

LUERS

MICRO APPLICATION

2

AMSTRAD

**PROGRAMMES BASIC
CPC 464**

<http://www.amstradeus.com>



— UN LIVRE DATA BECKER —

Distribué par : EDITIONS RADID
3 rue de L'Éperon
75006 PARIS

et

MICRO APPLICATION
147 av. Paul Doumer
92500 RUEIL-MALMAISON

(c) Reproduction interdite sans l'autorisation de MICRO APPLICATION

"Toute représentation ou reproduction, intégrale ou partielle, faite sans le consentement de MICRO APPLICATION est illicite (loi du 11 mars 1957, alinéa 1er de l'article 40).

Cette représentation ou reproduction illicite, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

La loi du 11 mars 1957 n'autorise, aux termes des alinéas 2 et 3 de l'article 41, que les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à l'utilisation collective d'une part, et d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration".

ISBN 2-86899-007-X

Copyright (c) 1984 DATA BECKER
Marauderstr. 30
4000 Düsseldorf
Allemagne de l'Ouest

Copyright (c) Traduction française 1985 MICRO APPLICATION
147 Avenue Paul Doumer
92500 RUEIL-MALMAISON
FRANCE

Traduit de l'allemand en français par Pascal HAUSMANN

Édité par Frédérique BEAUDONNET
Léo BRITAN
Philippe OLIVIER

Table des matières

Introduction.....	3
I A la découverte de la mémoire	
Mémoire 1.....	5
Mémoire 2.....	8
Mémoire 3.....	10
Mémoire 4.....	15
Mémoire 5.....	23
II Des éditeurs pour vous faciliter le travail	
Editeur graphique.....	27
Editeur sonore.....	37
Editeur de texte.....	48
III Différents jeux de caractères	
Les accents français.....	56
Signes mathématiques.....	61
Ecriture informatique.....	70
IV Programmer plus facilement en BASIC	
Messages d'erreur complète.....	94
Références des variables.....	107

V Des programmes utiles

Calendrier.....	112
Gestion de vos disques ou d'autres données.....	117
Tableaux sportifs.....	129

VI Jeux

Jeu de dés Cameroun.....	143
Décodeur.....	154
Réflexes.....	157

VII Vos premiers pas en langage-machine

Conversion numérique.....	161
Désassembleur.....	166

VIII Annexe

Les tokens du BASIC du CPC 464.....	181
-------------------------------------	-----

Chère lectrice!

Cher lecteur!

Chère programmeuse

Cher programmeur!

Vous avez en main le livre 'Programmes BASIC' et vous voudriez peut-être pouvoir porter un premier jugement sur cet ouvrage. C'est pourquoi vous avez déjà jeté un coup d'œil à la table des matières, après quoi vous vous préparez à sauter l'introduction et à vous diriger immédiatement vers la pièce tranquille où vous vous livrez à la programmation ou à l'écriture de programmes sur votre CPC 464.

J'aimerais toutefois pouvoir dire quelques mots avant que vous ne vous lanciez dans votre occupation favorite.

Le principe qui a en effet présidé à la réalisation de cette collection de programmes est différent de ce qu'on pourrait imaginer au premier abord. Cette collection de programmes ne comprend pas que des programmes mais également 40 pages de textes. La place occupée par ce texte n'a pas pour but de résédier à un manque d'idées: si j'avais publié tous les programmes que j'ai réalisés, cet ouvrage comporterait autour de 400 pages! Le texte est donc là pour que vous ne vous contentiez pas

de taper des programmes à longueur de journée mais pour que vous puissiez aussi apprendre un certain nombre de choses sur le CPC 464 en tapant ces programmes.

Nous n'avons pas expliqué chaque ligne de programme pour ne pas mettre votre patience à trop rude épreuve et parce que nous n'en

avons pas la place, mais nous attirons votre attention dans les textes introductifs sur des détails intéressants et les programmes sont également richement pourvus en lignes REM explicatives.

En ce qui concerne le contenu, c'est surtout le côté pratique pour l'utilisateur qui a été favorisé, soit que vous vouliez éditer de la musique, du graphisme ou des textes, soit que vous ayez besoin de gérer votre stock de disques ou les résultats du championnat de football. Un autre point important de cet ouvrage est enfin de vous permettre d'avoir une idée de la façon dont travaille votre CPC. Le "désassembleur" ou la "liste des références de variables" ne constituent que deux exemples à cet égard.

En tout cas, et ce sera ma conclusion, nous souhaitons que vous aurez beaucoup de plaisir à pénétrer plus profondément dans l'univers des ordinateurs en général et du CPC 464 en particulier et que, peut-être, vous deviendrez grâce à cette collection de programmes un programmeur en herbe.

L'auteur des programmes et du texte vous souhaite encore une fois beaucoup de plaisir dans votre travail avec cet ouvrage.

Mémoire 1

Si nous examinons à l'écran le résultat de ce programme, nous constatons que la zone de la mémoire dont nous demandons le contenu avec l'instruction "PEEK", n'est pas vide, puisqu'elle ne contient pas que des 0.

Ce que nous voyons à l'écran nous semble une suite de chiffres sans signification, mais nous allons pouvoir constater que derrière ces chiffres se cachent aussi des lettres et des caractères très précis. Faisons une petite expérience:

Entrez dans votre CPC: 'PRINT ASC(" ! ")' (ENTER). Résultat: 33. Donc, derrière le signe d'exclamation se cache pour l'ordinateur le nombre 33. Faisons maintenant l'expérience dans l'autre sens; entrez: 'PRINT CHR\$(33)' (ENTER).

Résultat: '!'. Nous avons donc demandé à l'ordinateur de convertir à nouveau le nombre 33 en un caractère. Le résultat est notre signe d'exclamation!

Nous pouvons maintenant apprendre à interpréter de même la masse de chiffres que nous avons obtenue avec le programme Mémoire 1. Pour le vérifier, veuillez entrer à la suite le programme 'Mémoire 2'.

Indication pour l'utilisation du programme 'Mémoire 1':

Eviter de choisir des adresses de début et de fin de la zone de la mémoire distantes de plus de 150 octets, de façon à ce que toutes

les données que vous voulez examiner apparaissent sur une seule page d'écran.

```
10 REM Examen de la memoire 1
20 REM Programmes Basic sur CPC464
30 REM Copyright 1984 DATA BECKER &
   Rainer Lueers
40 INK 0,1:INK 1,24:INK 2,1,24:effet=2:n
   normal=1
50 MODE 1
60 REM Delimiter cadre de l'examen
70 INPUT "Entrer adresse de debut " :a
80 INPUT "Entrer adresse de fin " :e
90 IF e<a OR e>65535 OR a<0 OR a>65535 T
   HEN GOSUB 210:GOTO 50
100 PRINT
110 REM Sortie du contenu de la memoire
   dans le cadre delimite
120 FOR n=a TO e
130 PRINT PEEK(n)
140 NEXT n
150 PRINT
160 PRINT
170 INPUT "Voulez-vous examiner d'autres
   Parties dela memoire ( /N ) " :f$
180 f$=UPPER$(f$)
190 IF f$<>"N" THEN GOTO 10
200 END
210 PEN effet:PRINT:PRINT TAB(12) "Entre
   e incorrecte!"
220 PEN normal:GOSUB 230:RETURN
230 PRINT:PRINT TAB(7) "<Appuyez sur une
   touche S.V.P.>"
240 f$=INKEY$:IF f$="" THEN GOTO 240
250 RETURN
```

Une fois que vous avez tapé ce programme et que vous l'avez lancé avec RUN, vous pouvez constater, suivant ce que vous faites, soit qu'il ne se passe rien, soit qu'il ne se passe que des événements imprévus (l'écran se vide, le mode change, PEN et PAPER se modifient, etc...). Pourquoi?

Comme nous vous l'indiquons également dans d'autres programmes de ce recueil, il y a des caractères directement représentables (de CHR\$(32)=espace à CHR\$(126)=ligne sig-zag et même jusqu'à CHR\$(255) si on compte les caractères graphiques) et des caractères qu'il n'est pas si facile de faire afficher (par exemple CHR\$(2)=déconnexion du curseur texte ou CHR\$(7)=faire entendre un bip, etc...).

Il nous faut donc écrire maintenant encore un troisième programme qui n'affichera à l'écran que les caractères ayant une fonction purement graphique. C'est ainsi que nous aboutissons en complétant un peu notre programme au programme Mémoire 3.

Indication pour l'utilisation du programme Mémoire 2:

Si vous ne voyez à l'écran plus aucune forme reconnaissable et que vous voulez malgré tout continuer à utiliser le programme que vous venez d'entrer, il vous faut faire preuve d'un peu d'astuce: ouvrez le manuel de votre ordinateur au Chapitre 9 et essayez d'annuler les facteurs perturbateurs en entrant PRINT et le code CHR\$(qui convient.

```

10 REM Examen de la memoire 2
20 REM Programmes Basic sur 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 REM Delimiter cadre de l'examen
70 INPUT "Entrez adresse de debut " :a
80 INPUT "Entrez adresse de fin " :e
90 IF e<a OR e>65535 OR a<0 OR a>65535 T
HEN GOSUB 210:GOTO 50
100 PRINT
110 REM Sortie du contenu de la memoire
   dans le cadre delimite
120 FOR n=a TO e
130 PRINT CHR$(PEEK(n));
140 NEXT n
150 PRINT
160 PRINT
170 INPUT "Voulez-vous examiner d'autres
   parties de la memoire ( /N ) " :f$
180 f$=UPPER$(f$)
190 IF f$<>"N" THEN GOTO 10
200 END
210 PEN effet:PRINT:PRINT TAB(12) "Entre
   e incorrecte!"
220 PEN normal:GOSUB 230:RETURN
230 PRINT:PRINT TAB(7) "<Appuyez sur une
   touche S.V.P.>"
240 f$=INKEY$:IF f$="" THEN GOTO 240
250 RETURN

```

En entrant ce programme vous faites d'une pierre deux coups: premièrement l'affichage à l'écran du contenu de la mémoire apparaît sous une forme plus ordonnée (adresse de début de la zone examinée, contenu de cette adresse et des 7 octets suivants avec chaque fois l'affichage CHR\$() correspondant, le tout sur une ligne de l'écran) et d'autre part, seuls sont affichés les caractères CHR\$() représentables (entre 32 = espace et 126 = ligne zig-zag). Les caractères non représentables ont été simplement remplacés par CHR\$(46) = '.'.

Vous pouvez utiliser ce programme de deux façons: si vous entrez simplement l'adresse de début et que vous appuyez ensuite sur la touche ENTER, vous verrez uniquement le contenu des 8 octets situés à partir de cette adresse. Si par contre, vous appuyez, immédiatement après avoir appuyé sur la touche ENTER et pendant que l'affichage à l'écran est en train de se constituer, sur une touche à fonction de répétition, par exemple la touche espace, vous verrez s'afficher à l'écran le contenu des octets suivants, toujours affichés par lignes de 8.

Examinons par exemple où se trouve en mémoire notre programme BASIC Mémoire 3. Sur le CPC comme sur tous les ordinateurs, les instructions ne sont stockées que sous forme d'un code («token») alors que les mots placés à la suite d'une instruction REM sont conservés comme ils ont été écrits. Si vous entrez par exemple 1005 comme adresse de départ, vous devriez maintenant découvrir le mot "octets". Voyez en effet la ligne 140 du listing:

8 octets

Ceci ne fonctionne évidemment que si vous avez bien entré le programme exactement comme il figure sur le listing, sans oublier même un espace.

Si vous entrez 1006 comme adresse de départ et que vous appuyez immédiatement sur la touche espace, vous voyez apparaître la suite de notre programme Mémoire 3.

Indication pour l'utilisation du programme 'Mémoire 3':

Comme vous avez certainement déjà remarqué, les codes correspondant au contenu des zones de mémoire que vous pouvez examiner avec notre programme ne sont pas indiqués avec le système décimal que nous connaissons bien pour l'avoir appris à l'école mais avec le système hexadécimal. Il y a à cela deux raisons: la première est que le système hexadécimal est le système par excellence des fans du langage-machine et de tous ceux qui auscultent la mémoire des ordinateurs; et c'est bien ce que nous essayons de faire! La deuxième raison est qu'on peut placer plus d'informations sur une ligne de 40 caractères avec le système hexadécimal qu'on ne le pourrait avec le système décimal (le nombre décimal de 3 chiffres '255' devient en hexadécimal un nombre de deux chiffres 'FF', de même 65535 (déc.) = FFFF (hexa.).

Si vous ne connaissez pas encore le système hexadécimal, n'ayez aucune crainte, votre CPC vous permet de convertir très aisément les nombres décimaux en nombres hexadécimaux avec l'instruction

PRINT HEX\$(xx). Par exemple PRINT HEX\$(65535) vous donnera: FFFF.

```

10 REM Examen de la memoire 3
20 REM Examen de la memoire avec
    PEEK et CHR$( dans un format
    clair
30 REM Programmes Basic sur le CPC464
40 REM Copyright 1984 DATA BECKER &
    Rainer Lueers
50 INK 0,1:INK 1,24:INK 2,1,24:effet=2:n
    ormal=1
60 KEY DEF 18,1
70 MODE 1
80 REM Delimiter le cadre de l'examen
90 INPUT "Entrer l'adresse de debut ":a
100 IF a<0 OR a>65535 THEN GOSUB 330:GOT
    0 70
110 a$=HEX$(a:4)
120 REM Les lignes suivantes Permettent
    d'obtenir une sortie sur ecran
    optimale avec 40 caracteres Par
    ligne
130 PRINT a$;" ";
140 REM Chaque ligne de l'ecran recoit
    8 octets
150 FOR z=0 TO 7
160 zz=PEEK(a+z):zz$=HEX$(zz,2)
170 PRINT zz$;" ";
180 NEXT z
190 REM Sortie de la chaine de
    caracteres avec un traitement
    Particulier lorsque la valeur
    PEEK est <32 ou >126
200 FOR z=0 TO 7
210 zz=a+z
220 zz=PEEK(zz)
230 IF zz<32 THEN zz=46
240 IF zz>126 THEN zz=46
250 PRINT CHR$(zz);
260 NEXT z
270 PRINT
280 a=a+8
290 a$=INKEY$

```

```

10 REM Examen de la memoire 3
20 REM Examen de la memoire avec
    PEEK et CHR$( dans un format
    clair
30 REM Programmes Basic sur le CPC464
40 REM Copyright 1984 DATA BECKER &
    Rainer Lueers
50 INK 0,1:INK 1,24:INK 2,1,24:effet=2:n
    ormal=1
60 KEY DEF 18,1
70 MODE 1
80 REM Delimiter le cadre de l'examen
90 INPUT "Entrer l'adresse de debut ":a
100 IF a<0 OR a>65535 THEN GOSUB 330:GOT
    0 70
110 a$=HEX$(a:4)
120 REM Les lignes suivantes Permettent
    d'obtenir une sortie sur ecran
    optimale avec 40 caracteres Par
    ligne
130 PRINT a$;" ";
140 REM Chaque ligne de l'ecran recoit
    8 octets
150 FOR z=0 TO 7
160 zz=PEEK(a+z):zz$=HEX$(zz,2)
170 PRINT zz$;" ";
180 NEXT z
190 REM Sortie de la chaine de
    caracteres avec un traitement
    Particulier lorsque la valeur
    PEEK est <32 ou >126
200 FOR z=0 TO 7
210 zz=a+z
220 zz=PEEK(zz)
230 IF zz<32 THEN zz=46
240 IF zz>126 THEN zz=46
250 PRINT CHR$(zz);
260 NEXT z
270 PRINT
280 a=a+8
290 a$=INKEY$

```



```

300 IF f$="" THEN GOTO 90
310 GOTO 110
320 END
330 PEN effet:PRINT:PRINT TAB(12) "Entre
e incorrecte!"
340 PEN normal:GOSUB 350:RETURN
350 PRINT:PRINT TAB(7) "<Appuyez sur une
touche S.V.P.>"
360 f$=INKEY$:IF f$="" THEN GOTO 360
370 RETURN

```

Mémoire 4

Nous avons examiné jusqu'à présent la mémoire de notre ordinateur et nous n'avons trouvé au début que des nombres (Mémoire 1), puis des caractères bizarres (Mémoire 2) mais nous avons enfin obtenu un affichage formaté des nombres et des caractères représentables (Mémoire 3). En examinant avec Mémoire 3 la zone occupée par le programme BASIC, et en comparant l'affichage sur la partie gauche de l'écran à celui de la partie droite, nous avons bien vu que notre programme BASIC ne se compose en fait que de nombres qui sont ensuite convertis par le CPC en diverses lettres ou caractères.

Pour comprendre comment cela est possible, il faut savoir comment l'ordinateur stocke en mémoire les données et les programmes. Pour économiser de la place en mémoire, les instructions sont stockées, après que vous les ayez entrées avec ENTER, sous forme d'abréviations appelées tokens (voir l'annexe). C'est ainsi que l'instruction PRINT qui devrait occuper normalement cinq octets puisqu'elle se compose de 5 lettres n'occupe qu'un octet en mémoire.

Malgré ce que vous écrivez 'PRINT' dans une ligne REM ou entre guillemets, par exemple PRINT"PRINT", le mot PRINT que l'ordinateur ne considérera pas comme une instruction mais comme du texte sera stocké lettre pour lettre et occupera bien 5 octets en mémoire.

Si vous essayez maintenant de rechercher grâce à notre programme

Mémoire 3 a un mot ou un texte quelconque se trouve dans un programme BASIC en mémoire et où il se trouve en mémoire, ceci risque de durer très longtemps jusqu'à ce que vous ayez pu examiner les 65535 octets de la mémoire. Autant chercher une aiguille dans une botte de foin!

Il faut donc procéder de façon plus systématique et c'est ce que permet le programme Mémoire 4! Vous pouvez en effet entrer un mot comprenant jusqu'à 6 lettres et vous indiquez ensuite à l'ordinateur dans quelle zone il doit chercher cette expression. Bien sûr, plus la zone à l'intérieur de laquelle vous lui dites d'effectuer cette recherche est grande et plus vous risquez d'attendre longtemps. Une fois la recherche effectuée, et si l'expression recherchée se trouve bien dans la zone que vous avez délimitée, l'ordinateur vous indique l'adresse où figure cette expression. Vous pouvez alors vous amuser à modifier avec l'instruction POKE votre programme BASIC (employez uniquement des codes compris entre 32 et 126 inclus). Par exemple:

En ligne 10 se trouve le mot 'Examen'. Faites d'abord rechercher à quelle adresse se trouve ce mot. Lancez le programme avec RUN, entrez 'Examen' (ENTER) et faites rechercher ces lettres dans toute la mémoire; entrez N (ENTER) pour répondre à la question suivante. Peu de temps après, vous voyez apparaître à l'écran: 'Examen' trouvé à l'adresse 374. Si vous tombez sur un nombre légèrement différent, ce n'est pas très grave. Appelons la valeur obtenue NOMBRE (pour nous NOMBRE=374). Voyons maintenant ce qui se trouve en mémoire à l'adresse NOMBRE: PRINT PEEK(NOMBRE) (ne tapez pas NOMBRE bien sûr mais la valeur que vous avez obtenue). Résultat: 'E'! Or que signifie le code 69: PRINT CHR\$(69).

Résultat: 'E'!

Le nombre 69 correspond donc bien à la première lettre du mot recherché 'Examen'. Si vous listez la ligne 10, vous y voyez le mot 'Examen'. Si vous placez avec l'instruction POKE une autre valeur que 69 à l'adresse NOMBRE, vous allez changer le mot figurant à la ligne 10. Essayez par exemple de transformer le 'E' en 'G'. Pour savoir quel est le code de la lettre G, utilisez l'instruction PRINT ASC(xx): PRINT ASC("G") => 71. Il vous suffit donc d'entrer POKE NOMBRE,71 et 'Examen' en ligne 10 est devenu 'Gaxmen'!

Voici comment vous pouvez grâce à nos petits programmes en apprendre un peu plus sur le mode de stockage en mémoire des programmes BASIC et avoir une idée peut-être plus précise de la complexité d'un ordinateur tel que le CPC. N'oubliez pas que le BASIC est le produit de nombreux développements et travaux.

Mais notre programme 'Mémoire 4' vous révèle également un autre aspect des programmes BASIC: le stockage du numéro de ligne. Lancez à nouveau le programme avec RUN, entrez le critère de recherche 'Examen' (ou 'Gaxmen' si vous avez modifié le programme) et faites rechercher cette expression uniquement dans le programme en choisissant donc l'option 'P'. Vous verrez peu de temps après non seulement l'adresse correspondante mais aussi le numéro de ligne correspondant dans notre programme BASIC.

Le calcul du numéro de ligne est relativement simple, ... à condition qu'on sache comment l'effectuer. Tout programme BASIC sur le CPC commence normalement en 368. Les adresses 368 et 369

contiennent la longueur de la première ligne de notre programme
 ('PRINT PEEK (368) => 29. PRINT PEEK (369) => 0. Le calcul de la
 longueur peut être obtenu à partir de ces deux valeurs en
 appliquant la formule suivante: $\text{PRINT PEEK}(368)+\text{PEEK}(369)*256 \Rightarrow$
 29. Donc la première ligne BASIC (y compris les deux octets 368
 et 369 qui en indiquent la longueur) est longue de 35 octets. Vous
 pouvez donc, grâce à cette information, calculer le début de la
 seconde ligne BASIC qui contient également les mêmes informations,
 et ainsi de suite. Or les adresses 370 et 371 contiennent le
 numéro de la première ligne BASIC: $\text{PRINT PEEK}(370) \Rightarrow 10$, PRINT
 $\text{PEEK}(371) \Rightarrow 0$. Calculez maintenant la valeur obtenue: PRINT
 $\text{PEEK}(370)+\text{PEEK}(371)*256 \Rightarrow 10$. Donc le premier numéro de ligne de
 notre programme est 10 et cette ligne comporte 35 octets.

Ce sont exactement les mêmes calculs qu'effectue encore et
 toujours (à partir de la ligne 160) le CPC pour pouvoir indiquer
 le numéro de ligne lorsqu'il trouve l'expression que vous lui avez
 fait rechercher.

Mais peut-être en avez-vous à présent assez de la théorie! Passons
 donc au programme 'Mémoire 5'.

Quel(s) caractère(s) cherchez-vous
 (longueur inférieure ou égale à 6) ? Examen

Dans quelle section de la mémoire
 faut-il chercher cette suite
 de caractères

(/H=dans toute la mémoire

/P=dans le programme) ? n

longue trouve à l'adresse 374

Exemple d'utilisation de Mémoire 4

```

10 REM Examen de la memoire 4
20 REM Rechercher en memoire une
   suite de caracteres
30 REM Programmes Basic sur le CPC464
40 REM Copyright 1984 DATA BECKER &
   Rainer Lueers
50 INK 0,1:INK 1,24:INK 2,1,24:effet=2:n
   ormal=1
60 MODE 1
70 REM Entree du critere de recherche
80 INPUT "Quel(s) caractere(s)
   recherchez-vous
   (jusqu'a 6 caracteres)
";f$
90 IF LEN(f$)>6 OR LEN(f$)=0 THEN GOSUB
   540:GOTO 60
100 PRINT
110 REM Choisir zone a faire rechercher
120 INPUT "Dans quelle zone de la memoir
   e doit s'effectuer la recherche
   ( /N=dans toute la memoire/P=
   Programme) ";f1$
130 f1$=UPPER$(f1$)
140 PRINT
150 IF f1$="N" THEN a=0:e=65535:GOTO 370
160 IF f1$="P" THEN a=368:GOTO 180 ELSE
   GOTO 300
170 REM Recherche de la suite de
   caracteres uniquement dans le
   Programme; si trouve ->
   affichage des numeros de ligne
   correspondants
180 aa=PEEK(a)+(PEEK(a+1)*256):bb=PEEK(a
   +2)+(PEEK(a+3)*256)
190 FOR n=a TO a+aa
200 IF CHR$(PEEK(n))=MID$(f$,1,1)THEN k=
   1
210 IF k=1 AND CHR$(PEEK(n+1))=MID$(f$,2
   ,1)THEN k=2
220 IF k=2 AND CHR$(PEEK(n+2))=MID$(f$,3
   ,1)THEN k=3

```

```

230 IF k=3 AND CHR$(PEEK(n+3))=MID$(f$,4
   ,1)THEN k=4
240 IF k=4 AND CHR$(PEEK(n+4))=MID$(f$,5
   ,1)THEN k=5
250 IF k=5 AND CHR$(PEEK(n+5))=MID$(f$,6
   ,1)THEN k=6
260 IF k=LEN(f$) THEN PRINT "trouve ";f$
   ," (";n;") en ligne";bb
270 k=0
280 NEXT n
290 a=a+aa:IF aa=0 OR bb=0 THEN GOTO 470
   ELSE GOTO 180
300 PRINT
310 REM Delimiter la zone a examiner
320 INPUT "Adresse de debut ";a
330 INPUT "Adresse de fin ";e
340 IF a>e OR a<0 OR e<0 OR a>65535 OR e
   >65535 THEN GOSUB 540:GOTO 320
350 PRINT
360 REM Recherche du critere de
   recherche f$ dans la memoire de
   l'adresse 'a' a l'adresse 'e'
370 FOR z=a TO e
380 IF CHR$(PEEK(z))=MID$(f$,1,1)THEN k=
   1
390 IF k=1 AND CHR$(PEEK(z+1))=MID$(f$,2
   ,1)THEN k=2
400 IF k=2 AND CHR$(PEEK(z+2))=MID$(f$,3
   ,1)THEN k=3
410 IF k=3 AND CHR$(PEEK(z+3))=MID$(f$,4
   ,1)THEN k=4
420 IF k=4 AND CHR$(PEEK(z+4))=MID$(f$,5
   ,1)THEN k=5
430 IF k=5 AND CHR$(PEEK(z+5))=MID$(f$,6
   ,1)THEN k=6
440 IF k=LEN(f$) THEN PRINT "trouve ";f$
   ," a l'adresse";z
450 k=0
460 NEXT z
470 PRINT

```

```

480 PRINT
490 REM Fin du Programme ou continuer
500 INPUT "Rechercher d'autres caractere
s ( /N) ",f$
510 f$=UPPER$(f$)
520 IF f$(">"N" THEN GOTO 60
530 END
540 PEN effet:PRINT:PRINT TAB(12) "Entre
e incorrecte!"
550 PEN normal:GOSUB 560:RETURN
560 PRINT:PRINT TAB(7) "<Appuyez sur une
touche S.V.P.>"
570 f$=INKEY$: IF f$="" THEN GOTO 570
580 PRINT:RETURN

```

Mémoire 5

Alors que le programme Mémoire 3 représentait 8 octets par ligne (valeurs décimales et 1 caractère CHR\$), nous allons maintenant avoir un programme plus clair puisqu'il affichera un octet seulement par ligne et fournira plus de renseignements sur chaque octet affiché.

Pour toutes les informations fournies, les codes décimaux et hexadécimaux sont successivement présentés:

- 1) Adresse décimale
- 2) Adresse hexadécimale
- 3) Contenu décimal de l'adresse
- 4) Contenu hexadécimal de l'adresse
- 5) Contenu de l'adresse (si possible) en caractère CHR\$(

Ce programme vous sera très utile dans votre exploration de la mémoire de votre ordinateur car il vous évitera de devoir sans cesse convertir les nombres que vous obtenez. Ce programme vous permettra d'autre part très aisément d'apprendre à connaître le grand nombre de tokens (souvenez-vous: token=code d'une instruction BASIC) dont l'annexe de cet ouvrage vous fournit la liste.

Faisons un essai: ajoutez à votre programme la ligne 9: 9 PRINT (ENTER). Comme vous le savez, la mémoire du programme BASIC commence en 368. Entrons donc 368 comme adresse de début. La valeur finale nous est égale, pourvu qu'elle soit supérieure en

tout cas à l'adresse de début (appuyons donc simplement sur ENTER). L'affichage de la mémoire se déroule jusqu'à l'adresse de début + 2000 = 2368. Pour notre expérience, il nous suffit de faire afficher les 10 premières adresses de la mémoire, donc adresse finale 378. Listez ensuite les ligne 9 et 10 pour contrôler l'affichage.

Voici brièvement comment interpréter l'affichage du contenu de la mémoire:

Adresse 368: contient 6	(368 et 369 donnent la longueur de la ligne)
Adresse 369: contient 0	($\text{Adr } 368 + (256 * \text{Adr } 369) = 6 \text{ octets}$)
Adresse 370: contient 9	(370 et 371 donnent le numéro de la ligne)
Adresse 371: contient 0	($\text{Adr } 370 + (256 * \text{Adr } 371) = \text{numéro } 9$)
Adresse 372: contient 191	(token de l'instruction PRINT)
Adresse 373: contient 0	(Marque de fin de ligne)
Adresse 374: contient 35	(374 et 375 donnent la longueur de la ligne)
Adresse 375: contient 0	($\text{Adr } 374 + (256 * \text{Adr } 375) = 29 \text{ octets}$)
Adresse 376: contient 10	(376 et 377 donnent le numéro de la ligne)
Adresse 377: contient 0	($\text{Adr } 376 + (256 * \text{Adr } 377) = \text{numéro } 10$)
Adresse 378: contient 197	(token de l'instruction REM)

Intéressons-nous à l'adresse 372. Entrez POKE 372,197 (197 est le token de 'REM') et listons la ligne 9. Vous voyez que l'instruction PRINT de la ligne 9 a été remplacée par une instruction REM. Notez simplement pour le moment qu'il n'existe pas seulement des tokens à un octet mais aussi des tokens à deux

octets. Nous n'en dirons pas plus pour le moment mais vous pouvez consulter l'annexe comportant la liste des tokens.

```

10 REM Examen de la memoire 5
20 REM Sortie du contenu de la memoire
   un octet par ligne
30 REM Programmes Basic Pour le CPC464
40 REM Copyright 1984 DATA BECKER &
   Rainer Lueers
50 INK 0:INK 1,24:INK 2,1,24:effet=2:normal=1
60 MODE 1
70 REM Delimiter la zone a examiner
80 INPUT "Adresse de debut " : a
90 REM
100 PRINT "Adresse de fin (Chiffre/<ENTE
R>->" : a+2000 : " ) " : INPUT e
110 IF a<0 OR e<0 OR a>65535 OR e>65535
THEN GOSUB 270:GOTO 60
120 IF e=0 THEN e=a+2000
130 CLS
140 REM Ordre de la sortie:
   Adresse decimale
   Adresse hexadecimale
   Contenu de l'adresse en decimal
   Contenu de l'adresse hexadecimale
   Contenu de l'adresse en CHR$(
150 FOR z=a TO e
160 PRINT z:
170 PRINT TAB(8) HEX$(z,4):
180 PRINT TAB(18) PEEK(z):
190 PRINT TAB(24) HEX$(PEEK(z),2):
200 PRINT TAB(30):
210 n1=PEEK(z)
220 IF n1<32 THEN n1=46
230 IF n1>126 THEN n1=46
240 PRINT CHR$(n1)
250 NEXT z
260 END
270 PEN effet:PRINT:PRINT TAB(12) "Entre
e incorrecte!"
280 PEN normal:GOSUB 290:RETURN
290 PRINT:PRINT TAB(7) "<Appuyez sur une
touche S.V.P.>"
300 f$=INKEY$:IF f$="" THEN GOTO 300
310 RETURN

```

Editeur graphique

Le graphisme est certainement plus simple à programmer sur le CPC464 que sur beaucoup d'autres microordinateurs similaires, mais il reste néanmoins relativement compliqué.

Il n'y a en effet que deux instructions graphiques directes (PLOT pour créer un point et DRAW pour dessiner une ligne) et il faut d'autre part savoir jongler avec les coordonnées si on ne veut pas risquer de ne pas créer ou dessiner exactement les points qu'on aurait souhaité.

Mais le graphisme devient un jeu d'enfant avec le programme d'éditeur graphique que nous vous présentons maintenant.

Nous avons longuement réfléchi avant de décider du mode de commande du curseur graphique qu'il convenait de choisir (il aurait été en effet inutile de réaliser un éditeur graphique, si c'était pour utiliser uniquement l'adressage absolu!). Nous avons éliminé le joystick et les touches curseur parce qu'ils ne nous permettent pas de distinguer plus de 4 directions.

Comme nous voulions avoir un bloc de 8 directions (les 4 points cardinaux plus les 4 diagonales), nous avons simplement réaffecté une grande partie du clavier de façon à disposer d'un bloc de 8 touches correspondant aux 8 directions.

Le programme "Editeur graphique" ne vous permet pas simplement de dessiner dans 8 directions:

'C' vous permet de positionner le curseur de manière absolue (par exemple 320,200 correspond au centre de l'écran).

'D' vous permet de dupliquer une zone de l'écran (vous marquez avec le premier curseur l'angle inférieur gauche de cette zone et avec le second curseur l'angle supérieur droit puis vous marquez avec le troisième curseur l'angle inférieur gauche de la zone où cette zone doit être copiée). La fonction 'OR', au contraire de la fonction 'AND' n'efface pas le fond de l'écran. Vous pouvez également faire afficher l'original agrandi deux fois.

'F' vous permet de choisir parmi les couleurs possibles.

'G' vous permet de tracer une ligne d'une longueur quelconque entre deux points.

'K' vous permet de tracer un cercle autour du curseur qui en sera le centre, après avoir entré le rayon de ce cercle.

'L' vous permet d'effacer une zone que vous définissez par deux positions du curseur (angle inférieur gauche et angle supérieur droit).

'R' vous permet de créer un rectangle que vous définissez par deux positions du curseur (angle inférieur gauche et angle supérieur droit).

'T' vous permet d'afficher un texte à partir de l'emplacement du curseur.

Vous voyez qu'il ne manque à notre éditeur graphique que la possibilité de peindre des zones, de sauvegarder l'image créée sur cassette (une sauvegarde de 16 K dure vraiment trop longtemps, attendons plutôt le lecteur de disquette) et une routine d'impression sur imprimante (hardcopy). Mais toutes les autres possibilités graphiques de votre ordinateur sont mises en oeuvre de façon très simple par cet éditeur graphique.

Bien sûr si vous voulez améliorer ce programme, vous pouvez donner le champ libre à votre imagination créatrice, sous réserve évidemment de ne pas surcharger la capacité mémoire de votre ordinateur.

```

10 REM Editeur graphique
20 REM Programmes Basic Pour le CPC454
30 REM Copyright 1984 DATA BECKER &
    Rainer Luers
40 INK 0,1:INK 1,24:INK 2,1,24:effets2=0
    normal=1
50 MODE 1
60 FOR n=3 TO 20:KEY DEF n/1:NEXT SPEED
    KEY 50,2
70 PAPER 0:PEN normal
80 INPUT "Quel mode ecran ";f$
90 IF VAL(f$)<0 OR VAL(f$)>2 THEN GOSUB
    1570:GOTO 80
100 IF VAL(f$)=0 THEN z1=4
110 IF VAL(f$)=1 THEN z1=2
120 IF VAL(f$)=2 THEN z1=1:mode2=3
130 MODE VAL(f$)
140 REM Fixation des fenetres graphique
    et de texte
150 ORIGIN 0,0,0,640,400,20
160 WINDOW #1,1,40,25,25
170 a=320:b=200
180 PAPER #1,3-mode2:CLS #1
190 PAPER 0:CLG:PLOT a,b,1
200 REM Menu Propose
    Possibilites offertes: CDFGKLRT
    C = Definir coordonnees Curseur
    D = Duplication (Or/And/Arround.)
    F = Fixer la couleur
    G = Dessiner une droite
210 REM K = Dessiner un cercle
    L = Supprimer
    R = Dessiner rectangle
    T = Entrer texte
220 CLS #1:PRINT #1,a,b,"CDFGKLRT";
230 a$=INKEY$
240 IF a$="" THEN GOTO 230
250 GOSUB 1570
260 DRAW a,b
270 IF ASC(a$)>57 THEN GOSUB 290

```

```

280 GOTO 220
290 CLS #1:PRINT #1,a,b;
300 a$=UPPER$(a$)
310 IF a$="C" THEN PRINT #1,"Pl.Curseur"
    :zz=2
320 IF a$="D" THEN PRINT #1,"Duplicati."
    :zz=3
330 IF a$="F" THEN PRINT #1,"Fix. Coule."
    :zz=4
340 IF a$="G" THEN PRINT #1,"Des.Droite"
    :zz=5
350 IF a$="K" THEN PRINT #1,"Des.Cercle"
    :zz=6
360 IF a$="L" THEN PRINT #1,"Supprimer"
    :zz=7
370 IF a$="R" THEN PRINT #1,"Rectangle"
    :zz=8
380 IF a$="T" THEN PRINT #1,"Entr.Texte"
    :zz=9
390 IF zz=0 THEN RETURN
400 a$=INKEY$
410 IF a$="" THEN GOTO 400
420 IF a$=CHR$(13) THEN GOTO 440
430 RETURN
440 ON zz GOSUB 410,470,520,560,900,1000
    ,1080,1280,1450
450 RETURN
460 REM C = Fixer curseur avec coord.
470 CLS #1:INPUT #1,"Ou(x,y)";a$,b$
480 IF VAL(a$)<0 OR VAL(a$)>640 OR VAL(a$)
    <20 OR VAL(a$)>400 THEN GOTO 470
490 a=VAL(a$):b=VAL(b$):PLOT a,b
500 RETURN
510 REM D = Duplication (Or/And/Arround.)
520 Pointa=a:Pointb=b:CLS #1:PRINT #1,"F
    ixer curseur 1"
530 a$=INKEY$
540 IF a$="" THEN GOTO 530
550 couleur=TEST(a,b):Point1=a:Point2=b
560 FOR n=1 TO 10:PLOT a,b,couleur:PLOT
    a,b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:P
    LOT a,b,couleur:NEXT n

```



```

570 GOSUB 1570
580 IF a$=CHR$(13) THEN a1=a:b1=b ELSE G
OTO 530
590 CLS #1:PRINT #1,"Fixer curseur 2"
600 a$=INKEY$
610 IF a$="" THEN GOTO 600
620 couleur=TEST(a,b):Point1=a:Point2=b
630 FOR m=1 TO 10:PLOT a,b,couleur:PLOT
a,b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:P
LOT a,b,couleur:NEXT m
640 GOSUB 1570
650 IF a$=CHR$(13) THEN a2=a:b2=b:IF a2<
=a1 OR b2<=b1 THEN GOTO 590 ELSE GOTO 66
0 ELSE GOTO 600
660 CLS #1:PRINT #1,"Fixer curseur 3"
670 a$=INKEY$
680 IF a$="" THEN GOTO 670
690 couleur=TEST(a,b):Point1=a:Point2=b
700 FOR m=1 TO 10:PLOT a,b,couleur:PLOT
a,b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:P
LOT a,b,couleur:NEXT m
710 GOSUB 1570
720 IF a$=CHR$(13) THEN a3=a:b3=b ELSE G
OTO 670
730 CLS #1:INPUT #1,"O(r) ou N(ot) ";f$:
f$=UPPER$(f$):IF LEFT$(f$,1)="O" THEN fl
a9=1 ELSE fla9=0
740 CLS #1:INPUT #1,"a9randi (O/r) ";f$:
f$=UPPER$(f$):IF LEFT$(f$,1)="O" THEN fl
a92=1 ELSE fla92=0
750 IF fla92<>0 THEN GOTO 800
760 n1=0:m1=0:FOR n=a1 TO a2 STEP z1:FOR
m=b1 TO b2 STEP 2
770 testcouleur=TEST(n,m):IF fla9=0 THEN
PLOT a3+n1,b3+m1,testcouleur ELSE IF TE
ST(a3+n1,b3+m1)=0 THEN PLOT a3+n1,b3+m1,
testcouleur
780 m1=m1+2:NEXT m:m1=0:n1=n1+z1:NEXT n
790 a=Pointa:b=Pointb:PLOT a,b,1:RETURN
800 n1=0:m1=0:FOR n=a1 TO a2 STEP z1:FOR
m=b1 TO b2 STEP 2

```

```

810 ft=TEST(n,m):IF fla9=0 THEN PLOT a3+
n1,b3+m1,ft:PLOT a3+n1,b3+m1+2,ft:PLOT a
3+n1+z1,b3+m1,ft:PLOT a3+n1+z1,b3+m1+2,f
t
820 IF fla9=1 AND TEST(a3+n1,b3+m1)=0 TH
EN PLOT a3+n1,b3+m1,ft:PLOT a3+n1,b3+m1+
2,ft:PLOT a3+n1+z1,b3+m1,ft:PLOT a3+n1+z
1,b3+m1+2,ft
830 m1=m1+4:NEXT m:m1=0:n1=n1+2*z1:NEXT
n
840 a=Pointa:b=Pointb:PLOT a,b,1:RETURN
850 REM F = Fixer couleur
860 CLS #1:INPUT #1,"Quel numero ";f$
870 IF VAL(f$)<0 OR VAL(f$)>z1/2 THEN GO
TO 860 ELSE PLOT a,b,VAL(f$)
880 RETURN
890 REM G = Dessiner droite
900 CLS #1:PRINT #1,"Fixer curseur":Poin
t1=a:Point2=b
910 a$=INKEY$
920 IF a$="" THEN GOTO 910
930 couleur=TEST(a,b)
940 FOR m=1 TO 10:PLOT a,b,couleur:PLOT
a,b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:P
LOT a,b,couleur:NEXT m
950 GOSUB 1570
960 IF a$=CHR$(13) THEN GOTO 980
970 GOTO 910
980 PLOT Point1,Point2,1:DRAW a,b,1:RETU
RN
990 REM K = Dessiner un cercle
1000 CLS #1:INPUT #1,"Rayon ";f$
1010 FOR aa=1 TO 360
1020 DEG
1030 PLOT a+VAL(f$)*COS(aa),b+VAL(f$)*SI
N(aa),1
1040 NEXT aa
1050 PLOT a,b
1060 RETURN
1070 REM L = Supprimer

```



```

1080 CLS #1:PRINT #1,"Fixer curseur 1"
1090 marque1=a:marque2=b
1100 a$=INKEY$
1110 IF a$="" THEN GOTO 1100
1120 couleur=TEST(a,b):Point1=a:Point2=b
1130 FOR m=1 TO 10:PLOT a,b,couleur:PLOT
a,b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:
PLOT a,b,couleur:NEXT m
1140 GOSUB 1570
1150 IF a$=CHR$(13) THEN a1=a:b1=b ELSE
GOTO 1100
1160 CLS #1:PRINT #1,"Fixer curseur 2"
1170 a$=INKEY$
1180 IF a$="" THEN GOTO 1170
1190 couleur=TEST(a,b):Point1=a:Point2=b
1200 FOR m=1 TO 10:PLOT a,b,couleur:PLOT
a,b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:
PLOT a,b,couleur:NEXT m
1210 GOSUB 1570
1220 IF a$=CHR$(13) THEN a2=a:b2=b ELSE
GOTO 1170
1230 ORIGIN 0,0,a1,a2,b2,b1:CLG:
1240 ORIGIN 0,0,0,640,400,20
1250 marque1=a:marque2=b:PLOT a,b,1
1260 RETURN
1270 REM R = Dessiner Rectangle
1280 CLS #1:PRINT #1,"Fixer curseur 1"
1290 marque1=a:marque2=b
1300 a$=INKEY$
1310 IF a$="" THEN GOTO 1300
1320 couleur=TEST(a,b):Point1=a:Point2=b
1330 FOR m=1 TO 10:PLOT a,b,couleur:PLOT
a,b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:
PLOT a,b,couleur:NEXT m
1340 GOSUB 1570
1350 IF a$=CHR$(13) THEN a1=a:b1=b ELSE
GOTO 1300
1360 CLS #1:PRINT #1,"Fixer curseur 2"
1370 a$=INKEY$
1380 IF a$="" THEN GOTO 1370
1390 couleur=TEST(a,b):Point1=a:Point2=b

```

```

1400 FOR m=1 TO 10:PLOT a,b,couleur:PLOT
a,b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:
PLOT a,b,couleur:NEXT m
1410 GOSUB 1570
1420 IF a$=CHR$(13) THEN a2=a:b2=b ELSE
GOTO 1370
1430 PLOT a1,b1:DRAW a2,b1,1:DRAW a2,b2,
1:DRAW a1,b2,1:DRAW a1,b1,1:a=marque1:b=
marque2:PLOT a,b,1:RETURN
1440 REM T = Entrer texte
1450 CLS #1:PRINT #1,"Fixer curseur"
1455 marque1=a:marque2=b
1460 a$=INKEY$
1470 IF a$="" THEN GOTO 1460
1480 couleur=TEST(a,b):Point1=a:Point2=b
1485 FOR m=1 TO 10:PLOT a,b,couleur:PLOT
a,b,1:PLOT a,b,2:PLOT a,b,3:PLOT a,b,1:
PLOT a,b,couleur:NEXT m
1490 GOSUB 1570
1500 IF a$=CHR$(13) THEN a1=a:b1=b:PLOT
a1,b1,1 ELSE GOTO 1460
1510 TAG
1520 CLS #1:INPUT #1,"Texte ":a$
1530 PRINT a$;
1540 a=marque1:b=marque2:PLOT a,b,1
1550 TAGOFF:RETURN
1560 REM Calcul des mouvements du
curseur (8 directions)
1570 IF ASC(a$)=54 THEN a=a+z1
1580 IF ASC(a$)=57 THEN a=a+z1:b=b+z1
1590 IF ASC(a$)=56 THEN b=b+2
1600 IF ASC(a$)=55 THEN a=a-z1:b=b+z1
1610 IF ASC(a$)=52 THEN a=a-z1
1620 IF ASC(a$)=49 THEN a=a-z1:b=b-z1
1630 IF ASC(a$)=50 THEN b=b-2
1640 IF ASC(a$)=51 THEN a=a+z1:b=b-z1
1650 RETURN
1660 a$=INKEY$
1670 IF a$="" THEN 1660
1680 PRINT ASC(a$)
1690 GOTO 1660

```

Le CPC dispose d'un processeur sonore vraiment remarquable mais plus d'un fana de musique aura certainement déjà été découragé par la complexité des instructions musicales et l'apparent illogisme de leur syntaxe. Comment s'expliquer en effet qu'il faille appeler avec l'instruction SOUND les canaux des notes en premier lieu et n'appeler le volume qu'en quatrième position? Comment se rappeler précisément des diverses possibilités des techniques de rendez-vous?

Nous ne pensons pas que l'apprentissage par coeur puisse être ici d'un grand secours mais plutôt que la seule solution est de rechercher sans cesse dans le manuel les informations dont on a besoin.

Où encore: comment se rappeler quelle courbe d'enveloppe ou de volume est actuellement activée et quelle courbe nous pouvons activer? Faut-il faire des petites notes sur papier?

Notre 'éditeur sonore' vous offre une solution bien plus commode. Vous trouverez en effet tous les paramètres avec leur signification en français sur une page de l'écran. Vous pourrez ainsi, simplement en appuyant simultanément sur la touche 'SHIFT' et sur une autre touche, monter (par exemple <SHIFT> 'H' > augmente la hauteur de la note) ou en appuyant simultanément sur la touche 'CTRL' et sur une autre touche, descendre (par exemple <CTRL> 'L' > diminue le volume).

Vous pouvez ainsi produire à l'écran, très tranquillement, n'importe quelle note ou bruit en appuyant ensuite sur la touche ENTER pour activer le son ainsi édité. N'oubliez pas qu'il faut activer, pour produire un son avec notre éditeur sonore, le volume, la hauteur de la note et au moins un canal sonore.

Vous pouvez activer jusqu'à 16 courbes différentes pour le son d'une part et pour le volume d'autre part et vous pouvez utiliser ces courbes lors de l'éditation des notes. En appuyant sur la touche SHIFT et la touche 'N' vous pouvez entrer un numéro ENV et les paramètres correspondants. Le numéro ENV est utilisé pour la sauvegarde et n'a donc pas la même signification que le numéro de la courbe d'enveloppe. Le nombre de paramètres minimum que vous devez entrer est donc dans notre programme non pas de 3 mais de 4.

Vous pouvez entrer les courbes ENT (courbes de hauteur de la note) de la même façon en appuyant simultanément sur les touches 'SHIFT' et 'O'. Vous pouvez consulter les courbes que vous avez réalisées avec <SHIFT> 'P'.

Nous espérons que vous aurez beaucoup de plaisir à éditer et créer de nouveaux sons sur votre CPC!

10 REM Programme sonore
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1984 DATA BECKER &
Rainer Lusers

```
40 INK 0:1:INK 1:24:INK 2:1:24
50 DIM a(15,15),a1(15,15)
60 MODE 1
70 SPEED INK 20:20
80 PEN 1
90 REM Creation image écran
100 PRINT " Le ";PEN 2:PRINT "son";PEN
1:PRINT " sur le CPC464"
110 PRINT
120 PRINT " (SHIFT) -> + (CTRL) ->
-"
130 PRINT
140 PRINT TAB(11) "A: Canal A ";IF a=0 THEN
HEN PRINT "et." ELSE PEN 2:PRINT "all";PEN
1
150 PRINT TAB(11) "B: Canal B ";IF a=0 THEN
HEN PRINT "et." ELSE PEN 2:PRINT "all";PEN
1
160 PRINT TAB(11) "C: Canal C ";IF a=0 THEN
HEN PRINT "et." ELSE PEN 2:PRINT "all";PEN
1
170 PRINT
180 PRINT TAB(4) "D: Rendezvous avec cana
1 A ";IF d=0 THEN PRINT "et." ELSE PEN 2
:PRINT "all";PEN 1
190 PRINT TAB(4) "E: Rendezvous avec cana
1 B ";IF e=0 THEN PRINT "et." ELSE PEN 2
:PRINT "all";PEN 1
200 PRINT TAB(4) "F: Rendezvous avec cana
1 C ";IF f=0 THEN PRINT "et." ELSE PEN 2
:PRINT "all";PEN 1
210 PRINT
220 PRINT TAB(7) "G: Période bruit";IF g
=0 THEN LOCATE 27,13:PRINT "et." ELSE PE
N 2:LOCATE 27,13:PRINT g:PEN 1
230 PRINT
240 PRINT TAB(4) "H: Hauteur note ";IF h
```

```

=0 THEN LOCATE 22,15:PRINT"et." ELSE PEN
2:LOCATE 22,15:PRINT h:PEN 1
250 PRINT
260 PRINT TAB(4) "I:Duree en 1/100 Sec."
  IF i=0 THEN PRINT"et." ELSE PEN 2:
LOCATE 28,21:PRINT i:PEN 1
270 PRINT
280 PRINT TAB(4) "J:Appel d'une courbe E
NV " IF j=0 THEN LOCATE 33,19:PRINT"0"
ELSE PEN 2:LOCATE 32,19:PRINT j:PEN 1
290 PRINT TAB(4) "K:Appel d'une courbe E
NT " IF k=0 THEN LOCATE 33,20:PRINT"0"
ELSE PEN 2:LOCATE 32,20:PRINT j:PEN 1
300 PRINT
310 PRINT TAB(11) "L:Volume " IF l=0 TH
EN LOCATE 25,22:PRINT"et." ELSE PEN 2:LO
CATE 25,22:PRINT l:PEN 1
320 PRINT
330 PRINT TAB(4) "N:Entr. ENV 0:Entr. E
NT P Liste"
340 a$=INKEY$
350 REM Fixer canal son et rendez-vous
360 a1=a+b+c+d+e+f
370 REM Interrogation clavier et
  Production son lorsque <ENTER>
  enfonce
380 IF a$=CHR$(13) THEN SOUND a1,h,i,l,j
,k,9
390 IF a$=CHR$(14) OR a$=CHR$(78) THEN G
OTO 2120
400 IF a$=CHR$(15) OR a$=CHR$(79) THEN G
OTO 2250
410 IF a$=CHR$(16) OR a$=CHR$(80) THEN G
OTO 2370
420 IF a$="" THEN GOTO 340
430 IF ASC(a$)>12 AND ASC(a$)<65 THEN GO
TO 340
440 IF ASC(a$)>76 THEN GOTO 340
450 IF ASC(a$)>12 THEN GOTO 1250
460 REM Modification de l'ecran quand

```

```

une touche a ete actionnee
470 ON ASC(a$) GOSUB 490,530,570,610,650
,690,730,820,910,1000,1080,1160
480 GOTO 340
490 a=0
500 LOCATE 21,5
510 PRINT"et."
520 RETURN
530 b=0
540 LOCATE 21,6
550 PRINT"et."
560 RETURN
570 c=0
580 LOCATE 21,7
590 PRINT"et."
600 RETURN
610 d=0
620 LOCATE 30,9
630 PRINT"et."
640 RETURN
650 e=0
660 LOCATE 30,10
670 PRINT"et."
680 RETURN
690 f=0
700 LOCATE 30,11
710 PRINT"et."
720 RETURN
730 IF g>0 THEN g=g-1
740 IF g=0 THEN LOCATE 27,13:PRINT"et."
RETURN
750 LOCATE 27,13
760 PRINT" "
770 LOCATE 27,13
780 PEN 2
790 PRINT g
800 PEN 1
810 RETURN
820 IF h>0 THEN h=h-1
830 IF h=0 THEN LOCATE 22,15:PRINT"et."
RETURN

```



```

840 LOCATE 22,15
850 PRINT " "
860 LOCATE 22,15
870 PEN 2
880 PRINT h
890 PEN 1
900 RETURN
910 IF i>0 THEN i=i-1
920 IF i=0 THEN LOCATE 28,17 PRINT"et."
RETURN
930 LOCATE 28,17
940 PRINT " "
950 LOCATE 28,17
960 PEN 2
970 PRINT i
980 PEN 1
990 RETURN
1000 IF j>0 THEN j=j-1
1010 LOCATE 32,19
1020 PRINT " "
1030 LOCATE 32,19
1040 IF j>0 THEN PEN 2
1050 PRINT j
1060 PEN 1
1070 RETURN
1080 IF k>0 THEN k=k-1
1090 LOCATE 32,20
1100 PRINT " "
1110 LOCATE 32,20
1120 IF k>0 THEN PEN 2
1130 PRINT k
1140 PEN 1
1150 RETURN
1160 IF l>0 THEN l=l-1
1170 LOCATE 25,22
1180 PRINT " "
1190 LOCATE 25,22
1200 IF l=0 THEN PRINT"et." RETURN
1210 PEN 2
1220 PRINT l
1230 PEN 1

```

```

1240 RETURN
1250 ON (ASC(a$)-64) GOSUB 1270,1330,139
0,1450,1510,1570,1630,1710,1790,1870,195
0,2030
1260 GOTO 340
1270 a=1
1280 LOCATE 21,5
1290 PEN 2
1300 PRINT"a11"
1310 PEN 1
1320 RETURN
1330 b=2
1340 LOCATE 21,6
1350 PEN 2
1360 PRINT"a11"
1370 PEN 1
1380 RETURN
1390 c=4
1400 LOCATE 21,7
1410 PEN 2
1420 PRINT"a11"
1430 PEN 1
1440 RETURN
1450 d=8
1460 LOCATE 30,9
1470 PEN 2
1480 PRINT"a11"
1490 PEN 1
1500 RETURN
1510 e=16
1520 LOCATE 30,10
1530 PEN 2
1540 PRINT"a11"
1550 PEN 1
1560 RETURN
1570 f=32
1580 LOCATE 30,11
1590 PEN 2
1600 PRINT"a11"
1610 PEN 1
1620 RETURN

```



```

1630 IF 9<15 THEN 9=9+1
1640 LOCATE 27,13
1650 PRINT " "
1660 LOCATE 27,13
1670 PEN 2
1680 PRINT 9
1690 PEN 1
1700 RETURN
1710 IF h<4095 THEN h=h+1
1720 LOCATE 22,15
1730 PRINT " "
1740 LOCATE 22,15
1750 PEN 2
1760 PRINT h
1770 PEN 1
1780 RETURN
1790 IF i<32767 THEN i=i+1
1800 LOCATE 28,17
1810 PRINT " "
1820 LOCATE 28,17
1830 PEN 2
1840 PRINT i
1850 PEN 1
1860 RETURN
1870 IF j<15 THEN j=j+1
1880 LOCATE 32,19
1890 PRINT " "
1900 LOCATE 32,19
1910 PEN 2
1920 PRINT j
1930 PEN 1
1940 RETURN
1950 IF k<15 THEN k=k+1
1960 LOCATE 32,20
1970 PRINT " "
1980 LOCATE 32,20
1990 PEN 2
2000 PRINT k
2010 PEN 1
2020 RETURN

```

```

2030 IF l<15 THEN l=l+1
2040 LOCATE 25,22
2050 PRINT " "
2060 LOCATE 25,22
2070 PEN 2
2080 PRINT l
2090 PEN 1
2100 RETURN
2110 REM Creation d'une nouvelle courbe
      ENV
2120 LOCATE 4,24
2130 PRINT " "
2140 LOCATE 4,24
2150 INPUT "No ENV et nombre de Parametre
s":z,z1
2160 IF z<0 OR z>15 OR (z1<>4 AND z1<>7
AND z1<>10 AND z1<>13 AND z1<>16) THEN
GOTO 2120 ELSE LOCATE 4,24:PRINT "
      ":LOCATE 4,2
4
2170 IF z1=4 THEN INPUT "4 Par. ":a(z,0),
a(z,1),a(z,2),a(z,3)
2180 IF z1=7 THEN INPUT "7 Par. ":a(z,0),
a(z,1),a(z,2),a(z,3),a(z,4),a(z,5),a(z,6)
)
2190 IF z1=10 THEN INPUT "10 Par. ":a(z,0)
,a(z,1),a(z,2),a(z,3),a(z,4),a(z,5),a(z
,6),a(z,7),a(z,8),a(z,9)
2200 IF z1=13 THEN INPUT "13 Par. ":a(z,0)
,a(z,1),a(z,2),a(z,3),a(z,4),a(z,5),a(z
,6),a(z,7),a(z,8),a(z,9),a(z,10),a(z,11)
,a(z,12)
2210 IF z1=16 THEN INPUT "16 Par. ":a(z,0)
,a(z,1),a(z,2),a(z,3),a(z,4),a(z,5),a(z
,6),a(z,7),a(z,8),a(z,9),a(z,10),a(z,11)
,a(z,12),a(z,13),a(z,14),a(z,15)
2220 ENV a(z,0),a(z,1),a(z,2),a(z,3),a(z
,4),a(z,5),a(z,6),a(z,7),a(z,8),a(z,9),a
(z,10),a(z,11),a(z,12),a(z,13),a(z,14),a
(z,15)

```

```

2230 GOTO 2370
2240 REM Creation d'une nouvelle courbe
      ENT
2250 LOCATE 4,24
2260 PRINT "
      "
2270 LOCATE 4,24
2280 INPUT "No ENT et nombre de Parametre
s") z,z1
2290 IF z<0 OR z>15 OR (z1<>4 AND z1<>7
AND z1<>10 AND z1<>13 AND z1<>16) THEN
GOTO 2250 ELSE LOCATE 4,24:PRINT "
      "
2300 IF z1=4 THEN INPUT "4 Par. " a1(z,0)
,a1(z,1),a1(z,2),a1(z,3)
2310 IF z1=7 THEN INPUT "7 Par. " a1(z,0)
,a1(z,1),a1(z,2),a1(z,3),a1(z,4),a1(z,5)
,a1(z,6)
2320 IF z1=10 THEN INPUT "10 Par. " a1(z,0)
,a1(z,1),a1(z,2),a1(z,3),a1(z,4),a1(z,5)
,a1(z,6),a1(z,7),a1(z,8),a1(z,9)
2330 IF z1=13 THEN INPUT "13 Par. " a1(z,0)
,a1(z,1),a1(z,2),a1(z,3),a1(z,4),a1(z,5)
,a1(z,6),a1(z,7),a1(z,8),a1(z,9),a1(z,10)
,a1(z,11),a1(z,12)
2340 IF z1=16 THEN INPUT "16 Par. " a1(z,0)
,a1(z,1),a1(z,2),a1(z,3),a1(z,4),a1(z,5)
,a1(z,6),a1(z,7),a1(z,8),a1(z,9),a1(z,10)
,a1(z,11),a1(z,12),a1(z,13),a1(z,14),
a1(z,15)
2350 ENT a1(z,0),a1(z,1),a1(z,2),a1(z,3)
,a1(z,4),a1(z,5),a1(z,6),a1(z,7),a1(z,8)
,a1(z,9),a1(z,10),a1(z,11),a1(z,12),a1(z,13)
,a1(z,14),a1(z,15)
2360 REM Representation sur l'ecran
      des courbes ENV et ENT
      existantes
2370 MODE 1
2380 a$=""

```

```

2390 INPUT "ENV-Liste o u ENT-Liste (V/T
) " a$
2400 IF a$="" THEN GOTO 60
2410 IF a$(">"V" AND a$(">"v" AND a$(">"T"
AND a$(">"t" THEN GOTO 2370
2420 IF a$(">"v" AND a$(">"V" THEN GOTO 25
30
2430 REM Sortie des courbes ENV
2440 FOR n=0 TO 15
2450 FOR m=0 TO 15
2460 IF a(n,0)=0 THEN GOTO 2470 ELSE PRI
NT a(n,m);
2470 NEXT m
2480 PRINT PAPER 3:PRINT n;" "; PAPER 4
2490 NEXT n
2500 a$=INKEY$
2510 IF a$="" THEN GOTO 2580 ELSE GOTO 6
0
2520 REM Sortie des courbes ENT
2530 FOR n=0 TO 15
2540 FOR m=0 TO 15
2550 IF a(n,0)=0 THEN GOTO 2560 ELSE PR
INT a(n,m);
2560 NEXT m
2570 PRINT PAPER 3:PRINT n;" "; PAPER 4
2580 NEXT n
2590 a$=INKEY$
2600 IF a$="" THEN GOTO 2590 ELSE GOTO 6
0

```

Nous n'avons volontairement pas appelé ce programme 'traitement de texte' car nous savions qu'en employant ce terme nous aurions nécessairement déçu l'attente des lecteurs ou programmeurs experts.

Un programme de traitement de texte doit en effet au moins permettre d'insérer des paragraphes supplémentaires dans un texte déjà existant, ce qui est cependant très difficile à réaliser si on ne dispose que de la sauvegarde sur cassette.

Un programme de traitement de texte devrait normalement permettre également un couplage du texte avec un fichier d'adresses (mailing) et permettre également une sauvegarde de section de texte indépendamment de la mémoire disponible, les diverses sections de texte ainsi sauvegardées devraient également pouvoir être affichées à l'écran lorsqu'on les "feuillette".

La sauvegarde sur cassette rend un tel programme de traitement de texte très difficile à réaliser et de toute façon un programme aussi performant et complet dépasserait largement le cadre de cet ouvrage.

Maintenant que nous avons limité votre attente, voici quelles sont les possibilités malgré tout très intéressantes de notre éditeur de texte:

- Vous pouvez stocker jusqu'à 10 pages de 22 lignes dans votre CPC

- Vous pouvez feuilleter votre texte vers le début ou vers la fin du texte en employant les commandes curseur habituelles.
- Vous pouvez écrire des textes ou les placer où vous pensez qu'ils doivent figurer.
- Vous pouvez supprimer des lettres ou des groupes de caractères.
- Si vous avez erré à l'aventure avec le curseur, vous pouvez le replacer dans son emplacement de départ simplement en appuyant sur la touche COPY.
- L'emplacement du curseur est indiqué dans une fenêtre explicative et sa position horizontale est en outre marquée nettement par un trait. Ce "trait de soulignage" est produit dans la fenêtre explicative en mode de réécriture (PRINT CHR\$(22);CHR\$(1)).
- Vous pouvez feuilleter votre texte vers le début en plaçant le curseur en ligne 1 et en appuyant sur la touche de curseur haut.
- Vous pouvez feuilleter votre texte vers la fin soit en plaçant le curseur en ligne 22 et en appuyant sur la touche de curseur bas, soit en appuyant simultanément sur les touches 'CTRL' et 'B'.
- Vous pouvez sauvegarder votre texte en appuyant simultanément sur les touches CTRL et S puis en entrant le nom de fichier souhaité.
- Vous pouvez charger un texte en appuyant simultanément sur les touches CTRL et L.
- Vous pouvez enfin faire rechercher certaines expressions dans votre texte (et faire poursuivre la recherche après qu'une expression ait été trouvée) et faire imprimer votre texte.

La dernière fonction, l'impression du texte que vous avez entré, mérite une explication supplémentaire. Vous avez entré votre texte

dans un format de 40 colonnes. Mais lors de l'impression, votre CPC éditera le texte sur l'imprimante avec le nombre de caractères par ligne que vous choisissez, le minimum étant 40 caractères par ligne.

Encore un conseil:

Si vous ne possédez pas d'imprimante, vous pouvez quand même vous amuser à faire éditer le texte adapté pour l'impression sur l'écran du moniteur, en mode 80 colonnes. Mais vous devez pour cela apporter de légères modifications au programme:

1) Ajoutez la ligne suivante:

```
755 MODE 2
```

2) Remplacez en lignes 780 et 790 les instructions PRINT #8 par de simples PRINT.

3) Il est aussi préférable que le programme attende un peu après avoir édité le texte. Sur l'imprimante, ce n'est pas nécessaire puisque vous pouvez toujours consulter le résultat obtenu sur papier. Vous pourriez donc rajouter en ligne 810 soit une boucle (FOR n=1 TO 10000) avant l'instruction de retour au mode 1 'MODE 1', soit même un message vous demandant si vous voulez continuer (PRINT "FRAPPEZ UNE TOUCHE") suivi d'une interrogation du clavier avec INKEY\$.

Vous pouvez maintenant faire diverses expériences et voir par exemple comment le texte se présente en 80, 70 ou 60 colonnes.

```
10 REM Editeur de texte
20 REM Programmes Basic pour le CPC464
30 REM Copyright 1994 DATA BECKER &
    Rainer Lueers
```

```
40 ON ERROR GOTO 850
50 MODE 1: DIM a$(10,23): page=1: si=1
60 FOR n=1 TO 10
70 FOR m=1 TO 22
80 a$(n,m)=STRING$(40," ")
90 NEXT m,n
100 REM Creation du Premier ecran
110 WINDOW #1,2,39,1,1
120 WINDOW #2,1,40,2,23
130 WINDOW #3,2,39,24,24
140 WINDOW #4,1,40,25,25
150 PAPER #1,3
160 PAPER #3,3
170 PAPER #4,3
180 CLS #1: CLS #2: CLS #3: CLS #4
190 PRINT #1,STRING$(38,CHR$(94));
200 PRINT #3,STRING$(38,CHR$(241));
210 CLS #4: PRINT #4,VPOS(#2);POS(#2)
220 LOCATE #4,15,1: PRINT #4,"(CTRL) + SL
    RIF", LOCATE #4,POS(#2),1: PRINT #4,CHR$(
22);CHR$(1);CHR$(95);CHR$(22);CHR$(0);
230 IF si<page THEN si=page
240 REM Attendre la frappe d'une touche
250 a$=INKEY$
260 IF a$="" THEN GOTO 250
270 GOTO 630
280 GOSUB 330
290 GOSUB 550
300 GOSUB 570
310 PRINT #2,a$: IF a$=CHR$(241) THEN PR
    INT #2
320 GOTO 210
330 REM Editeur ecran:
    reconnaissance des limites,
    Possibilité de feuilleter et
    touche (DEL)
340 IF ASC(a$)<241 AND ASC(a$)>240 AND m
```



```

marque<>1 THEN marque=1:v=VPOS(#2):p=POS(
#2)
350 REM CHR$(224)
    c'est-a-dire: 'COPY' enfoncee
    signifie: ramener curseur au
    point de depart
360 IF ASC(a$)=224 AND marque=1 THEN mar
que=0:LOCATE #2,p,v:a$="" :GOTO 540
370 IF POS(#2)=40 AND VPOS(#2)=22 AND a$
<>CHR$(13) AND ASC(a$)<128 THEN PRINT CH
R$(7):LOCATE #2,1,22:RETURN
380 REM CHR$(127)
    c'est-a-dire: 'DEL' enfoncee
    signifie: supprimer caractere
    a gauche du curseur
390 IF ASC(a$)=127 AND VPOS(#2)<>1 AND P
OS(#2)=1 THEN LOCATE #2,40,VPOS(#2)-1:PR
INT #2," " :LOCATE #2,40,VPOS(#2)-1:a$=""
:GOTO 540
400 IF ASC(a$)=127 AND POS(#2)<>1 THEN L
OCATE #2,POS(#2)-1,VPOS(#2):PRINT #2," "
:LOCATE #2,POS(#2)-1,VPOS(#2):a$="" :GOT
O 540
410 REM CHR$(240)
    c'est-a-dire: fleche 'haut'
    enfoncee
    signifie: curseur une ligne ou
    une page plus haut
420 IF ASC(a$)=240 AND VPOS(#2)<>1 THEN
LOCATE #2,POS(#2),VPOS(#2)-1:a$="" :GOTO
540 ELSE IF ASC(a$)=240 AND VPOS(#2)=1 T
HEN IF Page<>1 THEN Page=Page-1:CLS #2:
FOR n=1 TO 22:PRINT #2,a$(Page,n):NEXT
n:LOCATE #2,1,1
430 REM CHR$(241)
    c'est-a-dire: fleche 'bas'
    enfoncee
    signifie: curseur une ligne ou
    une page plus bas
440 IF ASC(a$)=241 AND VPOS(#2)<>22 THEN
LOCATE #2,POS(#2),VPOS(#2)+1:a$="" :GOTO

```

```

540 ELSE IF ASC(a$)=241 AND VPOS(#2)=22
THEN Page=Page+1:CLS #2:FOR n=1 TO 22:P
RINT #2,a$(Page,n):NEXT n:LOCATE #2,1,1
450 REM CHR$(242)
    c'est-a-dire: fleche 'gauche'
    enfoncee
    signifie: curseur une case vers
    la gauche
460 IF ASC(a$)=242 AND VPOS(#2)<>1 AND P
OS(#2)=1 THEN LOCATE #2,VPOS(#2)-1,40:a$
="" :GOTO 540
470 IF ASC(a$)=242 AND VPOS(#2)<>1 THEN
LOCATE #2,POS(#2)-1,VPOS(#2):a$="" :GOTO
540
480 REM CHR$(243)
    c'est-a-dire: fleche 'droite'
    enfoncee
    signifie: curseur une case vers
    la droite
490 IF ASC(a$)=243 AND VPOS(#2)<>22 AND
POS(#2)=40 THEN LOCATE #2,1,VPOS(#2)+1:a
$="" :GOTO 540
500 IF ASC(a$)=243 AND VPOS(#2)<>22 THEN
LOCATE #2,POS(#2)+1,VPOS(#2):a$="" :GOTO
540
510 IF ASC(a$)<244 AND ASC(a$)>239 THEN
marker=0:a$="" :GOTO 540
520 IF ASC(a$)=224 THEN a$=""
530 IF a$=CHR$(13) THEN a$=CHR$(241)
540 RETURN
550 IF VPOS(#2)=22 AND a$=CHR$(241) THEN
PRINT #2,a$:MID$(a$(Page,VPOS(#2)),POS
(#2)-1,1)=a$:Page=Page+1:a$="" :CLS #2
560 RETURN
570 LOCATE #1,15,1:PRINT #1," Page":Page
580 LOCATE #3,15,1:PRINT #3," Page":Page
+1
590 IF a$<>"" THEN MID$(a$(Page,VPOS(#2)
),POS(#2)-1)=a$

```



```

600 RETURN
610 REM Tester quelle est la touche
    enfoncee
620 REM <CTRL> 'F'
    Femilleter
630 IF ASC(a$)=6 AND Page<>10 THEN Page=
    Page+1:CLS #2:FOR n=1 TO 22:PRINT #2,a$(
    Page/n):NEXT n:LOCATE #2,1,1:GOTO 300
640 REM <CTRL> 'S'
    Sauvegarde sur cassette
650 IF ASC(a$)=19 THEN CLS #4:INPUT #4,"
    Nom " :f$ IF f$="" THEN GOTO 820 ELSE f$=
    "!" + f$:OPENOUT f$:PRINT #9,s1:FOR n=1 TO
    s1:FOR m=1 TO 22:PRINT #9,a$(n,m):NEXT
    m:n:CLOSEOUT:GOTO 710
660 REM <CTRL> 'L'
    Charger a partir de la cassette
670 IF ASC(a$)=12 THEN CLS #4:INPUT #4,"
    Nom " :f$ IF f$="" THEN GOTO 820 ELSE f$=
    "!" + f$:OPENIN f$:INPUT #9,s1:FOR n=1 TO
    s1:FOR m=1 TO 22:INPUT #9,z$(n,m):NEXT m
    n:CLOSEIN:Page=1:CLS #2:FOR n=1 TO 22:P
    RINT #2,a$(Page/n):NEXT n:GOTO 300
680 REM <CTRL> 'R'
    Recherche
690 IF ASC(a$)=18 THEN CLS #4:INPUT #4,"
    Terme a rechercher " :f$ IF f$="" THEN GO
    TO 710 ELSE FOR n=1 TO s1:FOR m=1 TO 22:
    IF INSTR(a$(n,m),f$)=0 THEN CLS #2:FOR n
    m=1 TO 22:PRINT #2,a$(n,m):NEXT nm:Pag
    e=n:LOCATE #2,1,n:GOTO 710 ELSE NEXT m:n
700 REM Si le terme recherche est trouve
    la recherche peut etre
    Poursuivie
705 o$="0"
710 IF ASC(a$)=18 THEN CLS#4:PRINT#4 o$
    : INPUT#4,"Encore(0/)" :f$ f$=UPPER$(f$
    $):IF f$=04 THEN FOR n=Page TO s1:FOR m=
    VPOS(#2)+1 TO 23:IF INSTR(a$(n,m),f$)=0
    THEN CLS#2:FOR nm=1 TO 22:PRINT#2,a$(n,m)
    :NEXT nm:Page=n:LOCATE#2,1,n:GOTO 710

```

```

    ELSE NEXT m:n
720 REM <CTRL> 'I'
    Routine d'impression en 30
    colonnes
730 IF ASC(a$)=9 THEN CLS#4 ELSE GOTO 82
    0
740 INPUT #4,"Imprimante connectee (0/ )
    " :f$ f$=UPPER$(f$):IF LEFT$(f$,1)<>
    "0" THEN GOTO 820
750 CLS#4:INPUT #4,"Caracteres par ligne
    (>=40)" :z1 IF z1<40 THEN GOTO 750
760 a$="" :FOR z2=1 TO s1:FOR z3=1 TO 22:
    FOR z4=1 TO 40
770 IF MID$(a$(z2,z3),z4,1)="" THEN z5=
    LEN(a$) ELSE IF MID$(a$(z2,z3),z4,1)=CHR
    $(241) THEN z4=40:GOTO 790
780 a$=a$+MID$(a$(z2,z3),z4,1) IF LEN(a$
    )=z1 THEN PRINT #9,LEFT$(a$,z5)-a$=RIGHT
    $(a$,LEN(a$)-z5):GOTO 800 ELSE GOTO 800
790 PRINT #9,a$:a$=""
800 IF LEFT$(a$,1)="" THEN a$=RIGHT$(a$
    ,LEN(a$)-1)
810 NEXT z4,z3,z2:MODE 1:GOTO 100
820 IF ASC(a$)<32 AND ASC(a$)>>13 THEN a
    $=""
830 GOTO 200
840 REM Routine de traitement des erreurs
    850 a$="" :RESUME NEXT
860 a$=a$+MID$(a$(z2,z3),z4,1) IF (LEN(a
    $)=z1 AND MID$(a$(z2,z3),z4,1)<>"") THE
    N PRINT LEFT$(a$,z5)-a$=RIGHT$(a$,LEN(a$
    )-z5):GOTO 800 ELSE IF RIGHT$(a$,1)=CHR$
    (241) THEN a$=LEFT$(a$,LEN(a$)-1) :z4=40
    ELSE GOTO 800

```

Si vous placez votre CPC moderne à côté d'une bonne vieille machine à écrire traditionnelle, vous ferez vite un constat agaçant. Vous voyez en effet d'un côté un ordinateur international (ou plutôt anglo-américain) et qui ne dispose donc pas d'accents français ni de la cédille; et de l'autre côté une machine à écrire qui possède bien toutes les lettres dont on a besoin pour écrire en français mais qui -en principe- n'est pas un ordinateur.

Notre programme a donc pour but de hisser votre CPC au niveau de la machine à écrire, au moins en ce qui concerne le clavier: le 'z' se trouve là où il doit être, à côté du 'a' et d'une manière générale toutes les touches se trouvent là où elles figurent sur les machines à écrire française normales. Vous pourrez en outre utiliser les accents et la cédille.

Mais que deviendront les lettres servant à désigner les différentes touches de votre ordinateur? Bien entendu, à moins que vous ne colliez de petites étiquettes sur les touches du clavier, les touches resteront comme avant.

Et que deviendront les caractères que vous pouvez normalement produire avec les touches qui ont maintenant reçu une nouvelle affectation? Une phrase d'exemple en ligne 250 vous permet de voir quels sont les changements effectués. Si vous faites lister cette ligne avant de lancer le programme, vous pourrez voir quels signes ont été remplacés par des accents ou la cédille.

Si vous en avez assez d'utiliser ce jeu de caractères français,

vous pouvez éteindre votre ordinateur ou provoquer un RESET avec les touches SHIFT, CTRL ou ESC.

```

10 REM Jeu de caracteres francais
20 REM Programmes Basic Pour le CPC464
30 REM Copyright 1984 DATA BECKER &
    Rainer Luers
40 SYMBOL AFTER 32
50 REM Lecture de la forme des
    caracteres francais en lignes
    DATA
60 FOR n=1 TO 10
70 READ a
80 FOR n=1 TO 8
90 READ a(n)
100 NEXT n
110 SYMBOL a,a(1),a(2),a(3),a(4),a(5),a(
    6),a(7),a(8)
120 NEXT n
130 REM Nouveau jeu de caracteres
    FRANCAIS
140 KEY DEF 67,1,97,65
150 KEY DEF 59,1,122,90
160 KEY DEF 69,1,113,81
170 KEY DEF 29,1,109,77
180 KEY DEF 71,1,119,87
190 KEY DEF 38,1,58,42
200 KEY DEF 57,1,51,92
210 KEY DEF 24,1,94,126
220 KEY DEF 22,1,64
230 MODE 1
240 REM Phrase d'exemple
250 PRINT "Le Probleme ou Plutôt l'intérêt
    qui naît de toute volonté de créer des
    textes français sur un ordinateur est
    le nombre d'accents propres de notre lan
    gue. . . Eh bien, qui l'a-t-il cru? Il n'y
    a pas (d'endroit où) le bêt blesse pou
    r le CPC!"
260 END
270 REM Lignes DATA avec la forme
    des caracteres francais
280 REM Lettre 'c cedille'
290 DATA 126
    
```

300 DATA &x 00000000
 310 DATA &x 00000000
 320 DATA &x 00111100
 330 DATA &x 01100110
 340 DATA &x 01100000
 350 DATA &x 01100110
 360 DATA &x 00111100
 370 DATA &x 00011000
 380 Lettre a accent grave
 390 DATA 123
 400 DATA &x 01100000
 410 DATA &x 00110000
 420 DATA &x 01111000
 430 DATA &x 00001100
 440 DATA &x 01111100
 450 DATA &x 11001100
 460 DATA &x 01110110
 470 DATA &x 00000000
 480 Lettre o accent circonflexe
 490 DATA 124
 500 DATA &x 00111100
 510 DATA &x 01100110
 520 DATA &x 00111100
 530 DATA &x 01100110
 540 DATA &x 01100110
 550 DATA &x 01100110
 560 DATA &x 00111100
 570 DATA &x 00000000
 580 Lettre u accent grave
 590 DATA 125
 600 DATA &x 00110000
 610 DATA &x 00011000
 620 DATA &x 01100110
 630 DATA &x 01100110
 640 DATA &x 01100110
 650 DATA &x 01100110
 660 DATA &x 00111111
 670 DATA &x 00000000
 680 Lettre e accent grave
 690 DATA 91

700 DATA &x 00110000
 710 DATA &x 00011000
 720 DATA &x 00111100
 730 DATA &x 01100110
 740 DATA &x 01111110
 750 DATA &x 01100000
 760 DATA &x 00111100
 770 DATA &x 00000000
 780 Lettre e accent aigu
 790 DATA 92
 800 DATA &x 00001100
 810 DATA &x 00011000
 820 DATA &x 00111100
 830 DATA &x 01100110
 840 DATA &x 01111110
 850 DATA &x 01100000
 860 DATA &x 00111100
 870 DATA &x 00000000
 880 Lettre e accent circonflexe
 890 DATA 93
 900 DATA &x 00111100
 910 DATA &x 01100110
 920 DATA &x 00111100
 930 DATA &x 01100110
 940 DATA &x 01111110
 950 DATA &x 01100000
 960 DATA &x 00111100
 970 DATA &x 00000000
 980 Lettre a accent circonflexe
 990 DATA 94
 1000 DATA &x 01111000
 1010 DATA &x 11001100
 1020 DATA &x 01111000
 1030 DATA &x 00001100
 1040 DATA &x 01111100
 1050 DATA &x 11001100
 1060 DATA &x 01110110
 1070 DATA &x 00000000
 1080 Lettre u accent circonflexe
 1090 DATA 95

```

1100 DATA &x 00111100
1110 DATA &x 01100110
1120 DATA &x 00000000
1130 DATA &x 01100110
1140 DATA &x 01100110
1150 DATA &x 01100110
1160 DATA &x 00111111
1170 DATA &x 00000000
1180 Lettre i accent circonflexe
1190 DATA 96
1200 DATA &x 00111100
1210 DATA &x 01100110
1220 DATA &x 00011000
1230 DATA &x 00011000
1240 DATA &x 00011000
1250 DATA &x 00011000
1260 DATA &x 00111100
1270 DATA &x 00000000

```

Jeu de signes mathématiques

Ce programme ne comprend que deux sections:

Les lignes 410 à 2560 contiennent en ligne de DATA 24 signes mathématiques couramment utilisés.

Il s'agit exactement des signes suivants, que vous pouvez appeler en appuyant sur la touche CTRL et la touche figurant en regard de chaque signe:

alpha	a	Beta	b	Gamma	c
n'appartient pas à	d	appartient à	e	Intégral	f
Pi	g	Sigma	h	Omega	i
infini	j	angle	k	Triangle	l
Carré	m	Cercle	n	parallèle	o
Rectangle	p	Congru	q	Proche de	r
Ensemble vide	s	1 à 1	t	différent de	u
Val. approchée	v	Proportionnel	w	approché à	x
	y		z		

Nous avons choisi de représenter les formes de ces signes en nombres binaires de façon à ce que vous puissiez aisément les modifier si vous n'êtes pas satisfaite des formes figurant en lignes DATA. Mais faites attention à ne fournir que 8 caractères par ligne (des 0 ou des 1: 0 signifie pixel éteint, 1 signifie pixel allumé).

Chaque signe se compose de 3 lignes de 8 chiffres binaires. Si vous ajoutez à un signe quelconque une ligne supplémentaire, le

jeu des caractères suivants sera complètement modifié (vous pouvez fort bien essayer pour voir ce que cela donne!).

Les lignes 80 à 390 lisent les données avec l'instruction READ. Pour que la représentation binaire des différents signes soit bien interprétée comme une suite de valeurs numériques qui puissent être transformées en nombres décimaux pour leur traitement par l'ordinateur, il faut ajouter à chaque chaîne de 0 et de 1 les caractères '&x' qui désignent les nombres binaires.

Après avoir lancé le programme avec 'RUN', vous pouvez afficher à l'écran les signes mathématiques nouvellement créés en appuyant sur la touche CTRL et sur la touche correspondant au signe voulu dans la table de la page précédente.

Pour que le jeu normal de caractères ASCII puisse continuer à être utilisé, nous avons affecté nos signes mathématiques aux codes 1 à 26 qui ne correspondent pas normalement à des caractères affichables. Nous n'avons cependant pas modifié le code ASCII 13 qui correspond à la fonction de la touche ENTER ni le code ASCII 16 qui correspond à la fonction de la touche DEL.

```

10 REM Jeu de caracteres mathematiques
20 REM Programmes Basic Pour le CPC464
30 REM Copyright DATA BECKER &
    Rainer Lueers
40 RESTORE
50 SYMBOL AFTER 1
60 REM Reaffectation des
    caracteres ASCII 1 a 12
70 REM Pouvant etre appeles
    avec <CTRL> et la touche
80 FOR m=1 TO 12
90 FOR n=1 TO 8
100 READ a$
110 REM Conversion du nom de la chaine
    en nombre binaire puis decimal
120 a$(n)="%x "+a$
130 a(n)=VAL("%x "+a$)
140 NEXT n
150 SYMBOL m,a(1),a(2),a(3),a(4),a(5),a(
6),a(7),a(8)
160 NEXT m
170 REM Reaffectation des
    caracteres ASCII 14 a 15
180 REM Pouvant etre appeles
    avec <CTRL> et la touche
190 FOR m=14 TO 15
200 FOR n=1 TO 8
210 READ a$
220 REM Conversion du nom de la chaine
    en nombre binaire puis decimal
230 a$(n)="%x "+a$
240 a(n)=VAL("%x "+a$)
250 NEXT n
260 SYMBOL m,a(1),a(2),a(3),a(4),a(5),a(
6),a(7),a(8)
270 NEXT m
280 REM Reaffectation des
    caracteres ASCII 17 a 26
290 REM Pouvant etre appeles
    avec <CTRL> et la touche

```

```

300 FOR m=17 TO 26
310 FOR n=1 TO 8
320 READ a$
330 REM Conversion du nom de la chaine
    en nombre binaire Puis decimal
340 a$(n)="%x "+a$
350 a(n)=VAL("%x "+a$)
360 NEXT n
370 SYMBOL m,a(1),a(2),a(3),a(4),a(5),a(
6),a(7),a(8)
380 NEXT m
390 END
400 REM Voici la representation binaire
    du jeu de caracteres mathema-
    tiques.
    Chaque caractere est precede
    de la signification du caractere
    en ligne REM
410 REM Alpha
420 DATA 00000000
430 DATA 00011100
440 DATA 00100100
450 DATA 01000100
460 DATA 01000100
470 DATA 00100100
480 DATA 00011100
490 DATA 00000010
500 REM Beta
510 DATA 00000000
520 DATA 01111000
530 DATA 01000100
540 DATA 01011000
550 DATA 01000100
560 DATA 01000100
570 DATA 01111000
580 DATA 10000000
590 REM Gamma
600 DATA 00000000
610 DATA 10000010
620 DATA 01000100

```

```

630 DATA 00101000
640 DATA 00010000
650 DATA 00010000
660 DATA 00010000
670 DATA 00010000
680 REM n'appartient Pas a
690 DATA 00000000
700 DATA 00000100
710 DATA 00011110
720 DATA 00100100
730 DATA 00111110
740 DATA 00100100
750 DATA 00011110
760 DATA 00000100
770 REM appartient a
780 DATA 00000000
790 DATA 00000000
800 DATA 00011110
810 DATA 00100000
820 DATA 00111110
830 DATA 00100000
840 DATA 00011110
850 DATA 00000000
860 REM Entier
870 DATA 00110000
880 DATA 00101000
890 DATA 00101000
900 DATA 00100000
910 DATA 00100000
920 DATA 10100000
930 DATA 10100000
940 DATA 01100000
950 REM Pi
960 DATA 00000000
970 DATA 00000000
980 DATA 01111110
990 DATA 00100100
1000 DATA 00100100
1010 DATA 00100100
1020 DATA 00100100

```

1030 DATA 00000000
 1040 REM Sigma
 1050 DATA 00000000
 1060 DATA 01111111
 1070 DATA 00100001
 1080 DATA 00010000
 1090 DATA 00010000
 1100 DATA 00100001
 1110 DATA 01111111
 1120 DATA 00000000
 1130 REM Omega
 1140 DATA 00000000
 1150 DATA 00111100
 1160 DATA 00100100
 1170 DATA 01000010
 1180 DATA 01000010
 1190 DATA 00100100
 1200 DATA 11100111
 1210 DATA 00000000
 1220 REM infini
 1230 DATA 00000000
 1240 DATA 00000000
 1250 DATA 01101100
 1260 DATA 10010010
 1270 DATA 10010010
 1280 DATA 10010010
 1290 DATA 01101100
 1300 DATA 00000000
 1310 DATA 00000000
 1320 REM angle
 1330 DATA 00000010
 1340 DATA 00000100
 1350 DATA 00001000
 1360 DATA 00010000
 1370 DATA 00100000
 1380 DATA 01000000
 1390 DATA 11111110
 1400 REM triangle
 1410 DATA 00000000
 1420 DATA 00000000

1430 DATA 00000000
 1440 DATA 00010000
 1450 DATA 00101000
 1460 DATA 01000100
 1470 DATA 11111110
 1480 DATA 00000000
 1490 REM carre
 1500 DATA 11111110
 1510 DATA 10000010
 1520 DATA 10000010
 1530 DATA 10000010
 1540 DATA 10000010
 1550 DATA 10000010
 1560 DATA 11111110
 1570 DATA 00000000
 1580 REM cercle
 1590 DATA 00000000
 1600 DATA 01111100
 1610 DATA 10000010
 1620 DATA 10000010
 1630 DATA 10000010
 1640 DATA 10000010
 1650 DATA 01111100
 1660 DATA 00000000
 1670 REM parallele a
 1680 DATA 01001000
 1690 DATA 01001000
 1700 DATA 01001000
 1710 DATA 01001000
 1720 DATA 01001000
 1730 DATA 01001000
 1740 DATA 01001000
 1750 DATA 01001000
 1760 REM angle droit de
 1770 DATA 00010000
 1780 DATA 00010000
 1790 DATA 00010000
 1800 DATA 00010000
 1810 DATA 00010000
 1820 DATA 00010000

```

1830 DATA 11111110
1840 DATA 00000000
1850 REM congruent
1860 DATA 01100000
1870 DATA 10011001
1880 DATA 00000110
1890 DATA 00000000
1900 DATA 11111111
1910 DATA 00000000
1920 DATA 11111111
1930 DATA 00000000
1940 REM Peu different, Proportionnel
1950 DATA 00000000
1960 DATA 00000000
1970 DATA 00000000
1980 DATA 01100000
1990 DATA 10011001
2000 DATA 00000110
2010 DATA 00000000
2020 DATA 00000000
2030 REM ensemble vide
2040 DATA 00000010
2050 DATA 01111100
2060 DATA 10001010
2070 DATA 10010010
2080 DATA 10100010
2090 DATA 01111100
2100 DATA 10000000
2110 DATA 00000000
2120 REM 1 Pour 1
2130 DATA 00000000
2140 DATA 00100100
2150 DATA 01000010
2160 DATA 11111111
2170 DATA 01000010
2180 DATA 00100100
2190 DATA 00000000
2200 DATA 00000000
2210 REM different
2220 DATA 00000001

```

```

2230 DATA 00000010
2240 DATA 11111111
2250 DATA 00001000
2260 DATA 11111111
2270 DATA 00100000
2280 DATA 01000000
2290 DATA 00000000
2300 REM approche
2310 DATA 00000000
2320 DATA 01100000
2330 DATA 10011001
2340 DATA 00000110
2350 DATA 01100000
2360 DATA 10011001
2370 DATA 00000110
2380 DATA 00000000
2390 REM Proportionnel a
2400 DATA 00000000
2410 DATA 00000000
2420 DATA 01101111
2430 DATA 10001000
2440 DATA 10001000
2450 DATA 10001000
2460 DATA 01101111
2470 DATA 00000000
2480 REM valeur approchée de ...
2490 DATA 00000000
2500 DATA 00000100
2510 DATA 00000010
2520 DATA 11111111
2530 DATA 00000010
2540 DATA 00000100
2550 DATA 00000000
2560 DATA 00000000

```

Il faut bien reconnaître que le jeu de caractères du CPC est très esthétique. Bien que les différentes lettres ne soient dessinées, comme sur d'autres ordinateurs, que dans une matrice de 8*8 (5 colonnes de 8 points allumés ou éteints), la qualité du dessin des différentes lettres rappelle vraiment la machine à écrire.

Pourquoi vouloir donc créer un nouveau jeu de caractères? Eh bien, cela fait maintenant plusieurs années que dure la grande mode des jeux électroniques. Et pour ces jeux futuristes, il est clair que ce n'est pas de lettres classiques mais bien de lettres et de symboles informatiques qu'on veut pouvoir se servir!

Le programme comprend deux parties principales:

- 1) Un océan de lignes de DATA comportant le jeu de caractères informatiques
- 2) La routine de traitement et de chargement des DATA.

Quelques précisions sur la deuxième partie du programme: le CPC présente tout (y compris les lettres et autres caractères) à l'écran sous une forme graphique. Il est donc, contrairement à ce qui est le cas sur les autres ordinateurs, relativement facile, non seulement de mélanger à l'écran texte et graphisme, mais aussi de représenter la même lettre plusieurs fois avec des formes différentes. Nous avons utilisé cette possibilité dans notre programme qui vous permet soit de transformer le jeu de caractères complet dans écriture informatique, soit de travailler simultanément avec les jeux de caractères normaux et

informatiques.

Si vous connaissez la puissante instruction SYMBOL, vous comprendrez certainement comment nous avons pu modifier complètement le jeu de caractères de votre ordinateur.

Mais comment peut-on conserver en mémoire deux jeux de caractères simultanément? Il faut pour cela que le programme continue de tourner car il n'est pas si facile d'appeler simultanément autant de caractères avec les touches du clavier. C'est pourquoi nous avons placé dans le programme un commutateur entre les deux jeux de caractères: si vous appuyez sur TAB vous passez de l'écriture normale à l'écriture informatique ou vice versa.

Voici la technique de programmation que nous avons utilisée: le code ASCII du caractère voulu est interprété (par exemple 'I'=33); puis le programme cherche la forme de caractère informatique correspondante et... celle-ci n'est pas placée avec l'instruction SYMBOL à la place de la forme normale du caractère choisi (sinon il ne serait plus possible de revenir au jeu de caractères normal)... mais elle est placée dans un caractère inutilisé (CHR\$(255)). C'est donc toujours ce même caractère qui est affiché, chaque fois avec une forme différente, lorsque vous affichez un caractère informatique.

```

10 REM Ecriture informatique
20 REM Programmes Basic Pour le CPC464
30 REM Copyright 1984 DATA BECKER &
    Rainer Lueers
40 MODE 1
50 DIM a(122,8)
60 REM liberer la zone de caracteres
    a redéfinir
70 REM Lecture de écriture informatique
    avec la variable a
80 SYMBOL AFTER 32
90 REM Charger provisoirement le code
    et la forme du caractère dans
    variable a
100 FOR n=1 TO 73
110 READ a
120 FOR m=1 TO 8
130 READ a$
140 REM Conversion des chaînes en
    nombres binaires puis en
    nombres décimaux
150 a$="&x "+a$
160 a(a,m)=VAL(a$)
170 NEXT m
180 NEXT n
190 REM Options Possibles:
    1) entrer un texte en écriture
        informatique ou normale
        pendant le cours du Programme
195 REM 2) transformer le jeu de
    caracteres complet en
    écriture informatique
200 PRINT "Entrez un texte quelconque"
    Si vous voulez qu'il apparaisse
    e en écriture informatique, appuyez
    d'abord sur la touche TAB.
    Si vous voulez que tous les ca
    racteres"

```

```

210 PRINT "apparaissent maintenant en écriture
    informatique (y compris le lis
    tin)", appuyez maintenant sur la touc
    he o"; a$="": INPUT a$
220 a$=UPPER$(a$); IF a$="0" THEN GOTO 51
    0
230 REM Le jeu de caracteres normal est
    conserve et les lettres ne sont convertie
    s en écriture informatique que une Pa
    r une a votre demande
240 PRINT
250 PRINT "Veuillez entrer du texte"
260 PRINT
270 a$=INKEY$
280 IF a$="" THEN GOTO 270
290 REM Petit editeur de texte:
    avec interpretation des
    touches <ENTER>=13, <TAB>=9 et
    <DELETE>=127
300 IF ASC(a$)=9 THEN GOTO 360
310 IF ASC(a$)=13 THEN PRINT:GOTO 270
320 IF ASC(a$)=127 THEN a$="": IF POS(#0)
    >1 THEN LOCATE POS(#0)-1,VPOS(#0):PRINT
    " "; LOCATE POS(#0)-1,VPOS(#0):GOTO 270
330 PRINT a$;
340 GOTO 270
350 END
360 a$=INKEY$
370 IF a$="" THEN GOTO 360
380 REM Petit editeur de texte:
    avec interpretation des
    touches <ENTER>=13, <TAB>=9 et
    <DELETE>=127
390 IF ASC(a$)=9 THEN GOTO 270
400 IF ASC(a$)=13 THEN PRINT:GOTO 360
410 IF ASC(a$)=127 THEN a$="": IF POS(#0)
    >1 THEN LOCATE POS(#0)-1,VPOS(#0):PRINT
    " "; LOCATE POS(#0)-1,VPOS(#0):GOTO 360
    ELSE GOTO 360
420 IF ASC(a$)>122 THEN PRINT a$;GOTO 3
    60

```

```

430 IF a(ASC(a$),1)=0 AND a(ASC(a$),2)=0
AND a(ASC(a$),3)=0 AND a(ASC(a$),4)=0 A
ND a(ASC(a$),5)=0 AND a(ASC(a$),6)=0 AND
a(ASC(a$),7)=0 AND a(ASC(a$),8)=0 THEN
PRINT a$;GOTO 360
440 REM Pour que le jeu de caracteres
normal ne soit pas detruit,
chaque caractere est represente
Par transformation de CHR$(255)
450 z=ASC(a$)
460 SYMBOL 255,a(z,1),a(z,2),a(z,3),a(z,
4),a(z,5),a(z,6),a(z,7),a(z,8)
470 PRINT CHR$(255);
480 GOTO 360
490 END
500 REM Lecture de l'écriture
informatique et modification
directe du jeu de caracteres
normal
510 RESTORE
520 FOR n=1 TO 73
530 READ m
540 FOR o=1 TO 8
550 READ p$
560 REM Conversion des chaines en
nombres binaires Puis en
nombres decimaux
570 p$="&x "+p$
580 b(o)=VAL(p$)
590 NEXT o
600 SYMBOL m,b(1),b(2),b(3),b(4),b(5),b(
6),b(7),b(8)
610 NEXT n
620 END
630 REM
640 DATA 33
650 DATA 00111000
660 DATA 00111000
670 DATA 00111000
680 DATA 00111000
690 DATA 00011000

```

```

700 DATA 00000000
710 DATA 00011000
720 DATA 00000000
730 REM "
740 DATA 34
750 DATA 01100110
760 DATA 01100110
770 DATA 01100110
780 DATA 00000000
790 DATA 00000000
800 DATA 00000000
810 DATA 00000000
820 DATA 00000000
830 REM #
840 DATA 35
850 DATA 01100110
860 DATA 11111111
870 DATA 01100110
880 DATA 01100110
890 DATA 11111111
900 DATA 01100110
910 DATA 00000000
920 DATA 00000000
930 REM '
940 DATA 39
950 DATA 00011000
960 DATA 00011000
970 DATA 00011000
980 DATA 00000000
990 DATA 00000000
1000 DATA 00000000
1010 DATA 00000000
1020 DATA 00000000
1030 REM (
1040 DATA 40
1050 DATA 00011110
1060 DATA 00011000
1070 DATA 00011000
1080 DATA 00111000
1090 DATA 00111000

```


3500 DATA 01110011
 3510 DATA 01110011
 3520 DATA 00000000
 3530 REM I
 3540 DATA 73
 3550 DATA 00001100
 3560 DATA 00001100
 3570 DATA 00001100
 3580 DATA 00001100
 3590 DATA 00111100
 3600 DATA 00111100
 3610 DATA 00111100
 3620 DATA 00000000
 3630 REM J
 3640 DATA 74
 3650 DATA 00001100
 3660 DATA 00001100
 3670 DATA 00001100
 3680 DATA 00001110
 3690 DATA 00001110
 3700 DATA 01101110
 3710 DATA 01111110
 3720 DATA 00000000
 3730 REM K
 3740 DATA 75
 3750 DATA 01100110
 3760 DATA 01100110
 3770 DATA 01101100
 3780 DATA 01111111
 3790 DATA 01100111
 3800 DATA 01100111
 3810 DATA 01100111
 3820 DATA 00000000
 3830 REM L
 3840 DATA 76
 3850 DATA 00110000
 3860 DATA 00110000
 3870 DATA 00110000
 3880 DATA 01110000
 3890 DATA 01110000

3900 DATA 01110000
 3910 DATA 01111110
 3920 DATA 00000000
 3930 REM M
 3940 DATA 77
 3950 DATA 01100111
 3960 DATA 01111111
 3970 DATA 01111111
 3980 DATA 01110111
 3990 DATA 01100111
 4000 DATA 01100111
 4010 DATA 01100111
 4020 DATA 00000000
 4030 REM N
 4040 DATA 78
 4050 DATA 01100111
 4060 DATA 01110111
 4070 DATA 01111111
 4080 DATA 01101111
 4090 DATA 01100111
 4100 DATA 01100111
 4110 DATA 01100111
 4120 DATA 00000000
 4130 REM O
 4140 DATA 79
 4150 DATA 01111111
 4160 DATA 01100011
 4170 DATA 01100011
 4180 DATA 01100111
 4190 DATA 01100111
 4200 DATA 01100111
 4210 DATA 01111111
 4220 DATA 00000000
 4230 REM P
 4240 DATA 80
 4250 DATA 01111111
 4260 DATA 01100011
 4270 DATA 01100011
 4280 DATA 01111111
 4290 DATA 01110000

5100	DATA	01100111	00011000	ATA0	0000
5110	DATA	01100111	00011000	ATA0	0000
5120	DATA	00000000	00000000	ATA0	0000
5130	REM	Y			
5140	DATA	89			
5150	DATA	01100111	00011000	ATA0	0000
5160	DATA	01100111	00011000	ATA0	0000
5170	DATA	01100111	00011000	ATA0	0000
5180	DATA	01111111	00011000	ATA0	0000
5190	DATA	00011100	00011000	ATA0	0000
5200	DATA	00011100	00011000	ATA0	0000
5210	DATA	00011100	00011000	ATA0	0000
5220	DATA	00000000	00000000	ATA0	0000
5230	REM	Z			
5240	DATA	90			
5250	DATA	01111111	00011000	ATA0	0000
5260	DATA	01100110	00011000	ATA0	0000
5270	DATA	01101100	00011000	ATA0	0000
5280	DATA	00011000	00011000	ATA0	0000
5290	DATA	00110111	00011000	ATA0	0000
5300	DATA	01100111	00011000	ATA0	0000
5310	DATA	01111111	00011000	ATA0	0000
5320	DATA	00000000	00000000	ATA0	0000
5330	REM	a			
5340	DATA	97			
5350	DATA	00000000	00000000	ATA0	0000
5360	DATA	00000000	00000000	ATA0	0000
5370	DATA	00111110	00000000	ATA0	0000
5380	DATA	00000110	00000000	ATA0	0000
5390	DATA	01111110	00000000	ATA0	0000
5400	DATA	01100110	00000000	ATA0	0000
5410	DATA	01111110	00000000	ATA0	0000
5420	DATA	00000000	00000000	ATA0	0000
5430	REM	b			
5440	DATA	98			
5450	DATA	00000000	00000000	ATA0	0000
5460	DATA	01110000	00000000	ATA0	0000
5470	DATA	01110000	00000000	ATA0	0000
5480	DATA	01111110	00000000	ATA0	0000
5490	DATA	01110110	00000000	ATA0	0000

5500	DATA	01110110	00011000	ATA0	0000
5510	DATA	01111110	00011000	ATA0	0000
5520	DATA	00000000	00000000	ATA0	0000
5530	REM	c			
5540	DATA	99			
5550	DATA	00000000	00000000	ATA0	0000
5560	DATA	00000000	00000000	ATA0	0000
5570	DATA	01111110	00000000	ATA0	0000
5580	DATA	01110000	00000000	ATA0	0000
5590	DATA	01110000	00000000	ATA0	0000
5600	DATA	01110000	00000000	ATA0	0000
5610	DATA	01111110	00000000	ATA0	0000
5620	DATA	00000000	00000000	ATA0	0000
5630	REM	d			
5640	DATA	100			
5650	DATA	00000000	00000000	ATA0	0000
5660	DATA	00000110	00000000	ATA0	0000
5670	DATA	00000110	00000000	ATA0	0000
5680	DATA	01111110	00000000	ATA0	0000
5690	DATA	01101110	00000000	ATA0	0000
5700	DATA	01101110	00000000	ATA0	0000
5710	DATA	01111110	00000000	ATA0	0000
5720	DATA	00000000	00000000	ATA0	0000
5730	REM	e			
5740	DATA	101			
5750	DATA	00000000	00000000	ATA0	0000
5760	DATA	00000000	00000000	ATA0	0000
5770	DATA	01111110	00000000	ATA0	0000
5780	DATA	01100110	00000000	ATA0	0000
5790	DATA	01111110	00000000	ATA0	0000
5800	DATA	01100000	00000000	ATA0	0000
5810	DATA	01111110	00000000	ATA0	0000
5820	DATA	00000000	00000000	ATA0	0000
5830	REM	f			
5840	DATA	102			
5850	DATA	00000000	00000000	ATA0	0000
5860	DATA	00011110	00000000	ATA0	0000
5870	DATA	00011000	00000000	ATA0	0000
5880	DATA	00111110	00000000	ATA0	0000
5890	DATA	00011000	00000000	ATA0	0000

6700 DATA 01100110
 6710 DATA 01100110
 6720 DATA 00000000
 6730 REM o
 6740 DATA 111
 6750 DATA 00000000
 6760 DATA 00000000
 6770 DATA 01111110
 6780 DATA 01101110
 6790 DATA 01101110
 6800 DATA 01101110
 6810 DATA 01111110
 6820 DATA 00000000
 6830 REM p
 6840 DATA 112
 6850 DATA 00000000
 6860 DATA 00000000
 6870 DATA 01111110
 6880 DATA 01110110
 6890 DATA 01110110
 6900 DATA 01111110
 6910 DATA 01110000
 6920 DATA 01110000
 6930 REM q
 6940 DATA 113
 6950 DATA 00000000
 6960 DATA 00000000
 6970 DATA 01111110
 6980 DATA 01101110
 6990 DATA 01101110
 7000 DATA 01111110
 7010 DATA 00001110
 7020 DATA 00001110
 7030 REM r
 7040 DATA 114
 7050 DATA 00000000
 7060 DATA 00000000
 7070 DATA 01111110
 7080 DATA 01110110
 7090 DATA 01110000

7100 DATA 01110000
 7110 DATA 01110000
 7120 DATA 00000000
 7130 REM s
 7140 DATA 115
 7150 DATA 00000000
 7160 DATA 00000000
 7170 DATA 01111110
 7180 DATA 01100000
 7190 DATA 01111110
 7200 DATA 00001110
 7210 DATA 01111110
 7220 DATA 00000000
 7230 REM t
 7240 DATA 116
 7250 DATA 00000000
 7260 DATA 00111000
 7270 DATA 01111110
 7280 DATA 00111000
 7290 DATA 00111000
 7300 DATA 00111000
 7310 DATA 00111110
 7320 DATA 00000000
 7330 REM u
 7340 DATA 117
 7350 DATA 00000000
 7360 DATA 00000000
 7370 DATA 01101110
 7380 DATA 01101110
 7390 DATA 01101110
 7400 DATA 01101110
 7410 DATA 01111110
 7420 DATA 00000000
 7430 REM v
 7440 DATA 118
 7450 DATA 00000000
 7460 DATA 00000000
 7470 DATA 01101110
 7480 DATA 01101110
 7490 DATA 01101110

```

7500 DATA 00111100
7510 DATA 00011000
7520 DATA 00000000
7530 REM w
7540 DATA 119
7550 DATA 00000000
7560 DATA 00000000
7570 DATA 01100011
7580 DATA 01101011
7590 DATA 01111111
7600 DATA 01111111
7610 DATA 00110110
7620 DATA 00000000
7630 REM x
7640 DATA 120
7650 DATA 00000000
7660 DATA 00000000
7670 DATA 01100110
7680 DATA 00111100
7690 DATA 00011000
7700 DATA 00111100
7710 DATA 01100110
7720 DATA 00000000
7730 REM y
7740 DATA 121
7750 DATA 00000000
7760 DATA 00000000
7770 DATA 01101110
7780 DATA 01101110
7790 DATA 01101110
7800 DATA 01111110
7810 DATA 00001110
7820 DATA 01111110
7830 REM z
7840 DATA 122
7850 DATA 00000000
7860 DATA 00000000
7870 DATA 01111110
7880 DATA 00001100
7890 DATA 00011000

```

```

7900 DATA 00110000
7910 DATA 01111110
7920 DATA 00000000
7930 a$=INKEY$:IF a$="" THEN GOTO 7930
7940 PRINT ASC(a$):GOTO 7930

```