 4
    (Addition de tous les 4)
2020 FOR n=1 TO 5

```

```

2030 IF nombre(n)=4 THEN resultat=resultat
at+4
2040 NEXT n
2050 RETURN
2060 REM Compter les 5
    (Addition de tous les 5)
2070 FOR n=1 TO 5
2080 IF nombre(n)=5 THEN resultat=resultat
at+5
2090 NEXT n
2100 RETURN
2110 REM Compter les 6
    (Addition de tous les 6)
2120 FOR n=1 TO 5
2130 IF nombre(n)=6 THEN resultat=resultat
at+6
2140 NEXT n
2150 RETURN
2160 REM Test du brelan;
    Si brelan, Points de tous les
    des
2170 IF (nombre(1)=nombre(2) AND nombre(
2)=nombre(3)) OR (nombre(2)=nombre(3) AN
D nombre(3)=nombre(4)) OR (nombre(3)=nom
bre(4) AND nombre(4)=nombre(5)) THEN FOR
n=1 TO 5:resultat=resultat+nombre(n):NE
XT n
2180 RETURN
2190 REM Test du carre;
2200 IF (nombre(1)=nombre(2) AND nombre(
2)=nombre(3) AND nombre(3)=nombre(4)) OR
(nombre(2)=nombre(3) AND nombre(3)=nom
bre(4) AND nombre(4)=nombre(5)) THEN FOR
n=1 TO 5:resultat=resultat+nombre(n):NE
XT n
2210 RETURN
2220 REM Test du Quinton;
    Quinton => 50 Points
2230 IF nombre(1)=nombre(2) AND nombre(2)
=>nombre(3) AND nombre(3)=nombre(4) AND
nombre(4)=nombre(5) THEN resultat=50

```

```

2240 RETURN
2250 REM Test de la Petite suite;
    Petite suite=30 Points
2260 IF nombre(1)=nombre(2)-1 AND nombre(
2)=nombre(3)-1 THEN resultat=30
2262 IF nombre(2)=nombre(3)-1 AND nombre(
3)=nombre(4)-1 THEN resultat=30
2264 IF nombre(3)=nombre(4)-1 AND nombre(
4)=nombre(5)-1 THEN resultat=30
2266 IF nombre(1)=nombre(4)-1 AND nombre(
2)=nombre(5)-1 THEN resultat=30
2270 IF nombre(2)=nombre(4)-1 AND nombre(
4)=nombre(5)-1 THEN resultat=30
2280 RETURN
2290 REM Test de la grande suite;
    Grande suite=40 Points
2300 IF nombre(1)=nombre(2)-1 AND nombre(
2)=nombre(3)-1 AND nombre(3)=nombre(4)-
1 THEN resultat=40
2305 IF nombre(2)=nombre(3)-1 AND nombre(
3)=nombre(4)-1 AND nombre(4)=nombre(5)-
1 THEN resultat=40
2307 IF nombre(1)=nombre(3)-1 AND nombre(
3)=nombre(4)-1 AND nombre(4)=nombre(5)-
1 THEN resultat=40
2310 RETURN
2320 REM Test Full House;
    Full House = 25 Points
2330 IF (nombre(1)=nombre(2) AND nombre(
3)=nombre(4) AND nombre(4)=nombre(5)) OR
(nombre(1)=nombre(2) AND nombre(2)=nom
bre(3) AND nombre(4)=nombre(5)) THEN resu
ltat=25
2340 RETURN
2350 a$=INKEY$: IF a$="" THEN GOTO 2350 E
LSE PRINT ASC(a$):GOTO 2350

```

Ce jeu existe en de nombreuses versions, par exemple sous le nom de Matamiki.

Nous avons préféré le nom de Décodeur parce que notre programme s'écligne malgré tout notablement de ses modèles.

Vous pouvez tout d'abord choisir parmi combien de nombres-couleurs l'ordinateur pourra sélectionner les nombres-couleurs dont doit se composer le code. Vous disposez d'un maximum de 8 codes-couleurs.

Vous devez ensuite faire un second choix: de combien d'éléments le code doit-il être composé? Il peut y avoir plus de nombres-couleurs que d'éléments du code mais pas le contraire.

L'écran affiche après chaque entrée combien de couleurs sont justes et se trouvent effectivement dans le code mais aussi combien de couleurs justes ont été également placées à la bonne place. Les débutants en Décodeur auront certainement du mal mais c'est en forgeant qu'on devient forgeron. Vous pouvez à tout moment consulter la liste de vos entrées en appuyant sur la touche ENTER.

Nous vous souhaitons maintenant de bien vous amuser avec ce jeu.

```

10 REM Sport cerebral
20 REM Programmes Basic Pour le CPC464
30 REM Copyright 1984 DATA BECKER &
      Rainer Lueers
40 REM Preparatifs et dimensionnements
50 DIM marque$(100),noir(100),blanc(100)
60 PAPER 0: PEN 1: MODE 1
70 INPUT "Nombre de couleurs (1 a 8) ";f$
80 IF VAL(f$)<1 OR VAL(f$)>8 THEN GOSUB
460:GOTO 70
90 INPUT "Nombre de chiffres (1 a 8) ";f1
$ 
100 IF VAL(f1$)<1 OR VAL(f1$)>8 THEN GOSUB
140:GOTO 90
110 IF VAL(f1$)>VAL(f$) THEN GOSUB 460:G
OTO 90
120 REM Choix des nombres par la
      fonction de hasard RND
130 FOR n=1 TO VAL(f1$)
140 a=INT(10*RND(TIME))
150 IF a<1 OR a>VAL(f$) THEN GOTO 140
160 REM Tous les nombres choisis
      doivent etre differents
170 FOR n=1 TO n-i
180 a(n)=VAL(MID$(a$,n,1))
190 NEXT n
200 FOR m=1 TO n-1
210 IF a(m)=a(n) THEN GOTO 140 ELSE NEXT m
220 a$=a$+RIGHT$(STR$(a),1)
230 NEXT n
240 FOR n=1 TO VAL(f1$)
250 a(n)=VAL(MID$(a$,n,1))
260 NEXT n
270 CLS
280 REM Routine d'entree
290 PRINT "Un nombre avec";VAL(f1$); "chif
      fres": INPUT f3$
300 REM Listage de tous les nombres qui
      ont ete entrez jusqu'ici

```

```

1000 REM Jeux de réflexes
1100 REM Auteur: J.-P. MAY
1200 REM Date: 1982-06-15
1300 REM Version: 1.0
1400 REM Langage: BASIC CPC
1500 REM Version: CPC 464
1600 REM Version: CPC 6128
1700 REM Version: CPC 664
1800 REM Version: CPC 6100
1900 REM Version: CPC 6128
2000 REM Version: CPC 6128
2100 REM Version: CPC 6128
2200 REM Version: CPC 6128
2300 REM Version: CPC 6128
2400 REM Version: CPC 6128
2500 REM Version: CPC 6128
2600 REM Version: CPC 6128
2700 REM Version: CPC 6128
2800 REM Version: CPC 6128
2900 REM Version: CPC 6128
3000 REM Version: CPC 6128
3100 IF f3$="" THEN CLS:FOR z1=1 TO z:PRI
NT"Tentative":z1;TAB(8):marque$(z1):"
OK:";noir(z1):" Pas OK:";blanc(z1):NEXT
z1:PRINT:GOTO 290 ELSE IF LEN(f3$)>VAL
(f1$) THEN GOSUB 460:GOTO 290
320 FOR n=1 TO VAL(f1$)
330 b(n)=VAL(MID$(f3$,n,1))
340 IF a(n)=b(n) THEN noir=noir+1:c(n)=-
1
350 NEXT n
360 FOR m=1 TO VAL(f1$)
370 FOR n=1 TO VAL(f1$)
380 IF a(n)=b(m) AND n>m THEN blanc=bla
nc+1:GOTO 410
390 NEXT n
400 REM Routine de sortie avec contrôle
de la justesse
410 NEXT n:PRINT"Couleurs en mauvaise Po
sition";blanc
420 z=z+1:marque$(z)=f3$:blanc(z)=blanc:-
noir(z)=noir:blanc=0
430 PRINT"Couleurs en bonne Position";\n
noir
440 IF noir=VAL(f1$) THEN PRINT:PAPER 1:
PEN 0:PRINT"Parfaitement exact en":z;"co
ups";:PAPER 0:PEN 1:PRINT:PRINT:INPUT"En
core une fois (O/ ) ";f$:IF UPPER$(LEFT$(f$,
1))="O" THEN RUN ELSE END ELSE noir=
0:GOTO 290
450 GOTO 290
460 PRINT:PRINT"Entrée incorrecte!":PRIN
T:RETURN

```

Réflexes

Le jeu consiste à faire l'ordinateur répondre à certaines questions.

Nous vous recommandons ce programme si vous voulez essayer de surprendre votre monde ou même pour le présenter à des personnes plutôt hostiles à tout ce qui touche aux ordinateurs.

Le but de ce programme est uniquement de mesurer le temps de réaction: il faut appuyer sur une touche quelconque du clavier dès que l'ordinateur vous le demande. Suit alors une évaluation par l'ordinateur des réflexes de la personne qui vient de jouer et on vous demande alors si vous voulez à nouveau participer à une partie avec un maximum de quatre participants.

Mais l'intérêt principal de ce programme est la façon dont les utilisateurs de l'ordinateur sont traités. Les ordinateurs, comme vous le savez certainement sont parfaitement stupides, ce sont seulement les hommes qui les ont rendus intelligents. C'est ce que ce programme a pour but de démentir aux utilisateurs non-initiés:

Si vous n'entrez pas de nom au début du programme, le CPC ou plutôt notre programme devient grincheux et prévient qu'on va casser l'ordinateur à force de faire des erreurs. Mais si les utilisateurs font ce qui leur est demandé, l'ordinateur répond "merci!".

Pendant le déroulement du jeu, le CPC surveille que personne n'essaie de tricher. Si quelqu'un triche, l'ordinateur lance un avertissement et une pénalité.

Non seulement le temps de réaction est affiché précisément, au centième de seconde près, mais le CPC vous dispense également son commentaire personnel sur les performances des divers joueurs ("SUPER!...très faible").

Sur le plan programmation, un jeu d'enfant. Mais ce programme pourra peut-être vous servir "d'excuse" auprès de ceux qui vivent avec vous en leur faisant peut-être entrevoir l'intérêt de l'achat du CPC 464! Qui sait, peut-être pourrez-vous en faire des fans de l'informatique, ce qui serait bien sûr une belle réussite. Ceci dit, voici encore une application possible de ce programme: essayez de mesurer vos temps de réaction quand vous êtes dans des humeurs ou états d'esprit différents!

```
10 REM Reaction
20 REM Programmes Basic Pour le CPC464
30 REM Copyright 1984 DATA BECKER &
      Rainer Luehrs
40 REM Preparatifs
50 INK 0,0:INK 1,24:PEN 1:PAPER 0:MODE 1
60 INPUT"Combien de Participants (1 à 4)"
    " f$
70 IF VAL(f$)>1 OR VAL(f$)>4 THEN GOTO 6
80 FOR n=1 TO VAL(f$)
90 PRINT:PRINT"Votre nom, s'il vous Plait, joueur":n:INPUT f1$(n)
100 REM Reaction si entrée incorrecte
110 IF f1$(n)="" AND marque=0 THEN PRINT
      PRINT"Nevez Pas Peur!
      Chaque chose a son nom.
      Il en va de même Pour les hommes
      et les bêtes, si j'en crois mes circuits
      !" marque=1:GOTO 90
120 IF f1$(n)="" AND marque=1 THEN PRINT
      PRINT"Je suis AMSTRAD, mais vous Pevez
      m'appeler Par mon Prenom: CPC, a
      condition que vous me donniez vot
      re nom!" marque=2:GOTO 90
130 IF f1$(n)="" AND marque=2 THEN PRINT
      PRINT"Faîtes attention: encore une entr
      ee incorrecte et l'ordinateur sera h
      i! " marque=3:GOTO 90
140 IF f1$(n)="" AND marque=3 THEN FOR n
    =1 TO 10000:PRINT CHR$(PEEK(n)):NEXT
150 marque=0
170 NEXT n
180 MODE 8:LOCATE 8,12:PRINT"Merci!"
190 FOR n=1 TO 2000:NEXT n
200 MODE 1:FOR n=1 TO VAL(f$)
210 PRINT f1$(n):"!!!!":PRINT:PRINT:PRIN
      T"C'est à vous!":PRINT"Frappez une touc
      h du clavier des que vous verrez un ch
      angement de couleur."
```

```

220 REM Calcul de l'intervale
230 FOR m=1 TO 2000:NEXT m:a$=1000*RND(1)
FOR n=1 TO a$:a$=INKEY$:IF a$="" THEN NE
XT n:GOSUB 280 ELSE MODE 0:PRINT"Tricheur
!!!!!! Il vous a fallu 5 secondes":f(n)=5
FOR n=1 TO 2000:NEXT n:MODE 1
240 REM Sortir resultat
250 NEXT n:CLS:FOR n=1 TO VAL(f$):PRINT
f1$(n)%;" a mis":f(n);"secondes!":PRINT:N
EXT n:PRINT:PRINT:INPUT"Encore une Partie
(";"":z$:IF LEFT$(UPPER$(z$),1)="n"
THEN END
260 PRINT:INPUT"Avec les memes joueurs
(0/1)":z$:IF LEFT$(UPPER$(z$),1)="0" THE
N GOTO 200 ELSE RUN
270 REM Sous-Programmes de controle
du delai
280 MODE 0:PAPER 1:CLS:b=TIME
290 a$=INKEY$:IF a$="" THEN GOTO 290
300 c=TIME-PAPER 0:MODE 1:f(n)=(c-b)/250
:PRINT f1$(n):PRINT"Youe avez mis":f(n)
:"secondes!"
310 PRINT
320 REM Evaluation
330 IF f(n)<0.005 THEN PRINT"S U P E R"
:GOTO 370
340 IF f(n)<0.1 THEN PRINT"tres bien":GO
TO 370
350 IF f(n)<0.5 THEN PRINT"faible":GOTO
370
360 PRINT"tres tres faible":GOTO 370
370 FOR n=1 TO 2000:NEXT n:CLS:RETURN

```

Conversion numérique

Le CPC n'affiche pas les nombres uniquement selon le système décimal qui nous est si familier puisqu'il dispose également d'instructions spéciales permettant de convertir les nombres dans d'autres systèmes décimaux: AH placé avant un nombre indique qu'il s'agit d'un nombre hexadécimal, AX qu'il s'agit d'un nombre binaire.

Entrez par exemple PRINT AHFF et vous obtiendrez la conversion en décimal de ce nombre hexadécimal: 255.

Si vous entrez maintenant PRINT AX111, vous obtiendrez la conversion en décimal de ce nombre binaire: 7.

Les difficultés commencent quand vous essayez de convertir directement un nombre hexadécimal en un nombre binaire. Il est même tout à fait impossible de travailler avec un système numérique existant autre que les systèmes hexadécimal, décimal ou binaire.

Nous avons donc réalisé ce programme pour vous permettre de vous familiariser avec les systèmes numériques. Vous pouvez convertir tout nombre que vous entrez dans chacun des trois systèmes en faisant précédé le nombre entré de la lettre H pour hexadécimal, B pour binaire, D pour décimal.

Vous pouvez en outre en appuyant sur la touche ENTIE, faire convertir le nombre décimal que vous venez d'entrer en n'importe

quel système numérique (base) entre 2 et 9. Cette conversion se produit d'un façon fort intéressante puisque l'ordinateur calcule devant vous étape par étape le résultat de la conversion.

Comment cela se déroule-t-il? Prenons un exemple: comptons dans le système numérique de base 5. Appuyez sur la touche ENTRÉE jusqu'on vous demande "Déc., Hex. ou Bin.". Répondez 5 à la question suivante et entrez le nombre 230. Ce nombre converti en base 5 sera '1440'.

Voyons maintenant comment l'ordinateur est arrivé à ce résultat: le système numérique de base 5 comporte les chiffres 0, 1, 2, 3 et 4. Le CPC divise le nombre 230 que vous avez entré par 5. Résultat 46, reste 0. Il divise ensuite 46 par 5: résultat 9, reste 1, 9 est à nouveau divisé par 5 car c'est encore un nombre supérieur au plus grand chiffre de ce système numérique. Le résultat est 1 avec un reste de 4. Il suffit maintenant de récapituler les valeurs des restes:

1ère division: 0 2ème division: 1 3ème division: 4

On pourrait effectuer une quatrième division: $1/5 = 0$, reste 1, 1 est donc bien le quatrième reste,

En plaçant les restes de droite à gauche, nous obtenons bien le nombre 1440 comme équivalent en base 5 du nombre décimal (base 10) 230. En voici d'ailleurs la preuve:

$$\begin{array}{l} 1) 0 \cdot 5^0 = 0 \quad 2) 1 \cdot 5^1 = 5 \quad 3) 4 \cdot 5^2 = 100 \quad 4) 1 \cdot 5^3 = 125 \\ 5) 0 + 5 + 100 + 125 = 230 \end{array}$$

Pour être complet nous devons vous préciser qu'il est possible avec le BASIC du CPC d'effectuer, sans notre programme, des conversions directes du système décimal au système hexadécimal ou au système binaire: il faut utiliser pour cela les instructions PRINT HEX\$() et PRINT BIN\$(). Par contre les autres opérations de conversion numérique ne sont possibles qu'avec notre programme.

```

10 REM Convertisseur numerique
20 REM Programmes Basic Pour le CPC464
30 REM Copyright 1984 DATA BECKER S.
40 INK 0,1:INK 1,24:INK 2,1,24:effet=2:n
normal=1
50 ON ERROR GOTO 410
60 MODE 1
70 REM Choix du systeme numerique
80 INPUT "Entree (Dec.,Hex ou Bin.) ";a$
: IF VAL(a$)>65535 THEN GOTO 90 ELSE IF a$="" THEN GOTO 160 ELSE b$=LEFT$(a$,1):b$=UPPER$(b$):IF b$="H" THEN GOSUB 310 ELSE IF b$="B" THEN GOSUB 360 ELSE IF b$="D" THEN a$=RIGHT$(a$,LEN(a$)-1)
90 REM Calcul des nombres decimaux,
binaires et hexadecimaux Pour
la sortie sur ecran
100 PRINT:PRINT "decimal      binaire
hexadecimal"
110 PRINT "-"
120 PRINT a$:TAB(9);";";
130 b$=BIN$(VAL(a$),16):PRINT LEFT$(b$,8);
;"":RIGHT$(b$,8);";";
140 PRINT TAB(32),HEX$(VAL(a$),4)
150 PRINT:PRINT:GOTO 80
160 MODE 1
170 REM Choix d'un systeme numerique
de la base 2 a la base 9
180 INPUT "Quel systeme numerique (2 a 9)
> ";a
190 IF a=0 THEN RUN ELSE IF a<2 OR a>9 T
HEN GOSUB 440:GOTO 160
200 CLS:PRINT "Conversion DEC. - base";a
:PRINT "=====":PRINT
210 REM Calcul des nombres dans le
systeme numerique choisi (2 a 9)
220 INPUT "Votre entree (DEC.) ";b:IF b>
32767 THEN GOSUB 440:GOTO 160 ELSE PRINT
230 c=b:a

```

```

240 PRINT "Nombre: ";c
250 d=b MOD a
260 PRINT TAB(15),"MOD: ";d
270 a$=RIGHT$(STR$(d),1)+a$:
280 IF c<a THEN a$=STR$(c)+a$:PRINT:PRINT
T a$:a$="":PRINT:PRINT:PRINT:GOTO 180
290 b=c:GOTO 230
300 REM Controler et convertir le
nombre hexadecimal entre
310 IF LEN(a$)>5 THEN RETURN
320 b$="&H "+MID$(a$,2,LEN(a$)-1)
330 IF VAL(b$)<0 THEN a$=STR$(VAL(b$))+65
536) ELSE a$=STR$(VAL(b$))
340 RETURN
350 REM Controler et convertir le
nombre binaire entre
360 IF LEN(a$)>17 THEN RETURN
370 b$="&B "+MID$(a$,2,LEN(a$)-1)
380 IF VAL(b$)<0 THEN a$=STR$(VAL(b$))+65
536) ELSE a$=STR$(VAL(b$))
390 RETURN
400 REM Routine de traitement des
erreurs
410 RESUME 420
420 RUN
430 END
440 PEN-effet:PRINT:PRINT TAB(12),"Entre
e incorrecte!"
450 PEN normal:GOSUB 460:RETURN
460 PRINT:PRINT TAB(7),"Appuyez sur une
touche S.V.P."
470 f$=INKEY$:IF f$="" THEN GOTO 470
480 RETURN

```

Les programmes Mémoire 1 à 5 ainsi que le programme de références des variables nous ont déjà permis de pénétrer un peu dans les "secrets internes" du CPC. Par secrets internes nous voulons dire: comment cela se passe-t-il quand un être inhumain tel que l'ordinateur arrive à penser. N'oubliez jamais à cet égard qu'un ordinateur ne peut jamais être plus intelligent qu'il n'a été programmé.

Les programmes Mémoire 1 à 5 ne nous ont cependant pas permis de rentrer suffisamment à l'intérieur de notre CPC. Nous n'avons en effet pu qu'examiner comment nos programmes BASIC sont stockés ligne par ligne, instruction par instruction dans la mémoire de l'ordinateur. Mais nous n'avons pas réussi à savoir comment l'ordinateur traite ces instructions. Nous avons notamment indiqué à cette occasion que le CPC, comme tout autre ordinateur ramène tout à l'alternative la plus simple, éteint ou allumé, oui ou non, 0 ou 1. C'est en effet seulement à partir d'une suite de 0 et de 1 que l'ordinateur arrive à reconnaître un caractère ou une instruction.

Notre BASIC est un langage de programmation évolutif: il est en effet structuré en un nombre important de mots intelligibles (surtout quand on comprend l'anglais). Ces instructions nous permettent d'avoir beaucoup plus facilement une idée de ce que peut faire l'ordinateur que si nous ne pouvions entrer que des 0 et des 1 dans notre ordinateur.

Le code assembleur auquel s'attache ce programme constitue une étape intermédiaire entre le BASIC et le langage binaire de la machine. Nous disposons ici d'un nombre encore plus grand d'instructions (le processeur Z80A du CPC possède plus de 600 instructions) mais ces instructions sont très loin d'être aussi puissantes que les instructions BASIC. Ces instructions occupent 1, 2 ou 3 octets en mémoire. Le programme "Déassembleur" connaît tous les codes assembleur du Z80A (ces codes sont aussi appelés mnémoniques) et il recherche dans la zone que nous voulons quelques codes il rencontre. Il recherche pour cela s'il rencontre des suites de valeurs déterminées (de même que notre programme Mémoire recherchait le mot "Examen"). S'il rencontre ces suites de nombres, il écrit le code assembleur correspondant sur l'écran.

Comme nous ne pensons pas que vous allez vous mettre à écrire du jour au lendemain un grand nombre de programmes en langage-machine, nous avons intégré une fonction pédagogique dans notre "Déassembleur": la ligne 3530 contient en effet une instruction DATA suivie de nombres hexadécimaux qui constituent un petit programme en langage-machine. Ce programme calcule le résultat de deux nombres et le place dans une adresse déterminée. Vous pouvez regarder comment ces nombres se transforment en un programme assembleur en répondant, une fois le programme lancé, à toutes les questions en appuyant simplement sur la touche ENTER.

Vous pouvez aussi écrire d'autres nombres (uniquement en format hexadécimal!) dans cette ligne DATA, par exemple si vous trouvez dans une revue d'informatique un long programme en langage-machine pour le processeur Z80A. Vous pouvez d'ailleurs écrire plusieurs lignes de DATA si nécessaire. Amusez-vous bien!

```

10 REM Desassembler
20 REM Programmes Basic Pour le CPC464
30 REM Copyright 1984 DATA BECKER &
      Rainer Lueers
40 CLEAR
50 REM Liberer la zone memoire a Partir
      de 43776 Pour des routines
      en langage-machine
60 MEMORY 43775:GOSUB 3450
70 DIM i$(255),b1(255):n=0:m=0:sy=1
80 GOTO 1280
90 MODE 2:c=0:PRINT TAB(20)* * * Desass
embleur Z-80 * * *
100 REM Determiner les souhaits de
      l'utilisateur
110 PRINT:INPUT "Titre ";f$
120 IF LEN(f$)>80 THEN GOTO 110
130 PRINT
140 REM Adresse de dePart de la routine
      en langage-machine stockee
      en lignes DATA
150 INPUT "Adresse de dePart (en dec.) "
      :a:IF a<0 OR a>65535 THEN GOTO 130 ELSE
      IF a=0 THEN a=43776
160 INPUT "Adresse de fin (en dec.) "
      b
170 IF b<e OR b>65535 OR b<0 THEN GOTO 1
60 ELSE IF b=0 THEN b=compteur1
180 PRINT:INPUT "Y a-t-il des zones de d
onnees (0/ ) ";v$
190 v$=LEFT$(UPPER$(v$),1):IF v$="0" THE
N GOSUB 2140
200 PRINT:INPUT "Imprimante ou Ecran (I/
E) ";b$
210 REM Determination de la variable
      drubi qui fixe le type de sortie
      avec PRINT:
      8-> imprimante, 0-> ecran
220 b$=LEFT$(UPPER$(b$),1):IF b$="I" THE
N drubi=8 ELSE drubi=0

```

```

230 PRINT #drubi
240 REM Titre apparaissant en milieu
      de ligne
250 PRINT #drubi,TAB((80-LEN(f$))/2-5)f$
260 PRINT #drubi
270 REM Debut du desassembler
      Proprement dit. Lecture
      de la zone de la memoire
      indiquee et interpretation des
      donnees en mnemoniques
      d'instructions en langage-mach.
280 q=a
290 h1=INT(a/4096):h2=INT((a-h1*4096)/25
6)
300 l1=INT((a-h1*4096-h2*256)/16)
310 l2=a-h1*4096-h2*256-11*16
320 IF m<>0 THEN GOTO 340
330 GOTO 390
340 IF a>=a(n) AND a<=b(n) THEN GOTO 360
350 GOTO 390
360 d=PEEK(a):GOSUB 3380:fh$="":f1$=""":q
      h$="":s1$=""":t$=i$(d):i$(d)="DATA"
370 IF a=b(n) THEN n=n+1:m=m-1:GOTO 400
380 GOTO 400
390 d=PEEK(a):GOSUB 650
400 v=d:GOSUB 2740
410 dh$=h$:dl$=1$
420 IF h1>=10 THEN GOTO 560
430 h1$=STR$(h1):h1$=MID$(h1$,2,1)
440 IF h2>=10 THEN GOTO 570
450 h2$=STR$(h2):h2$=MID$(h2$,2,1)
460 IF l1>=10 THEN GOTO 580
470 l1$=STR$(l1):l1$=MID$(l1$,2,1)
480 IF l2>=10 THEN GOTO 590
490 l2$=STR$(l2):l2$=MID$(l2$,2,1)
500 REM Sortie de la zone de la memoire
      desassemblee
510 qt=qt+1:IF qt>105 THEN qt=0:FOR q1=
      TO 9:PRINT #drubi:NEXT q1:GOSUB 3400:sy
      =sy+1:PRINT #drubi,TAB(30) "Page":sy:PRI

```

```

NT #drubi:GOTO 510 ELSE PRINT #drubi,9;L
ENK(dh$+dl$+eh$+el$+fh$+fl$+9h$+9l$)/2;"0
ctets":TAB(18);
520 PRINT #drubi,USING "      ;" ",h1$
,h2$,11$,12$:PRINT #drubi," "
;:PRINT #drubi,USING "!!!!!!!" ;",dhs,
dl$,eh$,el$,fh$,fl$,9h$,9l$:PRINT #drubi,TAB(50),i$(d)
530 a=a+1:c=c+1:IF a>=b+1 THEN GOTO 600
540 IF 9h$="T" THEN i$(d)=t$
550 GOTO 280
560 x=h1-10+65:h1$=CHR$(x):GOTO 440
570 x=h2-10+65:h2$=CHR$(x):GOTO 460
580 x=11-10+65:11$=CHR$(x):GOTO 480
590 x=12-10+65:12$=CHR$(x):GOTO 510
600 PRINT:PRINT "* * Sortie terminee *"
610 REM Demander s'il faut recommencer
620 PRINT:INPUT "Encore une fois (O/ ) "
;b$=LEFT$(UPPER$(b$),1)
630 IF b$="O" THEN RUN
640 END
650 IF d<64 OR d>127 THEN GOTO 710
660 IF d=118 THEN RETURN
670 dh=INT(d/16):dl=d-dh*16
680 9=dl AND 7:f=((d AND 56)/8)
690 i$(d)="LD      "+j$(f)+"," "
700 i$(d)=i$(d)+j$(9):GOSUB 3380:RETURN
710 IF d<128 OR d>191 THEN GOTO 830
720 dh=INT(d/16):dl=d-(dh*16)
730 9=dl AND 7:f=((d AND 120)/8)
740 IF f=0 THEN i$(d)="ADD    A," "
750 IF f=1 THEN i$(d)="ADC    A," "
760 IF f=2 THEN i$(d)="SUB   "
770 IF f=3 THEN i$(d)="SBC    A," "
780 IF f=4 THEN i$(d)="RND   "
790 IF f=5 THEN i$(d)="XOR   "
800 IF f=6 THEN i$(d)="OR    "
810 IF f=7 THEN i$(d)="CP    "
820 i$(d)=i$(d)+j$(9):GOSUB 3380:RETURN

```

```

830 IF b1(d)=1 THEN GOTO 880
840 IF b1(d)=2 THEN GOTO 890
850 IF b1(d)=3 THEN GOTO 900
860 IF b1(d)=4 THEN GOTO 1770
870 IF d=221 OR d=253 THEN GOTO 2220
880 GOSUB 3380:RETURN
890 a=a+1:z=PEEK(a):GOTO 910
900 a=a+1:z=PEEK(a):a=a+1:z1=PEEK(a)
910 v=z:GOSUB 2740:eh$=h$:el$=l$
920 IF b1(d)=3 THEN GOTO 940
930 GOSUB 3390:GOTO 950
940 v=z1:GOSUB 2740:fh$=h$:fl$=l$
950 P1$=fh$+fl$+eh$+el$:P2$=eh$+el$:9h$=
":9l$="""
960 f=d AND 7:9=d RND 56:9=9/8
970 IF f=6 RND (d AND 192)=0 THEN GOTO 1
230
980 IF f=2 AND (d AND 192)=192 THEN GOTO
1260
990 IF f=4 THEN GOTO 1270
1000 IF f=0 RND <9>2 THEN GOTO 1240
1010 i$(33)="LD      HL," +p1$"
1020 i$(34)="LD      (" +p1$+ "),HL"
1030 i$(59)="LD      (" +p1$+ "),A"
1040 i$(205)="CALL   " +p1$"
1050 i$(195)="JP      " +p1$"
1060 i$(58)="LD      A,(" +p1$+ ")"
1070 i$(254)="CP      " +p2$"
1080 i$(42)="LD      HL,(" +p1$+ ")"
1090 i$(49)="LD      SP," +p1$"
1100 i$(17)="LD      DE," +p1$"
1110 i$(16)="DJNZ   " +p2$"
1120 i$(1)="LD      BC," +p1$"
1130 i$(198)="ADD    R," +p2$"
1140 i$(206)="ADC    R," +p2$"
1150 i$(211)="OUT   " +p2$+ ",A"
1160 i$(214)="SUB   " +p2$"
1170 i$(219)="IN      R," +p2$"
1180 i$(222)="SBC   " +p2$"
1190 i$(230)="RND   " +p2$"
1200 i$(238)="XOR   " +p2$"

```

1210 i\$(246)="OR" "+P2\$
 1220 RETURN
 1230 i\$(d)="LD" "+J\$(9)+" "+P2\$:RE
 TURN
 1240 IF 9=3 THEN i\$(d)="JR" "+P2\$:RE
 RETURN
 1250 9=9-4:i\$(d)="JR" "+S\$(9)+" "+P2\$:RETURN
 1260 i\$(d)="JP" "+S\$(9)+" "+P1\$:RE
 TURN
 1270 i\$(d)="CALL" "+S\$(9)+" "+P1\$:RE
 TURN
 1280 FOR s=0 TO 63:b1(s)=1:NEXT s
 1290 FOR s=192 TO 255:b1(s)=1:NEXT s
 1300 b1(118)=1:b1(6)=2:b1(14)=2:b1(16)=2
 :b1(22)=2:b1(24)=2:b1(30)=2:b1(32)=2:b1(38)=2
 :b1(40)=2:b1(46)=2:b1(48)=2:b1(54)=2
 :b1(56)=2
 1310 b1(62)=2:b1(198)=2:b1(206)=2:b1(211)
 :=2:b1(214)=2:b1(219)=2:b1(222)=2:b1(230)
 :=2:b1(238)=2:b1(246)=2:b1(254)=2
 1320 b1(1)=3:b1(17)=3:b1(33)=3:b1(34)=3:
 b1(42)=3:b1(49)=3:b1(50)=3:b1(58)=3:b1(1
 94)=3:b1(195)=3:b1(196)=3:b1(202)=3:b1(2
 04)=3:b1(205)=3:b1(210)=3:b1(212)=3:b1(2
 18)=3:b1(220)=3
 1330 b1(226)=3:b1(228)=3:b1(234)=3:b1(23
 6)=3:b1(242)=3:b1(244)=3:b1(250)=3:b1(25
 2)=3:b1(203)=3:b1(237)=3:b1(221)=0:b1(25
 3)=3
 1340 j\$(0)="B":j\$(1)="C":j\$(2)="D":j\$(3)
 ="E":j\$(4)="H":j\$(5)="L"
 1350 j\$(6)="HL":j\$(7)="R"
 1360 s\$(0)="NZ":s\$(1)="Z":s\$(2)="NC":s\$(
 3)="C":s\$(4)="PO":s\$(5)="PE":s\$(6)="P":s
 \$(7)="M"
 1370 P\$(0)="SBC" HL," :P\$(1)="ADC
 HL,"
 1380 P\$(2)="SBC" HL," :P\$(3)="RDC
 HL,"

1390 P\$(4)="SBC" HL," :P\$(5)="RDC
 HL,"
 1400 P\$(7)="ADC" HL," :n\$(0)="BC":n\$
 (1)="BC":n\$(2)="DE"
 1410 n\$(3)="DE":n\$(4)="HL":n\$(5)="HL":n\$
 (7)="SP"
 1420 I\$(0)="N":I\$(1)="I":m\$(0)="O":m\$(2)
 ="I":m\$(3)="Z"
 1430 o\$(0)="I,A":o\$(1)="R,R":o\$(2)="A,I"
 :o\$(3)="A,R"
 1440 q\$(0)="LD":q\$(1)="CP":q\$(2)="IN":q\$
 (3)="OUT"
 1450 i\$(0)="NOP":i\$(2)="LD" (BC),R
 :"i\$(3)="INC BC"
 1460 i\$(7)="RLC" R":i\$(8)="EX
 AF,AF"
 1470 i\$(9)="ADD" HL,BC":i\$(10)="LD
 A,(BC)"
 1480 i\$(11)="DEC" BC":i\$(15)="RRC
 R"
 1490 i\$(18)="LD" (DE),R":i\$(19)="I
 NC DE"
 1500 i\$(23)="RLA":i\$(25)="ADD" HL,D
 E":i\$(26)="LD" R,(DE)"
 1510 i\$(27)="DEC" DE":i\$(31)="RRR":
 i\$(35)="INC" HL"
 1520 i\$(39)="DAA":i\$(41)="ADD" HL,H
 L":i\$(43)="DEC" HL"
 1530 i\$(47)="CPL":i\$(51)="INC" SP":
 i\$(55)="SCF"
 1540 i\$(57)="ADD" HL,SP":i\$(59)="DE
 C SP":i\$(63)="CCF"
 1550 i\$(43)="DEC" HL"
 1560 i\$(192)="RET" NZ":i\$(197)="PUS
 H BC"
 1570 i\$(199)="RST" O":i\$(200)="RET
 Z":i\$(207)="RST" O"
 1580 i\$(208)="RET" NC":i\$(209)="POP
 DE"
 1590 i\$(213)="PUSH" DE":i\$(215)="RST
 10H"

```

1600 i$(216)="RET      C": i$(223)="RST
    18H"
1610 i$(224)="RET      PO": i$(227)="EX
    (SP),HL"
1620 i$(231)="RST      20H": i$(232)="RE
T      PE"
1630 i$(233)="JP      (HL)": i$(235)="E
X      DE,HL"
1640 i$(239)="RST      28H": i$(240)="RE
T      P"
1650 i$(201)="RET"
1660 i$(241)="POP      RF": i$(243)="DI"
    : i$(245)="PUSH      RF"
1670 i$(247)="RST      30H": i$(248)="RE
T      M": i$(251)="EI"
1680 i$(249)="LD      SP,HL": i$(255)=""
RST      38H"
1690 i$(197)="PUSH      BC"
1700 i$(193)="POP      BC": i$(229)="PUS
H      HL"
1710 i$(225)="POP      HL"
1720 i$(217)="EXX"
1730 i$(118)="HALT"
1740 l=0:FOR d=5 TO 45 STEP 8:i$(d)="DEC
    "+i$(l):l=l+1:NEXT d
1750 l=0:FOR d=4 TO 44 STEP 8:i$(d)="INC
    "+i$(l):l=l+1:NEXT d
1760 i$(60)="INC      A": i$(61)="DEC
A": GOTO 90
1770 IF d=203 THEN GOTO 1790
1780 IF d=237 THEN GOTO 2950
1790 z=a+1:z=PEEK(a)
1800 eh=INT(z/16):el=z-eh*16:g=el AND 7
1810 f=INT((el AND 8)/8):f=f+(eh\2):GOSU
B 1830
1820 GOTO 1980
1830 IF f=0 THEN i$(d)="RLC      "":RETU
RN
1840 IF f=1 THEN i$(d)="RRC      "":RETU
RN

```

```

1850 IF f=2 THEN i$(d)="RL      "":RETU
RN
1860 IF f=3 THEN i$(d)="RR      "":RETU
RN
1870 IF f=4 THEN i$(d)="SLA     "":RETU
RN
1880 IF f=5 THEN i$(d)="SRA     "":RETU
RN
1890 IF f=7 THEN i$(d)="SRL     "":RETU
RN
1900 IF f>8 AND f<=15 THEN GOTO 1940
1910 IF f>=16 AND f<=23 THEN GOTO 1960
1920 f=f AND 7:f$=CHR$(f+48)
1930 i$(d)="SET      "+f$+":":RETURN
1940 f=f AND 7:f$=CHR$(f+48)
1950 i$(d)="BIT      "+f$+":":RETURN
1960 f=f AND 7:f$=CHR$(f+48)
1970 i$(d)="RES      "+f$+":":RETURN
1980 IF eh>=10 THEN GOTO 2030
1990 eh$=STR$(eh):eh$=MID$(eh$,2,1)
2000 IF el>=10 THEN GOTO 2040
2010 el$=STR$(el):el$=MID$(el$,2,1)
2020 GOSUB 3390:i$(d)=i$(d)+j$(9):RETURN
2030 x=eh+55:eh$=CHR$(x):GOTO 2000
2040 x=el+55:el$=CHR$(x):GOTO 2020
2050 IF d AND 7=5 THEN GOTO 2080
2060 IF d AND 7=4 THEN GOTO 2080
2070 GOTO 880
2080 l=d AND 56
2090 g=1/8
2100 IF d AND 7=5 THEN i$(d)="DEC
    "+j$(9):GOTO 2120
2110 i$(d)="INC      "+j$(9)
2120 GOSUB 3380:RETURN
2130 REM Sous-Programme Pour exclure
        certaines zones de la memoire
        du desassemblage
2140 n=1
2150 INPUT "Adresse de dePart=";a(n)
2160 INPUT "Adresse de fin=";b(n)

```

```

2170 IF a(n)>a OR b(n)>b THEN GOTO 2150
2180 INPUT "Encore une zone de donnees (0/1)";v$
2190 v$=LEFT$(UPPER$(v$),1)
2200 IF v$="0" THEN n=n+1:GOTO 2150
2210 n=n:n=1:RETURN
2220 IF d=221 THEN v$="IX":GOTO 2240
2230 v$="IY"
2240 a=a+1:z=PEEK(a)
2250 IF z=203 THEN GOTO 2660
2260 IF z>=70 AND z<=190 THEN GOTO 2810
2270 IF z=33 OR z=34 OR z=42 THEN GOTO 2430
2280 IF z=52 OR z=53 THEN GOTO 2530
2290 IF z=54 THEN GOTO 2590
2300 v=z:GOSUB 2740:eh$=h$:el$=l$:GOSUB 3390
2310 IF z=9 THEN i$(d)="ADD" "+v$+
",BC"
2320 IF z=25 THEN i$(d)="ADD" "+v$+
",DE"
2330 IF z=35 THEN i$(d)="INC" "+v$-
2340 IF z=41 THEN i$(d)="ADD" "+v$-
",+v$"
2350 IF z=43 THEN i$(d)="DEC" "+v$-
2360 IF z=57 THEN i$(d)="ADD" "+v$-
",SP"
2370 IF z=225 THEN i$(d)="POP" "+v$-
2380 IF z=227 THEN i$(d)="EX" (SP) "+v$-
2390 IF z=229 THEN i$(d)="PUSH" "+v$-
2400 IF z=233 THEN i$(d)="JP" ("+"v$+")
2410 IF z=249 THEN i$(d)="LD" SP,"+v$-
2420 RETURN
2430 v=z:GOSUB 2740
2440 eh$=h$:el$=l$:a=a+1
2450 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740
2460 fh$=h$:f1$=l$:a=a+1

```

```

2470 z1=PEEK(a):v=z1:GOSUB 2740
2480 9h$=h$:9l$=l$
2490 IF z=33 THEN i$(d)="LD" "+v$+
",+"+9h$+9l$+fh$+f1$-
2500 IF z=34 THEN i$(d)="LD" ("+"+9h$+9l$+fh$+f1$+")
2510 IF z=42 THEN i$(d)="LD" "+v$+
",("+"+9h$+9l$+fh$+f1$+")
2520 RETURN
2530 v=z:GOSUB 2740:eh$=h$:el$=l$
2540 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740
2550 fh$=h$:f1$=l$:9h$="" :9l$=""
2560 IF z=52 THEN i$(d)="INC" ("+"v$+
"+"+fh$+f1$+")
2570 IF z=53 THEN i$(d)="DEC" ("+"v$+
"+"+fh$+f1$+")
2580 RETURN
2590 v=z:GOSUB 2740:eh$=h$:el$=l$
2600 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740
2610 fh$=h$:f1$=l$:a=a+1
2620 z1=PEEK(a):v=z1:GOSUB 2740
2630 9h$=h$:9l$=l$
2640 i$(d)="LD" ("+"v$+"+"+fh$+f1$+
"),+"+9h$+9l$-
2650 RETURN
2660 a=a+1:eh$=CHR$(67):el$=CHR$(66)
2670 v=PEEK(a):GOSUB 2740
2680 fh$=h$:f1$=l$
2690 a=a+1:z2=PEEK(a):o=z2 AND 248:o=o/8
2700 q$="("+"v$+"+"+fh$+f1$+")
2710 f=o:GOSUB 1830
2720 i$(d)=i$(d)+q$-
2730 v=z2:GOSUB 2740:9h$=h$:9l$=l$:RETURN
2740 h=INT(v/16):l=v-(h*16)
2750 IF h>=10 GOTO 2790
2760 h$=STR$(h):h$=MID$(h$,2,1)
2770 IF l>=10 THEN GOTO 2800
2780 l$=STR$(l):l$=MID$(l$,2,1):RETURN
2790 x1=h+55:h$=CHR$(x1):GOTO 2770

```

```

2800 x1=1+55:1$=CHR$(x1):RETURN
2810 v=z:GOSUB 2740:eh$=h$:el$=1$:gh$=""  
:91$=""
2820 a=a+1:z1=PEEK(a)
2830 v=z1:GOSUB 2740:fh$=h$:f1$=1$
2840 IF z=126 THEN i$(d)="LD      A,<"  
+v$+"+"+fh$+f1$+"")":RETURN
2850 P=z AND 240
2860 IF P=112 THEN GOTO 2900
2870 IF P>=128 THEN GOTO 2920
2880 P=z AND 56:P=P/8:GOSUB 3290
2890 i$(d)="LD      "+9$+",(" "+v$+"+"+fh$+f1$+"")":RETURN
2900 P=z AND 7:GOSUB 3290
2910 i$(d)="LD      (" +v$+"+"+fh$+f1$+"")"  
,"+9$":RETURN
2920 P=z AND 56:P=P/8:GOSUB 3290
2930 i$(d)=i$(d)+v$+"+"+fh$+f1$+"")
2940 RETURN
2950 a=a+1:z=PEEK(a):v=z:GOSUB 2740:eh$=  
h$:el$=1$  
2960 IF z=67 OR z=75 OR z=83 OR z=91 OR  
z=115 OR z=123 THEN GOTO 3200
2970 GOSUB 3390:f=z AND 248:9=z AND 7
2980 IF f=160 THEN GOTO 3160
2990 IF f=168 THEN GOTO 3170
3000 IF f=176 THEN GOTO 3180
3010 IF f=184 THEN GOTO 3190
3020 f=z AND 56:f=f/8:9=z AND 7
3030 IF f=6 THEN i$(d)="SBC      HL,SP"  
:RETURN
3040 IF 9=0 THEN i$(d)="IN      "+j$(f  
)+"<(C)":RETURN
3050 IF 9=1 THEN i$(d)="OUT      (C),"+  
j$(f)":RETURN
3060 IF 9=2 THEN i$(d)=p$(f)+n$(f):RETUR  
N
3070 IF 9=4 THEN i$(d)="NEG":RETURN
3080 IF 9=5 THEN i$(d)="RET"+l$(f):RETUR  
N

```

```

3090 IF 9=6 THEN i$(d)="IM "+m$(f):RETUR  
N
3100 IF 9>7 THEN i$(d)="* * *":RETURN
3110 IF f<=3 THEN GOTO 3150
3120 IF f=4 THEN i$(d)="RRD":RETURN
3130 IF f=5 THEN i$(d)="RLD":RETURN
3140 GOTO 3100
3150 i$(d)="LD      "+o$(f):RETURN
3160 i$(d)=q$(9)+"I":RETURN
3170 i$(d)=q$(9)+"D":RETURN
3180 i$(d)=q$(9)+"IR":RETURN
3190 i$(d)=q$(9)+"DR":RETURN
3200 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740:fh  
$=h$:f1$=1$  
3210 a=a+1:z1=PEEK(a):v=z1:GOSUB 2740:9h  
$=h$:91$=1$  
3220 99$=9h$+91$+fh$+f1$  
3230 IF z=67 THEN i$(d)="LD      (" +99  
$+""),BC":RETURN
3240 IF z=75 THEN i$(d)="LD      BC,<"  
+99$+"")":RETURN
3250 IF z=83 THEN i$(d)="LD      (" +99  
$+""),DE":RETURN
3260 IF z=91 THEN i$(d)="LD      DE,<"  
+99$+"")":RETURN
3270 IF z=115 THEN i$(d)="LD      (" +9  
$+""),SP":RETURN
3280 IF z=123 THEN i$(d)="LD      SP,<"  
+99$+"")":RETURN
3290 IF P=0 THEN 9$="B":i$(d)="ADD  
A,<"  
3300 IF P=1 THEN 9$="C":i$(d)="ADC  
A,<"  
3310 IF P=2 THEN 9$="D":i$(d)="SUB  
"  
3320 IF P=3 THEN 9$="E":i$(d)="SBC  
,  
3330 IF P=4 THEN 9$="H":i$(d)="AND  
"  
3340 IF P=5 THEN 9$="L":i$(d)="XOR  
"

```

```

3350 IF P=6 THEN 9$="***":i$(d)!="OR"
(")
3360 IF P=7 THEN 9$="A":i$(d)="CP"
(")
3370 RETURN
3380 eh$="";e1$="";fh$="";f1$="";gh$="";
91$="";RETURN
3390 fh$="";f1$="";gh$="";91$="";RETURN
3400 PRINT #drubi:RETURN
3410 PRINT #drubi:rt=0:sy=1
3420 PRINT #drubi:RETURN
3430 REM Fin de la bibliotheque de
sous-Programmes Pour creer
le code assembleur
3440 REM Verifier le nombre d'octets
(entrée en hexadecimal) a
desassembler en lignes DATA
3450 ON ERROR GOTO 3540:FOR comPtEUR1=43
776 TO 65535:READ a$:NEXT comPtEUR1
3460 REM Lecture ProPrement dite des
donnees en lignes DATA et
ecriture de ces donnees
a Partir de 43776
3470 RESTORE:FOR m=43776 TO comPtEUR1-1
3480 READ a$
3490 a=VAL("&h"+a$)
3500 POKE m,a
3510 NEXT m:POKE m,8C9
3520 RETURN
3530 DATA 3e,04,06,07,80,32,00,7d
3540 RESUME 3470

```

ANNEXE

Les tokens du BASIC du CPC64 (voir Mémoire 1 à 5)

128 AFTER	129 AUTO	130 BORDER	131 CALL
132 CAT	133 CHAIN	134 CLEAR	135 CLG
136 CLOSEIN	137 CLOSEOUT	138 CLS	139 CONT
140 DATA	141 DEF	142 DEFINT	143 DEFREAL
144 DEFSTR	145 DEG	146 DELETE	147 DIM
148 DRAW	149 DRAWX	150 EDIT	151 ELSE
152 END	153 ENT	154 ENV	155 ERASE
156 ERROR	157 EVERY	158 FOR	159 GOSUB
160 GOTO	161 IF	162 IMR	163 INPUT
164 KEY	165 LET	166 LINE	167 LIST
168 LOAD	169 LOCATE	170 MEMORY	171 MERGE
172 MID\$	173 MODE	174 MOVE	175 MOVE
176 NEXT	177 NEW	178 ON	179 ON BREAK
180 ON ERROR GOTO 0		181 ON SQ	182 OPENIN
183 OPENOUT	184 ORIGIN	185 OUT	186 PAPER
187 PEN	188 PLOT	189 PLOTH	190 POKE
191 PRINT	192 *		
193 BAD	194 RANDOMIZE	195 READ	196 RELEASE
197 REM	198 RENUM	199 RESTORE	200 RESUME
201 RETURN	202 RUN	203 SAVE	204 SOUND
205 SPEED	206 STOP	207 SYMBOL	208 TAG
209 TAGOFF	210 TROFF	211 TRON	212 WAIT
213 WEND	214 WHILE	215 WIDTH	216 WINDOW
217 WRITE	218 ZONE	219 DI	220 EI
221 a 226:	-		
227 ERL	228 FN	229 SPC	230 STEP
231 SWAP	232 et 233:	-	234 TAB
235 THEM	236 TO	237 USING	238 >
239 z	240 > -	241 <	242 <>
243 <-	244 +	245 -	246 *
247 /	248 ^	249 \	250 AND
251 MOD	252 OR	253 XOR	254 NOT

Avec le "prefixe" 255, on a les codes suivants:

0 ABS	1 ASC	2 ATN	3 CHR\$
4 CINT	5 COS	6 CREAL	7 EXP
8 FIX	9 FRE	10 INKEY	11 INP
12 INT	13 JOY	14 LEN	15 LOG
16 LOG10	17 LOWER\$	18 PEEK	19 REMAIN
20 SGN	21 SIN	22 SPACE\$	23 SQ
24 SQR	25 STR\$	26 TAN	27 UNI
28 UPPERS	29 VAL		
30 a 63:	-		
64 EOF	65 ERR	66 HIMEM	67 INKEY\$
68 PI	69 RND	70 TIME	71 XPOS
72 VPOS	73 112:		
113 BIN\$	114 DEC\$	115 HEX\$	116 INSTS\$
117 LEFT\$	118 MAX	119 MIN	120 POS
121 RIGHT\$	122 ROUND	123 STRING\$	124 TEST
125 TEST\$	126		127 VPOS

Achevé d'imprimer en avril 1985
sur les presses de l'imprimerie Laballery et C°
58500 Clamecy
Dépot légal : avril 1985
Numéro d'imprimeur : 504012

AMSTRAD OUVRE-TOI!



LE BASIC DE BASE DES CPC 464

Ce livre est une introduction aux fondamentaux du langage BASIC de vos ordinateurs AMSTRAD CITY 464. Il permet d'apprendre rapidement et facilement le fonctionnement des commandes BASIC tout en apprenant à programmer.

Pour tous les débutants.

Il contient :

- Les fondamentaux de la programmation.
- Les fonctions BASIC.
- Les commandes BASIC.
- Les structures de données.
- Les programmes de tests.

Il existe un deuxième volume qui va plus loin dans l'apprentissage du langage BASIC. Il s'agit du livre "Le BASIC avancé".

Commandez au librairie ou chez votre revendeur Amstrad ou à notre boutique en ligne : www.amstrad.com

Tél. 03 44 77 44 77

Tél. 03 44 77 44 78

MICRO APPLICATION

EDITION VENTIL-MALMAISON
147, av. Paul Desnoyer
Tél. : (0) 732.92.34
Télex : MA.236944F

AMSTRAD

BASIC PLATINUM

Ce livre vous aidera à utiliser votre CPC 464 de manière optimale. Il présente toutes les commandes BASIC et leur fonctionnement. Il explique également les fonctions de votre clavier.

Il existe un deuxième volume qui va plus loin dans l'apprentissage du langage BASIC. Il s'agit du livre "Le BASIC avancé".

Commandez au librairie ou chez votre revendeur Amstrad ou à notre boutique en ligne : www.amstrad.com

Tél. 03 44 77 44 77

Tél. 03 44 77 44 78

LES JEUX D'AVENTURE,

COMBIEN DE POINTS A GAGNER?

Ce livre vous aide à trouver les meilleurs points pour gagner des jeux d'aventure. Il explique comment fonctionnent ces jeux et comment les jouer. Il donne également des conseils pour améliorer vos chances de gagner.

Commandez au librairie ou chez votre revendeur Amstrad ou à notre boutique en ligne : www.amstrad.com

Tél. 03 44 77 44 77

Tél. 03 44 77 44 78

AMSTRAD CITY 464

Ce livre explique à quoi sert le CPC 464. Il présente les fondamentaux de l'ordinateur et ses fonctionnalités. Il explique également comment utiliser les différents logiciels disponibles sur le marché. Il donne également des conseils pour améliorer votre utilisation de l'ordinateur.

Commandez au librairie ou chez votre revendeur Amstrad ou à notre boutique en ligne : www.amstrad.com

Tél. 03 44 77 44 77

Tél. 03 44 77 44 78

PROGRAMME BASIC

Ce livre explique à quoi sert le CPC 464. Il présente les fondamentaux de l'ordinateur et ses fonctionnalités. Il explique également comment utiliser les différents logiciels disponibles sur le marché. Il donne également des conseils pour améliorer votre utilisation de l'ordinateur.

Commandez au librairie ou chez votre revendeur Amstrad ou à notre boutique en ligne : www.amstrad.com

Tél. 03 44 77 44 77

Tél. 03 44 77 44 78

Alimentez votre CPC 464.

Ce livre contient de super programmes :

- Un désassembleur
- Un éditeur graphique
- Un éditeur de textes
- Jeu de réflexion
- Calendrier
- Accent français
- Dump de la mémoire...

ISBN : 2-86899-007-X
PRIX : 129 FF

PRI : 12.17