

UTILITAIRE

# C O M M I

Valable pour  
 CPC 464  
 CPC 664  
 CPC 6128

Quel possesseur de CPC ne maîtrisant pour seul langage que le BASIC n'a pas désiré créer ses propres jeux d'aventures ou de société ?

L'AMSTRAD est certainement la machine qui, dans sa catégorie, dispose du BASIC le plus performant. Cependant, il ne permet ni le stockage ni l'affichage de pages-écrans et aucune commande n'autorise l'accès à la deuxième BANK du CPC 6128 ou la gestion de la mémoire écran.

Tout programme de ce type à partir du BASIC est donc impossible.

COMPACT peut faire cela et plus encore.



# PACT

1re partie

**C**OMPACT, c'est avant tout un programme de compactage qui permet de réduire vos pages-écrans et vos fenêtres graphiques à des fichiers de 1 à 9 kilos octets.

COMPACT est doté de différentes fonctions d'aide à la programmation tel le catalogue de la disquette, une calculatrice, un module de conversion décimal/hexadécimal, un module de calcul d'adresse mémoire et

de gestion des deux BANK, un module de calcul d'adresse écran ainsi que la possibilité de garder sur un fichier les caractéristiques des pages-écrans et fenêtres graphiques compactées.

Mais cela ne suffirait pas à la création de jeux si COMPACT n'était accompagné d'un jeu de commandes RSX. Ces commandes vont permettre de charger dans les deux BANK les fichiers créés par COMPACT, de les afficher et de les déplacer sur

l'écran. De par leur rapidité d'affichage, ces commandes rendent possible une certaine forme d'animation. A titre indicatif, l'affichage d'une page-écran couleur et mode écran demande une moyenne de 35 dixièmes de seconde.

L'utilisation de ces commandes ne nécessite aucune connaissance en assembleur, tout juste faut-il posséder quelques notions dans la structure de la mémoire, notions qui seront large-





ment supplées par la présence d'aides à la programmation contenues dans COMPACT.

En conclusion, COMPACT est le complément indispensable à votre éditeur ou à votre numériseur graphique.

## COMMENT UTILISER COMPACT ?

• COMPACT travaille sur des pages-écrans sauvegardées par la commande SAVE «nompag», B.&C000.&4000 qui donne sur disquette un fichier binaire de 17 Ko. Les pages-écrans issues d'un éditeur graphique devront être sauvegardées sous cette forme.

Pour des besoins d'ergonomie le choix de la présentation s'est porté sur un menu simple, car en effet peu de possesseurs de CPC disposent de souris, les menus déroulants ont donc été proscrits. COMPACT se présente par une succession de menus où la sélection est faite par l'appui de la touche correspondant au numéro se trouvant avant l'option choisie. Il suffit de sélectionner une option et de répondre aux questions posées par le programme.

## MODE D'EMPLOI

L'appui sur la barre ESPACE permet le changement des couleurs du papier et du stylo suivant dix combinaisons.

## COMPACTAGE BINAIRE

Point fort de ce logiciel, cette commande réduit un morceau ou la totalité de la page-écran à des fichiers réutilisables par les commandes RSX.

### Compactage de page-écran

- Entrez le mode écran
  - Entrez le nom du fichier page 17 Ko
  - Entrez le nom du fichier nouvellement créé
  - Entrez deux couleurs pour chaque stylo
- (Ceci afin de permettre le clignotement des stylos).
- Si vous avez des stylos de deux couleurs différentes, répondez oui à la question concernant la vitesse de clignotement des couleurs.

- Si oui, réglez la vitesse (mode d'emploi à l'écran).
- Appuyez sur une touche pour afficher la page-écran à compacter.
- Appuyez sur une touche afin de débiter le compactage.

Ceci afin de permettre de sauvegarder la page-écran compactée sur une autre disquette, le compactage étant automatiquement suivi de la sauvegarde.

Après le compactage, le programme affiche à l'écran les caractéristiques suivantes : mode écran, type de fichier (page-écran ou fenêtre graphique), nom du fichier, bénéfice par rapport à un fichier non compacté, adresse de début de la mémoire écran, adresse de fin de la mémoire écran, taille réelle du fichier en mémoire, taille du fichier sur la disquette. La possibilité vous est offerte de sauvegarder les caractéristiques sur un fichier.

- Désirez-vous sauvegarder les caractéristiques sur le fichier «BIBLIOTEC» ?
- Si oui, entrez un commentaire ou une remarque de votre choix.

Pour plus de détail sur le fichier «BIBLIOTEC», voir le paragraphe ARCHIVES.

### Compactage de fenêtre graphique

Deux possibilités se présentent à vous : vous connaissez les coordonnées graphiques de votre fenêtre ou vous les ignorez. Si vous êtes dans le second cas, vous pouvez les déterminer en répondant oui à question :

- Appel d'une page-écran pour déterminer les coordonnées ?
- Entrez le nom de la page-écran source.
- Est-ce une page-écran compactée ou une page 17 Ko ?

Si la page-écran que vous désirez appeler n'est pas compactée, l'ordinateur vous posera la question suivante.

- Entrez le mode écran

Si vous connaissez les coordonnées de votre fenêtre, un tableau s'affichera sur votre écran. Il vous faudra entrer

les coordonnées graphiques. Sinon lisez le mode d'emploi affiché à l'écran et appuyez sur une touche. A noter que le cadre permettant de déterminer les coordonnées ne se déplace pas de pixel en pixel mais d'octet en octet.

- Entrez le nom du fichier nouvellement créé

COMPACT permet de changer l'attribution des stylos. Ceci est très utile lorsque l'on veut accorder sur le même écran différentes fenêtres issues de pages-écrans différentes.

- Désirez-vous changer les stylos ?
- Appuyez sur une touche pour débiter le compactage.

Les caractéristiques du fichier nouvellement créé s'affichent à l'écran, la possibilité vous est offerte de les sauvegarder.

- Désirez-vous sauvegarder les caractéristiques sur le fichier «BIBLIOTEC» ?
- Si oui, entrez un commentaire ou une remarque de votre choix.

## DECOMPACTAGE BINAIRE

Cette option a pour but de vérifier si le compactage s'est bien déroulé, cela sans sortir de COMPACT. L'affichage des fenêtres graphiques se fera dans les couleurs utilisées lors du dernier compactage de page-écran ou par défaut dans les couleurs d'origine du programme.

### Décompactage de page-écran

- Entrez le nom du fichier.

Pour revenir au menu principal appuyez sur une touche.

### Décompactage de fenêtre graphique

- Entrez le nom du fichier.
- Entrez le mode écran.

Pour revenir au menu principal, appuyez sur une touche.

## CATALOGUE

Cette commande affiche le catalogue de la disquette ; pour revenir au menu principal, appuyez sur une touche.



## ARCHIVES

COMPACT vous offre la possibilité de sauvegarder les caractéristiques de vos pages-écrans et fenêtres graphiques sur un fichier appelé «BIBLIOTEC.BIN». Ce fichier est automatiquement créé lors de la première sauvegarde. Cette option est très utile car elle permet de se constituer une bibliothèque de fichiers, avec un fichier «BIBLIOTEC» sur chaque face de vos disquettes. La capacité de «BIBLIOTEC» est de 60 fichiers ce qui correspond à la capacité du catalogue d'une face de disquette 3 1/2.

### Calculatrice

Comme son nom l'indique, c'est une calculatrice qui regroupe les quatre opérations. Le résultat vous est donné en décimal et hexadécimal.

- Entrez deux nombres et appuyez sur la touche correspondant à l'opération désirée.

### Conversion décimal/hexadécimal

Comme son nom l'indique, cette option permet de convertir un nombre décimal en hexadécimal et vice versa.

### Calcul d'adresse écran

Cette option permet la conversion des coordonnées physiques d'un point en position à l'écran, exemple : X=0, Y=400 ⇒ (CONVERSION) ⇒ &C000. Ceci est très utile pour calculer la position d'une fenêtre à l'écran.

Comme dans le compactage d'une fenêtre, deux solutions se présentent à vous, vous connaissez les coordonnées ou vous ne les connaissez pas, en ce cas, la possibilité vous est offerte de les déterminer ; répondez oui à la question suivante.

- Appel d'une page-écran pour déterminer les coordonnées à l'écran ?

Si vous avez répondu oui, le programme vous posera les questions suivantes :

- Nom de la page-écran.  
- Page-écran compactée oui ou non ?

Si vous avez répondu oui, le programme vous demandera :

- Entrez le mode écran.

Si vous avez opté pour l'appel d'une page-écran, suivez le mode d'emploi affiché à l'écran sinon :

- Entrez les coordonnées X et Y de votre point.

### Calcul d'adresse mémoire

Cette option vous évitera de fastidieux calculs, car en effet elle permet de calculer les adresses de relogement pour vos fichiers, ceci dans les deux BANK. Pour cela il suffit de fixer une portion de mémoire située en BANK 0 dans laquelle vous désirez loger vos fichiers. Puis entrez la taille de vos fichiers, taille qui peut être entrée en décimal ou en hexadécimal, le programme calculera automatiquement les adresses mémoire. En cas de débordement en BANK 0, le programme relogera les fichiers restants en BANK 1. Si vous désirez loger vos fichiers uniquement en BANK 1, il vous faudra fixer une tranche mémoire égale à zéro, c'est-à-dire donner les deux mêmes nombres pour les adresses supérieures et inférieures.

L'accès à la BANK 1 étant soumis à certaines restrictions et les commandes RSX étant situées à une certaine adresse en mémoire, il vous est fortement recommandé de vous reporter au RAPPEL SUR L'ORGANISATION DE LA RAM DES CPC.

- Entrez l'adresse inférieure.  
- Entrez l'adresse supérieure.  
- Entrez la taille de vos fichiers (maximum 60 fichiers).

Afin d'obtenir une efficacité maximale dans le calcul des adresses mémoire, il est préférable de rentrer les adresses dans un ordre décroissant.

- Entrez un 0 pour sortir.

Les adresses défilent à l'écran treize par treize, notez-les et appuyez sur une touche pour obtenir les autres. Sur la droite, une représentation des deux «BANK» vous montre la place occupée par vos fichiers. Pour finir, un tableau apparaît vous indiquant la place encore disponible en mémoire. A noter que le nombre donné pour la BANK 0 correspond uniquement à la place disponible dans la tranche que vous avez déterminée précédemment.

## NETTOYAGE BASIC

Cette option est un plus à COMPACT ; en effet, ce programme permet d'effacer les blancs qui subsistent en bout de lignes de vos programmes BASIC ainsi que l'instruction REM, que ce soit sous sa forme entière ou abrégée (\*). Ceci dans le but évident de grappiller quelques octets. Seule restriction à son efficacité : les REM en bout de lignes ne seront pas effacés.

ATTENTION, il est nécessaire que la disquette contenant le programme BASIC à «nettoyer» dispose d'une place supérieure à la taille du programme.

- Introduisez la disquette contenant COMPACT et appuyez sur une touche.  
- Introduisez la disquette contenant le programme BASIC à nettoyer et appuyez sur une touche.  
- Entrez le nom de votre programme.  
- Appuyez sur F2.

- Sélectionner une option.  
- Entrez un nom pour votre programme.  
- Désirez-vous effacer l'ancien programme ?  
- Si oui, rappelez son nom.  
- Appuyez sur F2.

Votre programme est sauvegardé. Si vous voulez retourner à COMPACT, provoquez un RESET et rechargez le programme.

Eric CINGET

à suivre

### GAGNER AU LOTO

un rêve qui peut devenir réalité avec

### LOTO - MATIC

le programme qui vous révèle tout ce que vous devez savoir pour :

• trouver facilement les numéros qui ont le plus de chance de sortir

• établir scientifiquement les grilles les plus performantes grâce aux tests du Lotoscope

• contrôler sans peine les résultats de vos jeux

Editions écran et imprimante

Documentation détaillée

+ CADEAU contre 4 timbres

**INFORMATIC Applications**

B.P. 78 - 67800 BISCHHEIM

Tél. 88.33.58.85



# COMPACT

```

10 REM =====>AB
20 REM ==>YC
30 REM == * COMPACT * ==>QW
40 REM ==>YE
50 REM == compacteur d'images ==>QH
60 REM ==>YG
70 REM == par Eric CINGET ==>WP
80 REM ==>YJ
90 REM == 1988 ==>FK
100 REM ==>DF
110 REM =====>BC
120 REM>TK
130 MEMORY &74FF:LOAD"compa.bin":CALL &A0B0:POKE &A0D6,>KQ
1:CALL &A094,&7500,&25A0:in0(nn)=1:in1(nn)=26:DIM pe(16
),pe1(16),st(16),flc$(60):mo=0:ben=0:KEY DEF 11,1:KEY D
EF 14,1:KEY DEF 4,1:KEY DEF 20,1:RESTORE 1830:FOR i=0 T
O 9:READ a,b:in0(i)=a:in1(i)=b:NEXT
140 x1=&C000:a=0:x2=&FFFF:x3=80:x3$="":x1$="":x2$="":no>VK
m$="":nom1$="":a$="":aa$="":bb$="":sa=0:ec=0:a=0
150 st1=0:adr=&7500:RESTORE 1820:FOR i=0 TO 15:READ a:p>XB
e(i)=a:pe1(i)=a:NEXT:b4$="Mode ecran /0/1/2/ => ":b5$="
Nom de la page ecran a compacter => ":b7$="Nom de la no
uvel page ecran => ":b2$="Page ecran compactee /0/N/":
b18$="Entrez le premier nombre =>"
160 era=0:b3$="Entrez le deuxieme nombre => ":b8$="Appe>BT
l d'une page ecran afin de determinez les coordonnees /
0/N/":b9$="Nom de la page ecran source => ":b11$="Coule
ur 1 du stylo N ":b16$="Couleur 2 du stylo N ":ON ERROR
GOTO 1690
170 b19$="Desirez vous changer les stylo /0/N/":b13$="N>TF
om du nouveau fichier => ":b14$="Nom de la page ecran =
"> ":b15$="Couleur numerot 1 du bord de l'ecran => ":
b17$="Couleur numerot 2 du bord de l'ecran => ":b6$="
Coordonnees horizontales (X) => "
180 b12$="Coordonnees verticales (Y) => ":b10$="Adresse>LG
inferieur => ":b1$="Adresse superieur => ":b20$="Entre
z la taille du fichier N":bo=0:bol=0:vlt=1:v1=10:v2=10:
cc=0:sup$="":inf$=""
190 PEN 0:PRINT FRE(""):MODE 2:INK 0,in0(nn):INK 1,in1(>LY
nn):BORDER in0(nn):ORIGIN 0,0,6,632,390,374:GRAPHICS PE
N 0:CLG 1:TAG:MOVE 128,388,0,0:PRINT " * * * * *
C O M P A C T * * * * *":TAGOFF:ORIGIN 0,0,0,6
40,0,400:IF ben<0 THEN ben=0
200 WINDOW 8,73,8,22:PAPER 0:PEN 1:CLS:WINDOW#1,8,73,5,>MZ
5:MOVE 0,396,1:DRAW 638,396:DRAW 638,368:DRAW 0,368:DRA
W 0,396:MOVE 50,44:DRAW 590,44:DRAW 590,294:DRAW 50,294
:DRAW 50,44:MOVE 50,310:DRAW 590,310:DRAW 590,344:DRAW
50,344:DRAW 50,310:MOVE 54,42
210 DRAW 591,42:DRAW 591,290:MOVE 592,42:DRAW 592,290:M>TG
OVE 593,42:DRAW 593,290:MOVE 54,308:DRAW 591,308:DRAW 5
91,340:MOVE 592,308:DRAW 592,340:MOVE 593,308:DRAW 593,
340
220 CALL &A094,&7500,&25A0:GRAPHICS PAPER 0:IF sa=1 THE>GH
N sa=0:GOTO 970 ELSE PAPER#1,0:PEN#1,1:CLS#1:IF cc=1 TH
EN 280 ELSE IF ccc=1 THEN RETURN

```

```

230 TAG:MOVE 257,334,1:PRINT" MENU PRINCIPAL *":TAGOFF:>HD
LOCATE 6,3:PRINT"- 1 - COMPACTAGE BINAIRE":LOCATE 6,5:
PRINT"- 2 - DECOMPACTAGE BINAIRE":LOCATE 6,7:PRINT"- 3
- CATALOGUE DE LA DISQUETTE":LOCATE 6,9:PRINT"- 4 -
ARCHIVES"
240 LOCATE 6,11:PRINT"- 5 - OPERATIONS":LOCATE 6,13:PR>PM
INT"- 6 - NETTOYAGE BASIC"
250 a$="":CLEAR INPUT:WHILE NOT(a$="1" OR a$="2" OR a$=>AB
"3" OR a$="4" OR a$="5" OR a$="6" OR a$=" "):a$=INKEY$:
WEND
260 IF a$=" " THEN nn=nn+1:WHILE nn=10:nn=0:WEND:INK 0,>PU
in0(nn):INK 1,in1(nn):BORDER in0(nn):GOTO 250
270 CLS:CLS#1:IF a$="2" THEN 910 ELSE IF a$="3" THEN TA>TQ
G:MOVE 283,334:PRINT"CATALOGUE":TAGOFF:CAT:CLEAR INPUT
:CALL &BB06:GOTO 190 ELSE IF a$="4" THEN 1130 ELSE IF a
$="5" THEN 1250 ELSE IF a$="6" THEN 1840
280 MOVE 140,2:DRAW 498,2:DRAW 498,26:DRAW 140,26:DRAW>PF
140,2:MOVE 142,0:DRAW 499,0:DRAW 499,22:MOVE 500,0:DRAW
500,22:MOVE 501,0:DRAW 501,22:MOVE 266,2:DRAW 266,26:M
OVE 270,2:DRAW 270,26:MOVE 306,2:DRAW 306,26:MOVE 376,2
:DRAW 376,26:MOVE 380,2
290 DRAW 380,26:MOVE 448,2:DRAW 448,26:MOVE 190,2:DRAW>EK
190,26:GRAPHICS PAPER 0:TAG:MOVE 146,20:PRINT"DEBUT":M
OVE 276,20:PRINT"FIN":MOVE 386,20:PRINT"LARGEUR":TAGO
FF
300 IF cc=1 THEN RETURN>RH
310 TAG:MOVE 250,334:PRINT"COMPACTAGE BINAIRE":TAGOFF:>QV
LOCATE 4,3:PRINT"- 1 - COMPACTAGE DE PAGE ECRAN":LOCAT
E 4,6:PRINT"- 2 - COMPACTAGE DE FENETRE GRAPHIQUE":LOC
ATE 4,9:PRINT"- 3 - RETOUR MENU PRINCIPAL":a$=""
320 CLEAR INPUT:WHILE NOT(a$="1" OR a$="2" OR a$="3"):a>GY
$=INKEY$:WEND:CLS:CLS#1:IF a$="2" THEN 470 ELSE IF a$="
3" THEN 190 ELSE TAG:MOVE 232,334:PRINT"COMPACTAGE DE P
AGE ECRAN":TAGOFF
330 x1=&C000:x1$="c000":x2=&FFFF:x2$="ffff":TAG:MOVE 20>QD
6,20:PRINT"&C000":MOVE 320,20:PRINT"&FFFF":MOVE 462,2
0:PRINT"&50":TAGOFF:LOCATE 4,2:PRINT b4$:LOCATE 4,4:PR
INT b5$:LOCATE 4,6:PRINT b7$
340 LOCATE 4,2:PRINT " ";b4$::INPUT "",mo:IF mo<0 OR mo>>HR
2 THEN PRINT " ":GOTO 340 ELSE LOCATE 4,2:PRINT " ";b4$:
LOCATE 4,4:PRINT " ";b5$::INPUT "",nom$:LOCATE 4,4:PRINT
" ";b5$:LOCATE 4,6:PRINT " ";b7$::INPUT "",nom1$:LOCATE
4,6:PRINT " ";b7$
350 CLS:IF mo=0 THEN me=15 ELSE IF mo=1 THEN me=3 ELSE>BL
me=1
360 y=-1:FOR i=0 TO me:y=y+1:LOCATE 4,1+(y*2):PRINT " ";>QU
b11$;i::INPUT " => ",pe(i):LOCATE 4,1+(y*2):PRINT " ";b
11$;i:y=y+1:LOCATE 4,1+(y*2):PRINT " ";b16$;i::INPUT " =
"> ",pe1(i):LOCATE 4,1+(y*2):PRINT " ";b16$;i:IF y=7 THE
N CLS:y=-1
370 IF pe(i)>26 OR pe(i)<0 OR pe1(i)>26 OR pe1(i)<0 THE>HM
N ERROR 100 ELSE IF pe(i)<>pe1(i) THEN pee=pe(i):peel=p
e1(i)
380 NEXT i:CLS:LOCATE 4,2:PRINT " ";b15$::INPUT "",bo:LO>GC
CATE 4,2:PRINT " ";b15$:LOCATE 4,4:PRINT " ";b17$::INPUT
"",bol:IF bo<>bol THEN pee=bo:peel=bol
390 LOCATE 4,4:PRINT " ";b17$:LOCATE 4,6:PRINT" Desirez>XF
vous determiner la vitesse de clignotement des ":LOCATE
4,8:PRINT"couleurs /0/N/":a$="":WHILE NOT(a$="0" OR a$

```



```
=M"):a#=UPPER$(INKEY$):VEND:PRINT " ":CLS:IF a#="N" THE
N ec=1:GOTO 460
400 LOCATE 2,1:PRINT"- POUR AUGMENTER LA VITESSE DE CLI >UR
GNOTEMENT DE LA COULEUR N1 APPUYEZ SUR <1> POUR
DIMINUER APPUYEZ SUR <2>"
410 LOCATE 2,4:PRINT"- POUR AUGMENTER LA VITESSE DE CLI >UD
GNOTEMENT DE LA COULEUR N2 APPUYEZ SUR <9> POUR
DIMINUER APPUYEZ SUR <0>"
420 LOCATE 2,8:PRINT"- POUR SORTIR APPUYEZ SUR <ESPACE> >WA
":LOCATE 2,11:PRINT"Vitesse de la couleur n1 en 50/ieme
de secondes => 10":LOCATE 2,13:PRINT"Vitesse de la co
uleur n2 en 50/ieme de secondes => 10":BORDER pee,pee1
:vi=10:v2=10
-430 SPEED INK vi,v2:a#=INKEY$:IF a#="1" THEN vi=vi+1 EL >RD
SE IF a#="2" THEN vi=vi-1 ELSE IF a#="9" THEN v2=v2+1 E
LSE IF a#="0" THEN v2=v2-1 ELSE IF a#=" " THEN BORDER i
n0(nn):ec=1:GOTO 460
440 IF vi=0 THEN vi=1 ELSE IF vi=256 THEN vi=255 ELSE I >ZK
F v2=256 THEN v2=255 ELSE IF v2=0 THEN v2=1
450 LOCATE 52,11:PRINT v1:LOCATE 52,13:PRINT v2:GOTO 43 >BP
0
460 GOTO 800 >ZE
470 TAG:MOVE 202,334:PRINT"COMPACTAGE DE FENETRE GRAPHI >KZ
QUE";TAGOFF:LOCATE 2,6:PRINT " ";b8$:a#="":WHILE NOT(a#
="0" OR a#="N"):a#=UPPER$(INKEY$):VEND:PRINT " ":CLS:IF
a#="N" THEN cc=0 ELSE cc=1
480 LOCATE 4,2:PRINT " ";b9$;:INPUT "",nom$:LOCATE 4,2:PR >XU
INT " ";b9$:LOCATE 4,4:PRINT " ";b2$:aa#="":WHILE NOT(aa
#="N" OR aa#="0"):aa#=UPPER$(INKEY$):VEND:LOCATE 4,4:PR
INT " ";b2$:IF aa#="0" AND cc=0 THEN 680 ELSE IF aa#="0"
THEN 500
490 LOCATE 4,6:PRINT " ";b4$;" ";:INPUT "",mo:IF mo<0 OR >AM
mo>2 THEN 490 ELSE LOCATE 4,6:PRINT b4$:IF cc=0 THEN 6
80
500 CLS:LOCATE 25,2:PRINT"MODE D'EMPLOI":LOCATE 3,5:PRI >NG
NT"- DEPLACEMENT DU CADRE : Touches de deplacement du cu
rseur"
510 LOCATE 3,7:PRINT"- AGRANDISSEMENT DU CADRE : horizo >BQ
ntal => F4 ; vertical => F2 ":LOCATE 3,9:PRINT"- RETREC
ISSEMENT DU CADRE : horizontal => F6 ; vertical => F8 "
:LOCATE 3,11:PRINT"- VALIDATION DE LA POSITION ET RETOU
R : < espace >"
520 LOCATE 9,14:PRINT"< INTRODUISEZ UNE DISQUETTE ET PRE >XG
SSEZ UNE TOUCHE >":CALL &BB06:PEN 0:IF aa#="0" THEN LOA
D nom$, &7500:PAGE, &7500,0:CALL &9B56:mo=PEEK(&9B56):PO
KE &9B56,0
530 IF mo=0 THEN me=15 ELSE IF mo=1 THEN me=3 ELSE me=1 >WP
540 IF aa#="0" THEN 550 ELSE MODE mo:FOR j=0 TO me:INK >VG
j,pe(j),pe1(j):NEXT j:BORDER bo,bo1:LOAD nom$, &C000
550 PEN 1:CLEAR INPUT:SPEED KEY 1,1:dx=0:dy=0:xx=223:xx >EE
1=329:yy=150:yy1=250
560 MOVE xx+dx,yy+dy,1,1:DRAW xx1+dx,yy+dy:DRAW xx1+dx, >UV
yy1+dy:DRAW xx+dx,yy1+dy:DRAW xx+dx,yy+dy:a#="":CALL &B
B06
570 MOVE xx+dx,yy+dy,1,1:DRAW xx1+dx,yy+dy:DRAW xx1+dx, >QV
yy1+dy:DRAW xx+dx,yy1+dy:DRAW xx+dx,yy+dy
580 IF INKEY(1)<-1 THEN dx=dx+4 >ZE
```

```
590 IF INKEY(8)<-1 THEN dx=dx-4 >ZQ
600 IF INKEY(2)<-1 THEN dy=dy-2 >ZA
610 IF INKEY(0)<-1 THEN dy=dy+2 >ZJ
620 IF INKEY(4)<-1 THEN xx=xx+4 >AM
630 IF INKEY(20)<-1 THEN xx=xx-4 >AZ
640 IF INKEY(11)<-1 THEN yy=yy+2 >AA
650 IF INKEY(14)<-1 THEN yy=yy-2 >AG
660 IF INKEY(47)<-1 THEN MOVE 0,0,1,0:MODE 2:dc=(xx1-4 >ZV
)+dx:gc=(xx+4)+dx:hc=(yy1-1)+dy:bc=(yy+3)+dy:SPEED KEY
30,2:GOSUB 190:GOTO 740
670 GOTO 560 >AA
680 LOCATE 7,9:PRINT"COORDONNEES GRAPHIQUES DROITE GA >UK
UCHE HAUT BAS":MOVE 96,124:DRAW 544,124:DRAW 544,
170:DRAW 96,170:DRAW 96,124:MOVE 544,170:DRAW 544,80:DR
AW 288,80:DRAW 288,170:MOVE 352,80:DRAW 352,170:MOVE 41
6,80:DRAW 416,170:MOVE 480,80
690 DRAW 480,170:MOVE 100,123:DRAW 288,123:MOVE 292,79: >LJ
DRAW 545,79:DRAW 545,168:DRAW 546,79:DRAW 546,168:DRAW
547,79:DRAW 547,168
700 LOCATE 31,9:PRINT" DROITE":LOCATE 33,12:INPUT "",dc >CG
:LOCATE 31,9:PRINT" DROITE":LOCATE 39,9:PRINT" GAUCHE":
LOCATE 41,12:INPUT "",gc:LOCATE 39,9:PRINT" GAUCHE":LOC
ATE 48,9:PRINT" HAUT":LOCATE 49,12:INPUT "",hc:LOCATE 4
8,9
710 PRINT" HAUT":LOCATE 57,9:PRINT" BAS":LOCATE 57,12:I >RX
NPUT "",bc:LOCATE 57,9:PRINT" BAS":dc=dc-1:gc=gc+1:GOTO
740
720 IF f1<=0.0625 THEN f3=0 ELSE IF f1<=0.1875 THEN f3= >WP
1 ELSE IF f1<=0.3125 THEN f3=2 ELSE IF f1<=0.4375 THEN
f3=3 ELSE IF f1<=0.5625 THEN f3=4 ELSE IF f1<=0.6875 TH
EN f3=5 ELSE IF f1<=0.8125 THEN f3=6 ELSE f3=7
730 RETURN >ZF
740 f1=((400-hc)/16):f2=FIX(f1):f1=f1-f2:GOSUB 720:x1$= >EU
HEX$(49152+(f2*80)+(f3*2048)+FIX(gc/8)):f1=((400-hc)/16
):f2=FIX(f1):f1=f1-f2:GOSUB 720:x2$=HEX$(49152+(f2*80)+
(f3*2048)+FIX(dc/8)):x3$=HEX$(FIX((VAL("&"+x2$))-VAL("
&"+x1$))+1)
750 f1=((400-bc)/16):f2=FIX(f1):f1=f1-f2:GOSUB 720:x2$= >HV
HEX$(49152+(f2*80)+(f3*2048)+FIX(dc/8)):CLS:x1=VAL("&"+
x1$):x2=VAL("&"+x2$):x3=(VAL("&"+x3$))
760 TAG:MOVE 206,20:PRINT "&";x1$;:MOVE 320,20:PRINT "& >WA
";x2$;:MOVE 462,20:PRINT "&";x3$;" ";:MOVE 202,334:PRIN
T "COMPACTAGE DE FENETRE GRAPHIQUE";TAGOFF
770 LOCATE 4,4:PRINT b19$:LOCATE 4,2:PRINT " ";b13$;:IMP >ZT
UT "",nom1$:LOCATE 4,2:PRINT " ";b13$:LOCATE 4,4:PRINT "
";b19$;" ":a#="":WHILE NOT(a#="0" OR a#="N"):a#=UPPER$(
INKEY$):VEND:IF a#="N" THEN 800 ELSE IF cc=0 THEN me=15
780 CLS:y=0:FOR i=0 TO me:y=y+2:LOCATE 4,y:PRINT " ";Ec >TH
hange stylo N";i;"avec stylo N => ";:INPUT "",st(i):LOC
ATE 4,y:PRINT " ";Echange stylo N";i;"avec stylo N => "
:IF y=14 THEN y=0:CLS
790 IF st(i)>15 OR st(i)<0 THEN ERROR 100 ELSE NEXT:CLS >FU
:sti=1
800 CLS:LOCATE 8,3:PRINT"APRES AFFICHAGE DE VOTRE PAGE >EA
ECRAN APPUYEZ SUR UNE":LOCATE 15,5:PRINT"TOUCHE POUR CO
MMENCER LE COMPACTAGE":LOCATE 21,7:PRINT"<APPUYEZ SUR U
NE TOUCHE>":CLEAR INPUT:CALL &BB06:IF ec=1 THEN aa#="N"
```



```

810 vlt=1:cc=0:IF aa$="N" THEN aa$="":MODE mo:PEN 0:LOA >HT
D nom$, &C000:PEN 1:FOR i=0 TO me:INK i, pe(i), pe(1):NEX
T i:BORDER bo, bo1
820 IF aa$="0" THEN PEN 0:LOAD nom$, &7500:PAGE, &7500, 0 >EQ
:PEN 1
830 aa$="":IF st1=1 THEN st1=0:CALL &9B7F, (bc-10), (gc-1 >WQ
0), (hc+10), (dc+10), st(0), st(1), st(2), st(3), st(4), st(5),
st(6), st(7), st(8), st(9), st(10), st(11), st(12), st(13), st(
14), st(15)
840 POKE &9B38, 0:CALL &9B38, vlt:a=PEEK(&9B38):IF a=vlt >FA
THEN vlt=vlt+1:GOTO 840
850 CALL &A094, &7500, &25A0:CLEAR INPUT:IF ec=1 THEN ec= >KB
0:GOTO 880
860 CALL &BB06:adr=&7500:POKE adr, 103:POKE adr+1, 49:POK >TQ
E adr+2, 201:POKE adr+3, 217:POKE adr+4, vlt:POKE adr+5, x3
:ad$=MID$(x1$, 3, 2):POKE adr+8, VAL("&"+ad$):ad$=MID$(x1$
, 1, 2):POKE adr+9, VAL("&"+ad$)
-870 CALL &9F1E, x3, x1, x2, (adr+6), (adr+9), vlt:ad2$=(HEX$( >VU
PEEK(adr+7), 2))+ (HEX$(PEEK(adr+6), 2)):ad2$=HEX$(VAL("&
"+ad2$))+14, 4):SAVE nom1$, b, adr, VAL("&"+ad2$):sa=1:ty=0
:adr1=adr:ben=((hc-bc)/2)*x3-(VAL("&"+ad2$)):GOTO 190

880 CALL &BB06:adr=&7500:adr1=adr:POKE adr, 103:adr=adr+ >LT
1:POKE adr, 49:adr=adr+1:POKE adr, 201:adr=adr+1:POKE adr
, 126:adr=adr+1:POKE adr, vlt:adr=adr+1:adr2=adr:adr=adr+
2:POKE adr, mo
890 FOR i=0 TO me:ii=me-i:adr=adr+1:POKE adr, pe(ii):adr >JF
=adr+1:POKE adr, pe(11):NEXT:adr=adr+1:POKE adr, bo:adr=
adr+1:POKE adr, bo1:adr=adr+1:POKE adr, v1:adr=adr+1:POKE
adr, v2:CALL &9DE8, adr2, adr, vlt
900 ad$=HEX$(PEEK(adr2), 2):ad1$=HEX$(PEEK(adr2+1), 2):ad >DM
2$=ad1$+ad$:a=(VAL("&"+ad2$))+((me+1)*2)+12):SAVE nom1$,
b, adr1, a:sa=1:ty=1:ad2$=HEX$(a, 4):ben=16384-(VAL("&"+ad
2$)):GOTO 190
910 TAG:MOVE 242, 334:PRINT"DECOMPACTAGE BINAIRE";:TAGOF >JG
F:LOCATE 4, 3:PRINT"- 1 - DECOMPACTAGE DE PAGE ECRAN":L
OCATE 4, 6:PRINT"- 2 - DECOMPACTAGE DE FENETRE GRAPHIQU
E":LOCATE 4, 9:PRINT"- 3 - RETOUR MENU PRINCIPAL":a$=""

920 WHILE NOT(a$="1" OR a$="2" OR a$="3"):a$=INKEY$:WEN >HT
D:CLS:CLS#1:IF a$="2" THEN 940 ELSE IF a$="3" THEN 190
ELSE TAG:MOVE 218, 334:PRINT"DECOMPACTAGE DE PAGE ECRAN"
;:TAGOFF
930 LOCATE 4, 2:PRINT" ";b9$;:INPUT "", nom$:PRINT" ":PEN >YF
0:LOAD nom$, &7500:MODE 2:LOCATE 19, 11:PEN 1:PAGE, &750
0, 0:CLEAR INPUT:CALL &BB06:GOTO 190
940 TAG:MOVE 232, 334:PRINT"DECOMPACTAGE DE FENETRE";:TA >HX
GOFF:LOCATE 4, 2:PRINT"Nom de la fenetre => ":LOCATE 4, 4
:PRINT b4$:LOCATE 4, 2:PRINT" Nom de la fenetre => ";:IN
PUT "", nom$:LOCATE 4, 2:PRINT" Nom de la fenetre => "
950 LOCATE 4, 4:PRINT" ";b4$;:INPUT "", mo:IF mo<0 OR mo >RU
2 THEN PRINT" ":GOTO 950 ELSE LOCATE 4, 4:PRINT" ";b4$:I
F mo=0 THEN me=15 ELSE IF mo=1 THEN me=3 ELSE me=1
960 PEN 0:LOAD nom$, &7500:PEN 1:FOR i=0 TO me:INK i, pe( >UZ
i), pe(1):NEXT:BORDER bo, bo1:MODE mo:LOCATE 6, 11:IFENET
RE, 0, &7500, 0:CLEAR INPUT:CALL &BB06:GOTO 190
970 TAG:MOVE 285, 334:PRINT"ARCHIVES";:TAGOFF:LOCATE 2, 1 >ZU
:PRINT"MOH ":LOCATE 2, 3:PRINT"TYPE ":LOCATE 2, 5:PRINT
"COMMENTAIRES ":LOCATE 40, 1:PRINT"MODE ":LOCATE 40, 3:

```

```

PRINT"BENEFICE : "
980 MOVE 150, 196:DRAW 490, 196:DRAW 490, 84:DRAW 150, 84:D >WW
RAW 150, 196:MOVE 154, 83:DRAW 491, 83:DRAW 491, 193:MOVE 4
92, 83:DRAW 492, 193:MOVE 493, 83:DRAW 493, 193:MOVE 150, 11
3:DRAW 490, 113:MOVE 150, 140:DRAW 490, 140:MOVE 150, 167:D
RAW 490, 167:MOVE 150, 170
990 DRAW 490, 170:MOVE 232, 196:DRAW 232, 84:MOVE 235, 196: >QD
DRAW 235, 84:MOVE 320, 196:DRAW 320, 84:MOVE 405, 196:DRAW
405, 84:MOVE 490, 196:DRAW 490, 84:MOVE 150, 196:DRAW 235, 1
70
-1000 TAG:GRAPHICS PAPER 0:MOVE 168, 158, 1:PRINT"DEBUT";: >ZW
MOVE 168, 130:PRINT"FIN";:MOVE 168, 104:PRINT"TAILLE";:MO
VE 260, 190:PRINT"HEXA. ";:MOVE 336, 190:PRINT"DECIMAL";:M
OVE 434, 190:PRINT"K.O";:TAGOFF:IF bi=1 THEN 1180
1010 LOCATE 8, 1:PRINT UPPER$(nom1$):LOCATE 46, 1:PRINT " >ZJ
o:LOCATE 50, 3:PRINT CINT(ben);"OCTETS":LOCATE 9, 3:IF ty
=0 THEN PRINT"FENETRE GRAPHIQUE" ELSE PRINT"PAGE ECRAN"
1020 TAG:MOVE 256, 158:PRINT "&";HEX$(x1);:MOVE 336, 158: >MV
PRINT x1;:MOVE 256, 132:PRINT "&";HEX$(VAL("&"+x2$));:MO
VE 336, 132:PRINT (VAL("&"+x2$));:MOVE 256, 104:PRINT "&
";HEX$(VAL("&"+ad2$), 4);:MOVE 336, 104:PRINT VAL("&"+ad2$
);:a$=STR$(VAL("&"+ad2$))
1030 MOVE 420, 158:PRINT "XXXXXX";:MOVE 420, 132:PRINT " >VM
XXXXXX";:IF LEN(a$)>4 THEN b$=LEFT$(a$, 2):a=VAL(b$)+1
ELSE a=1
1040 MOVE 434, 104:PRINT a;:TAGOFF:LOCATE 2, 15:PRINT"Sau >BZ
vegarde sur le fichier BIBIOTHEQUE 70/N/":a$="":WHILE N
OT(a$="0" OR a$="N"):a$=UPPER$(INKEY$):WEND:IF a$="N" T
HEN 190 ELSE LOCATE 2, 5:PRINT" COMMENTAIRES : ";
1050 PRINT STRING$(48, 32);" ":LOCATE 16, 5:INPUT "", b$:W >VH
INDOW 79, 80, 24, 24:CLS:PEN 0:CALL &A094, &7500, &25A0
1060 PEN 0:LOAD "bibiotec.bin", &8000:PEN 1 >NF
1070 FOR i=0 TO 3840 STEP 64:a=PEEK(&8000+i):IF a=0 OR >CF
a=207 THEN 1080 ELSE NEXT i
1080 nom1$=UPPER$(nom1$):FOR j=1 TO 8:a$=MID$(UPPER$(no >WK
m1$), j, 1):IF a$="" THEN a$=CHR$(32)
-1090 POKE &8000+i+(j-1), ASC(a$):NEXT j:POKE &8000+i+8, t >GL
y:ad1$=MID$(x1$, 1, 2):ad4$=MID$(x1$, 3, 2):POKE &8000+i+9,
VAL("&"+ad1$):POKE &8000+i+10, VAL("&"+ad4$):ad4$=MID$(x
2$, 3, 2):ad1$=MID$(x2$, 1, 2):POKE &8000+i+11, VAL("&"+ad1$
):POKE &8000+i+12, VAL("&"+ad4$)
1100 POKE &8000+i+13, mo:ad4$=MID$(ad2$, 3, 2):ad1$=MID$(a >RC
d2$, 1, 2):POKE &8000+i+14, VAL("&"+ad1$):POKE &8000+i+15,
VAL("&"+ad4$):FOR h=1 TO 48:a$=MID$(b$, h, 1):IF a$="" TH
EN a$=CHR$(32)
1110 POKE &8000+i+15+h, ASC(a$):NEXT h >DJ
1120 SAVE "bibiotec", B, &8000, 3840:IF era=0 THEN !ERA, "b >MW
ibiotec.bak":GOTO 190 ELSE era=0:GOTO 190
1130 CLS#1:TAG:MOVE 285, 334:PRINT"ARCHIVES";:TAGOFF:PEN >KT
0:LOAD "bibiotec.bin", &8000:PEN 1:LOCATE 4, 4:PRINT"- 1
- CONSULTER UN FICHIER":LOCATE 4, 7:PRINT"- 2 - EFFACER
UN FICHIER":LOCATE 4, 10:PRINT"- 3 - RETOUR MENU":aa$=""
"
1140 CLEAR INPUT:WHILE NOT(aa$="1" OR aa$="2" OR aa$="3 >FG
"):aa$=INKEY$:WEND:CLS:IF aa$="3" THEN 190
1150 WINDOW 11, 70, 8, 22:CALL &9AB0:WINDOW 8, 73, 8, 22:g=PE >UF
EK(&9AAF):LOCATE 2, 15:CLEAR INPUT:PRINT" Entrez le nume
rot du fichier => ";:INPUT "", a:WHILE (a<=0 OR a>g):LO
CATE 33, 15:INPUT "", a:WEND:CLS:bi=1:IF aa$="1" THEN 970

```



```
1160 a=(a-1)*64:bi=0:nom$="":FOR j=0 TO 7:i=PEEK(&8000+ >ZB
a+j):nom$=nom$+CHR$(i):NEXT:FOR i=0 TO 7:POKE (&8000+a+
1),207:NEXT:FOR j=1 TO 63:POKE (&8000+a+j),0:NEXT:CLS:L
OCATE 4,6:PRINT"Fichier ";nom$;" efface"
1170 PEN 0:ERA,nom$+".bin":SAVE "bibliotec",b,&8000,384 >LW
0:ERA,"bibliotec.bak":PEN 1:GOTO 1230
1180 a=(a-1)*64:bi=0:nom$="":FOR j=0 TO 7:i=PEEK(&8000+ >FC
a+j):nom$=nom$+CHR$(i):NEXT j:LOCATE 8,1:PRINT (nom$):t
y=PEEK(&8000+a+8):LOCATE 9,3:IF ty=0 THEN PRINT"FENETRE
GRAPHIQUE" ELSE PRINT"PAGE ECRAN"
1190 TAG:ad1$=HEX$(PEEK(&8000+a+9),2):ad2$=HEX$(PEEK(&8 >ZF
000+a+10),2):ad3$=ad1$+ad2$:a$=ad3$:MOVE 256,158:PRINT"
"&ad3$:MOVE 336,158:PRINT VAL("&"+ad3$):ad1$=HEX$(PE
EK(&8000+a+11),2):ad2$=HEX$(PEEK(&8000+a+12),2):MOVE 42
0,158:PRINT "XXXXXX":MOVE 420,132
1200 PRINT "XXXXXX":ad3$=ad1$+ad2$:MOVE 256,132:PRINT >MB
"&ad3$:MOVE 336,132:PRINT VAL("&"+ad3$):ad1$=HEX$(P
EEK(&8000+a+14),2):ad2$=HEX$(PEEK(&8000+a+15),2):ad3$=a
d1$+ad2$:MOVE 256,104:PRINT"&ad3$:MOVE 336,104:PRINT
VAL("&"+ad3$);
1210 a$=STR$(VAL("&"+ad3$)):IF LEN(a$)>=5 THEN b$=LEFT$ >GZ
(a$,2):b=VAL(b$)+1 ELSE b=1
1220 MOVE 434,104:PRINT b;:mo=PEEK(&8000+a+13):TAGOFF:L >KF
OCATE 46,1:PRINT mo:LOCATE 17,5:FOR i=1 TO 48:PRINT CHR
$(PEEK(&8000+a+15+i)):NEXT i
1230 LOCATE 2,15:PRINT"<ESPACE> Pour le menu (E) >AE
NTRER> Pour la liste des noms":CLEAR INPUT
1240 a$=INKEY$:IF a$=" " THEN 190 ELSE IF a$=CHR$(13) T >DQ
HEN CLS:GOTO 1150 ELSE 1240
1250 TAG:MOVE 280,334:PRINT"OPERATIONS";:TAGOFF:LOCATE >KA
6,3:PRINT"- 1 - CALCULATRICE":LOCATE 6,5:PRINT"- 2 -
CONVERSION DECIMAL/HEXADECIMAL":LOCATE 6,7:PRINT"- 3 -
CALCUL D'UNE ADRESSE ECRAN":LOCATE 6,9
1260 PRINT"- 4 - CALCUL D'ADRESSES MEMOIRE":LOCATE 6,1 >WX
1:PRINT"- 5 - RETOUR MENU PRINCIPAL":CLEAR INPUT:a$=""
:WHILE NOT(a$="1" OR a$="2" OR a$="3" OR a$="4" OR a$="
5"):a$=INKEY$:WEND:CLS:CLS#1:IF a$="2" THEN 1330 ELSE I
F a$="3" THEN 1360
1270 IF a$="4" THEN 1470 ELSE IF a$="5" THEN 190 ELSE T >KC
AG:MOVE 268,334:PRINT"CALCULATRICE";:TAGOFF:LOCATE 2,1:
PRINT b18$:LOCATE 2,3:PRINT b3$:LOCATE 2,5:PRINT"1 - Ad
dition":LOCATE 2,7
1280 PRINT"2 - Soustraction":LOCATE 2,9:PRINT"3 - Multi >YB
plication":LOCATE 2,11:PRINT"4 - Divission":LOCATE 2,13
:PRINT"RESULTAT DECIMAL ":LOCATE 33,13:PRINT"RESULTAT
HEXADECIMAL ":LOCATE 2,1:PRINT " ";b18$;:INPUT "",nobr1
$:LOCATE 2,1:PRINT " ";b18$
1290 LOCATE 2,3:PRINT " ";b3$;:INPUT "",nobr2$:LOCATE 2, >GQ
3:PRINT " ";b3$:a$="":CLEAR INPUT:WHILE NOT(a$="1" OR a$
="2" OR a$="3" OR a$="4"):a$=INKEY$:WEND:IF a$="1" THEN
res=VAL(nobr1$)+VAL(nobr2$)
1300 IF a$="2" THEN res=VAL(nobr1$)-VAL(nobr2$) ELSE IF >RP
a$="3" THEN res=VAL(nobr1$)*VAL(nobr2$) ELSE IF a$="4"
THEN res=VAL(nobr1$)/VAL(nobr2$)
1310 LOCATE 20,13:PRINT res:LOCATE 56,13:PRINT"&";HEX$( >YB
res):LOCATE 2,15:PRINT"<ESPACE> pour le menu
<ENTRER> pour continuer":a$="":CLEAR INPUT:WHILE NOT(a$
=" " OR a$=CHR$(13)):a$=INKEY$:WEND:IF a$=" " THEN 190
ELSE IF a$=CHR$(13) THEN CLS
```

```
1320 GOTO 1270 >MD
1330 TAG:MOVE 204,334:PRINT"CONVERSION DECIMAL/HEXADECI >NC
MAL";:TAGOFF:LOCATE 2,8:PRINT"RESULTAT DECIMAL ":LOCAT
E 33,8:PRINT"RESULTAT HEXADECIMAL ":LOCATE 2,4:PRINT"E
ntrez le nombre => ";:INPUT "",nobr1$
1340 IF VAL(nobr1$)>65000 THEN CLS:GOTO 1330 ELSE LOCAT >FD
E 20,8:PRINT VAL(nobr1$):LOCATE 56,8:PRINT"&";HEX$(VAL(
nobr1$)):LOCATE 2,15:PRINT"<ESPACE> pour le menu
<ENTRER> pour continuer"
1350 a$="":CLEAR INPUT:WHILE NOT(a$=" " OR a$=CHR$(13)) >FP
:a$=INKEY$:WEND:IF a$=" " THEN 190 ELSE IF a$=CHR$(13)
THEN CLS:GOTO 1330
1360 LOCATE#1,21,1:PRINT#1,"CALCUL D'UNE ADRESSE ECRAN" >KF
:LOCATE 2,6:PRINT" Desirez vous appeller un ecran pour d
eterminer les coordonnees O/N ":a$="":CLEAR INPUT:WHILE
NOT(a$="O" OR a$="N"):a$=UPPER$(INKEY$):WEND:CLS:IF a$
="O" THEN 1400
1370 LOCATE 2,2:PRINT b6$:LOCATE 2,4:PRINT b12$:LOCATE >AF
2,8:PRINT"ADRESSE ECRAN ":LOCATE 2,2:PRINT " ";b6$;:INP
UT "",xx:LOCATE 2,2:PRINT " ";b6$:LOCATE 2,4:PRINT " ";b1
2$;:INPUT "",yy:LOCATE 2,4:PRINT " ";b12$
1380 f1=((400-(yy+1))/16):f2=FIX(f1):f1=f1-f2:GOSUB 720 >PX
:x1$=HEX$(49152+(f2*80)+(f3*2048)+FIX(xx/8)):LOCATE 18,
8:PRINT"&";HEX$(VAL("&"+x1$)):LOCATE 2,15:PRINT"<ESPACE
> pour le menu (ENTRER) pour continuer"
1390 a$="":CLEAR INPUT:WHILE NOT(a$=" " OR a$=CHR$(13)) >FX
:a$=INKEY$:WEND:IF a$=" " THEN 190 ELSE IF a$=CHR$(13)
THEN CLS:GOTO 1360
1400 LOCATE 4,2:PRINT " ";b9$;:INPUT "",nom1$:LOCATE 4,2 >BK
:PRINT " ";b9$:LOCATE 4,4:PRINT " ";b2$;" ":a$="":CLEAR I
NPUT:WHILE NOT(a$="O" OR a$="N"):a$=UPPER$(INKEY$):WEND
:IF a$="O" THEN 1420 ELSE LOCATE 4,4
1410 aa=1:PRINT b2$:LOCATE 2,6:PRINT " ";b4$;:INPUT "",m >JW
o:LOCATE 2,6:PRINT " ";b4$
1420 CLS:LOCATE 25,3:PRINT"MODE D'EMPLOI":LOCATE 2,6:PR >XB
INT"- DEPLACEMENT DE LA CROIX => touches de deplacement
du curseur":LOCATE 2,9:PRINT"- VALIDATION DE LA POSITI
ON => touche <ESPACE>":LOCATE 21,12:PRINT"<APPUYEZ SUR
UNE TOUCHE>":CALL &BB06
1430 CALL &A094,&7500,&25A0:PEN 0:IF aa=1 THEN aa=0:MOD >LY
E mo:BORDER bo,bo1:FOR i=0 TO me:INK i,pe(i),pe1(i):NEX
T:LOAD nom1$,&C000:PEN 1 ELSE LOAD nom1$,&7500:PEN 1:P
AGE,&7500,0
1440 CALL &9D2A:ccc=1:GOSUB 190:ccc=0:x=PEEK(&9D26):x1= >PU
PEEK(&9D27):y=PEEK(&9D28):y1=PEEK(&9D29):x$=HEX$(x):x1$
=HEX$(x1):xx$=x1$+x$:xx=VAL("&"+xx$):y$=HEX$(y):y1$=HEX
$(y1):yy$=y1$+y$:yy=VAL("&"+yy$)
1450 LOCATE#1,21,1:PRINT#1,"CALCUL D'UNE ADRESSE ECRAN" >PF
:LOCATE 2,2:PRINT b6$;xx:LOCATE 2,4:PRINT b12$;yy:LOCAT
E 2,8:PRINT"ADRESSE ECRAN ":f1=((400-(yy+1))/16):f2=FI
X(f1):f1=f1-f2:GOSUB 720:x1$=HEX$(49152+(f2*80)+(f3*204
8)+FIX(xx/8))
1460 LOCATE 18,8:PRINT"&";HEX$(VAL("&"+x1$)):LOCATE 2,1 >EK
5:PRINT"<ESPACE> pour le menu (ENTRER) pour c
ontinuer":a$="":CLEAR INPUT:WHILE NOT(a$=" " OR a$=CHR$
(13)):a$=INKEY$:WEND:IF a$=" " THEN 190 ELSE IF a$=CHR$
(13) THEN CLS:GOTO 1360
1470 TAG:MOVE 223,334:PRINT"CALCUL D'ADRESSES MEMOIRE"; >CC
:TAGOFF:LOCATE 4,2:PRINT"Fixez l'adresse inferieur et s
```



```
uperieur dans la bank 0":LOCATE 4,7:PRINT b1$:LOCATE 4,
5:PRINT " ;b10$;:INPUT "",inf$:LOCATE 4,5:PRINT " ;b10$
;:LOCATE 4,7
1480 PRINT " ;b1$;:INPUT "",sup$:PRINT " :CLS:LOCATE 21 >YC
,15:PRINT"< Entrez 0 pour sortir >":WINDOW 8,73,8,20:y=
2:FOR i=1 TO 100:LOCATE 4,y:PRINT " ;b20$;i;" => " ;:INP
UT "",fic$(i):IF VAL(fic$(i))<0 THEN ERROR 6 ELSE LOCAT
E 4,y:PRINT " ;b20$;i;" => "
1490 y=y+2:IF y=14 THEN y=2:CLS >ZM
1500 IF VAL(fic$(i))<0 THEN NEXT i ELSE WINDOW 8,73,8, >DE
22:CLS:b$=MID$(sup$,1,1):IF b$="&" THEN IF VAL(inf$)<0
THEN inf$=STR$(32768-(ABS(CINT(VAL(inf$))))+32768)
1510 IF b$="&" THEN IF VAL(sup$)<0 THEN sup$=STR$(32768 >PF
-(ABS(CINT(VAL(sup$))))+32768)
1520 a=VAL(sup$)-VAL(inf$):a$="V":b=0:IF a<=0 THEN a=16 >LR
384:a$="W"
1530 FOR j=1 TO (i-1):IF (MID$(fic$(j),1,1)="V" OR MID$ >CL
(fic$(j),1,1)="W" OR MID$(fic$(j),1,1)="X" OR MID$(fic$
(j),1,1)="Y" OR MID$(fic$(j),1,1)="Z") THEN 1540 ELSE I
F VAL(fic$(j))<a THEN a=a-(VAL(fic$(j))+2):b=b+1:fic$(j
)=a$+fic$(j):IF a<=0 THEN i=i
1540 NEXT j:IF b<i THEN a=16384:a$=CHR$(ASC(a$)+1):IF a >JZ
$="[" THEN 1550 ELSE GOTO 1530
1550 adr0=VAL(inf$):adr1=0:adr2=&4000:adr3=&8000:adr4=& >AE
C000:y=1:x=1:WINDOW 8,48,8,20:MOVE 580,50,1,0:DRAW 580,
270:MOVE 525,270:DRAW 525,50:FOR j=0 TO 4:MOVE 521,50+(
j*55):DRAW 580,50+(j*55):NEXT:MOVE 515,50:DRAW 515,270:
MOVE 460,270:DRAW 460,50
1560 FOR j=0 TO 4:MOVE 456,50+(j*55):DRAW 515,50+(j*55) >AY
:NEXT:TAG:MOVE 462,290:PRINT"BANK 0";:MOVE 529,290:PRIN
T"BANK 1";:MOVE 410,270:PRINT"&FFFF";:MOVE 410,215:PRIN
T"&C000";:MOVE 410,160:PRINT"&8000";:MOVE 410,105:PRINT
"&4000";:MOVE 440,60
1570 PRINT"&0";:TAGOFF:yy1=0:yy2=55:yy3=110:yy4=165:IF >DD
VAL(inf$)<0 THEN xx=(110-(ABS(CINT(VAL(inf$)/297.89)))+
110) ELSE xx=CINT(VAL(inf$)/297.89)
1580 FOR j=1 TO (i-1):b$=MID$(fic$(j),1,1):b=VAL(MID$(f >EB
ic$(j),2))+1:IF b$="V" THEN bank=0:adr=adr0 ELSE IF b$=
"W" THEN bank=1:adr=adr1 ELSE IF b$="X" THEN bank=1:adr
=adr2 ELSE IF b$="Y" THEN bank=1:adr=adr3 ELSE IF b$="Z
" THEN bank=1:adr=adr4
1590 LOCATE 2,y:PRINT"FICHIER Num. ";j;" Adr. ";&";HEX$( >TG
adr);" Bank";bank:y=y+1:b=b+1:x=x+1:IF b$="V" THEN adr
0=adr0+b ELSE IF b$="W" THEN adr1=adr1+b ELSE IF b$="X"
THEN adr2=adr2+b ELSE IF b$="Y" THEN adr3=adr3+b ELSE
IF b$="Z" THEN adr4=adr4+b
1600 fic$(j)=MID$(fic$(j),2):a=CINT(VAL(fic$(j))/297.89 >GA
):IF b$="V" THEN FOR h=1 TO a STEP 2:MOVE 460,50+xx+h:D
RAW 515,50+xx+h:NEXT:xx=xx+(h-1) ELSE IF b$="W" THEN xx
1=yy1 ELSE IF b$="X" THEN xx1=yy2 ELSE IF b$="Y" THEN x
x1=yy3 ELSE IF b$="Z" THEN xx1=yy4
1610 IF NOT(b$="V") THEN FOR h=1 TO a STEP 2:MOVE 525,5 >MA
0+xx1+h:DRAW 580,50+xx1+h:NEXT:IF b$="W" THEN yy1=yy1+(
h-1) ELSE IF b$="X" THEN yy2=yy2+(h-1) ELSE IF b$="Y" T
HEN yy3=yy3+(h-1) ELSE IF b$="Z" THEN yy4=yy4+(h-1) ELS
E yy4=221
1620 IF yy4>220 THEN PRINT" FICHIERS TROP IMPORTANTS... >NU
!":GOTO 1640 ELSE IF x=14 THEN WINDOW 8,73,8,22:LOCATE
1,15:PRINT"<Appuyez sur une touche pour obtenir la suit
```

```
e>":CLEAR INPUT:CALL &BBO6:LOCATE 1,15:PRINT STRING$(46
," "):WINDOW 8,48,8,20:x=1
1630 NEXT >LB
1640 ERASE fic$:DIM fic$(60):WINDOW 8,73,8,22:LOCATE 1, >AU
15:PRINT"< Appuyez sur une touche pour le tableau >":CL
EAR INPUT:CALL &BBO6:CLS:MOVE 100,290,0:DRAW 588,290:MO
VE 100,288:DRAW 588,288:LOCATE 22,2:PRINT "ESPACE RESTA
NT EN MEMOIRE":MOVE 200,100,1,0
1650 DRAW 440,100:DRAW 440,220:DRAW 200,220:DRAW 200,10 >YR
0:FOR i=0 TO 100 STEP 20:MOVE 200,100+i:DRAW 440,100+i:
NEXT:MOVE 250,200:DRAW 250,100:MOVE 300,200:DRAW 300,10
0:MOVE 204,99:DRAW 441,99:FOR i=0 TO 2:MOVE 441+i,99:DR
AW 441+i,217:NEXT:TAG:MOVE 208,216
1660 PRINT"BANK";:MOVE 258,216:PRINT"BLOC";:MOVE 308,21 >TB
6:PRINT"OCTETS RESTANTS";:MOVE 224,196:PRINT"0";:MOVE 2
62,196:PRINT"XXX";:FOR i=0 TO 3:MOVE 224,176-(i*20):PR
INT"1";:MOVE 264,176-(i*20):PRINT i;:NEXT:MOVE 312,196:
PRINT VAL(sup$)-adr0;" octets";
1670 MOVE 312,176:PRINT 16384-adr1;" octets";:MOVE 312, >MF
156:PRINT 16384-(adr2-16384);" octets";:MOVE 312,136:PR
INT ABS(adr3)-16384;" octets";:MOVE 312,116:PRINT ABS(a
dr4);" octets";:TAGOFF
1680 LOCATE 15,15:PRINT"< Appuyez sur une touche pour l >WX
e MENU >":CLEAR INPUT:CALL &BBO6:GOTO 190
1690 IF DERR=146 AND ERL=1060 THEN FOR i=0 TO 59:FOR j= >JJ
0 TO 7:POKE (&8000+(i*64)+j),207:NEXT j,i:era=1:i=0:RES
UME 1080 ELSE SOUND 1,88,15,15:FOR i=1 TO 160:NEXT:SOUN
D 1,88,15,15
1700 MODE 2:WINDOW 8,73,8,22:PAPER 0:PEN 1:INK 0,in0(nn >HA
):INK 1,in1(nn):BORDER in0(nn):LOCATE 19,4
1710 IF ERR=6 THEN der=1 ELSE IF ERR=13 OR ERR=5 THEN d >ZM
er=2 ELSE IF ERR=11 THEN der=3 ELSE IF ERR=100 THEN der
=2
1720 IF ERR=32 THEN der=(DERR AND &7F) >EJ
1730 IF der=1 THEN PRINT"NOMBRE TROP GRAND...!" >TE
1740 IF der=2 THEN PRINT"ARGUMENT INCORRECT...!" >UB
1750 IF der=3 THEN PRINT"PAS DE DIVISION PAR ZERO...!" >BL
1760 IF der=16 THEN PRINT"NOM DE FICHIER INCORRECT...!" >CU
1770 IF der=18 THEN PRINT"CE FICHIER N'EXISTE PAS...!" >AQ
1780 IF der=19 THEN PRINT"CATALOGE SATURE SAUVEGARDE IM >XT
POSSIBLE...!"
1790 IF der=20 THEN PRINT"DISQUETTE PLEINE SAUVEGARDE I >YG
MPOSSIBLE...!"
1800 IF der=66 THEN PRINT"ECRITURE IMPOSSIBLE DISQUETTE >XW
PROTEGE...!"
1810 FOR i=1 TO 4000:NEXT i:CLS:cc=0:cc1=0:ec=0:PEN 0:P >EA
RINT FRE("):RESUME 190
1820 DATA 0,25,3,2,17,13,22,5,14,19,26,24,1,6,11,23 * >QM
1830 DATA 13,0,12,0,16,0,14,0,15,0,0,14,0,17,0,22,0,15, >ZD
0,16
1840 TAG:MOVE 262,334:PRINT"NETTOYAGE BASIC";:TAGOFF:LO >JF
CATE 5,4:PRINT"INTRODUISEZ LA DISQUETTE CONTENANT LE PR
OGRAMME ^COMPACT^":LOCATE 32,7:PRINT"ET":LOCATE 22,10:P
RINT"APPUYEZ SUR UNE TOUCHE":CLEAR INPUT:CALL &BBO6:MEM
ORY &9B00:CLEAR
1850 POKE &A36F,101:RUN "compact1.bin" >KB
```

UTILITAIRE

# COMPACT

## 2ème partie

Vous voulez réduire la taille de vos pages-écrans ou déplacer des fenêtres graphiques ? Compact vous y aidera grâce à une batterie de RSX. Mais attention, il s'agit de la seconde partie et le mode d'emploi est contenu dans le numéro précédent, donc pas de précipitation : il vous faut impérativement les deux parties pour faire tourner Compact.

Valable pour  
CPC 464  
✕ CPC 664  
✕ CPC 6128





## UTILISATION DES COMMANDES

• L'utilisation des nouvelles commandes est très facile, mais afin de les utiliser avec un maximum d'efficacité, quelques connaissances dans la structure de la RAM sont requises. C'est pourquoi nous en ferons un bref rappel dans la rubrique suivante. Les commandes seront ensuite étudiées et détaillées une par une et largement illustrées d'exemples.

### Rappel sur l'organisation de la RAM des CPC

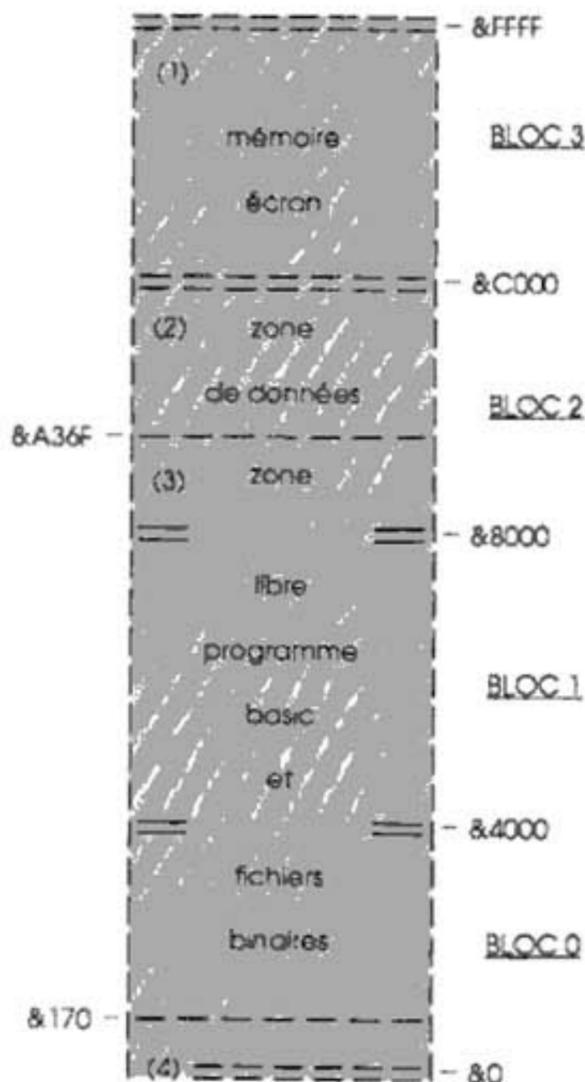
COMPACT a été créé pour le CPC 6128 mais toutefois la deuxième BANK n'étant pas utilisée par le programme, celui-ci est compatible avec le CPC 664. Néanmoins sa vocation étant d'être utilisé sur 6128, c'est l'organisation de la RAM d'un 6128 que nous allons étudier ici.

Pour ceux d'entre vous qui se poseraient la question de savoir ce qu'est la RAM, on peut tout simplement dire qu'il s'agit de la mémoire sur laquelle on peut écrire et lire des données. Par contre, la ROM est la mémoire qui ne peut être que lue. Le 6128 possède une RAM de 128 Ko divisée en 2 BANK de 64 Ko elles-mêmes divisées en 4 blocs de 16 Ko chacun, numérotés de 0 à 3.

### La BANK 0

L'écran exploite le bloc 3 de la BANK 0 (1), c'est-à-dire de l'adresse &C000 à &FFFF. La partie supérieure du bloc 2 de la BANK 0 est utilisée comme zone de branchement, zone de données BASIC ainsi que comme zone de logement des caractères définis par l'utilisateur (2). Elle se situe entre &A36F et &BFFF. La partie inférieure du bloc 0 de la BANK 0 est utilisée comme zone de copie de la ROM inférieure (4) elle est située de &0 à &170.

Il nous reste donc une zone comprise entre les adresses &170 et &A36F (3), zone réservée pour vos programmes BASIC et vos fichiers binaires. L'encodage de lignes de programme BASIC commence à l'adresse &170 (368 en décimal) et se poursuit en montant jusqu'à la fin du programme. C'est à cet endroit qu'il vous faudra fixer la commande MEMORY (voir manuel de l'utilisateur), les fichiers binaires pour-



ront être logés dans la partie immédiatement au-dessus. Si votre programme BASIC utilise toute la place disponible en BANK 0, les fichiers devront être logés en BANK 1.

Oui, mais comment connaître l'adresse la plus élevée utilisée par le BASIC ?

Une méthode simple vous est donnée ici. Provoquez un RESET, chargez votre programme et entrez toutes les

variables numériques, alphanumériques et les tableaux contenus dans votre programme.

EX : a=10:nobr=3000:nom\$="COU-COU":a\$="casserole":DIM a\$(300):

Tapez ENTER, puis cette ligne :

PRINT HEX\$(HIMEM-FRE(0)) + 4000)

Le résultat obtenu est l'adresse à laquelle il vous faudra fixer votre MEMORY. ATTENTION : ne pas oublier de variables sinon en cours de programme vous risquez de tomber sur le message : Memory full. Si par la suite, vous effectuez une modification sur votre programme, il est préférable de renouveler l'opération.

### La BANK 1

La deuxième BANK du 6128 est située parallèlement à la BANK 0, c'est-à-dire qu'elle est comprise entre les adresses &0000 et &FFFF, elle est constituée de manière identique, soit quatre blocs de 16 K octets chacun.

La difficulté pour y accéder réside dans le fait que le cœur de votre ordinateur, qui est le microprocesseur Z80, ne permet de gérer que 64 K octets à la fois. On ne peut y parvenir que par un moyen détourné. Ce moyen est la commutation des blocs, c'est-à-dire que nous allons remplacer un bloc situé en BANK 0 par un bloc situé en BANK 1.

Cette précision est très importante car en effet si la commutation ne peut se faire que bloc par bloc, on ne





pourra gérer que 16384 octets à la fois : particularité qui amènera certaines restrictions à la commande |CHARGE décrite plus loin.

Autre particularité de la BANK 1 : lors de l'initialisation de l'ordinateur, c'est-à-dire lors de l'allumage ou après un RESET, la BANK 1 n'est pas vide. En effet, au lieu d'être à 0 les octets sont soit à 0 soit à 255, détail qui pourrait nuire au bon fonctionnement des nouvelles commandes. C'est pour pallier cet inconvénient qu'a été créée la commande |NETTOIE étudiée plus loin.

## LES COMMANDES

- Les commandes sont au nombre de cinq. Elles permettent une parfaite gestion des fichiers créés par COMPACT. Elles sont accessibles à partir de nouveaux «mots clés» appelés RSX ou à partir de l'instruction BASIC CALL. La principale différence entre les deux, c'est que «CALL» est plus court de 59 octets, et si vous disposez d'un désassembleur, il peut être incorporé à un programme en assembleur. Ces commandes seront étudiées sous leurs deux formes.

Le programme «RSX» a une taille de 1098 octets et il est situé de l'adresse &9F24 à l'adresse &A36D. Le programme «CALL» a une taille de 1039 octets. Il est situé de l'adresse &9F5F à l'adresse &A36D. Il faudra donc tenir compte de ces adresses afin d'éviter tout effacement par le chargement d'un fichier.

Pour charger les commandes, faites pour le programme «RSX» :

```
LOAD «RSX.BIN»:CALL &9F24
```

Pour le programme «CALL» :

```
LOAD «CALL.BIN»
```

ATTENTION, il est fortement déconseillé de charger deux fois de suite les programmes «RSX» ou «CALL» sans effectuer de RESET entre les deux, car ceci désactiverait les commandes propres à AMSDOS (|ERA, |REN, |A...)

A noter que toutes les commandes sont accompagnées de paramètres. Si l'un d'entre eux est oublié, la commande ne sera pas exécutée. Il se

pourrait que lors d'une fausse manœuvre de votre part, un message d'erreur propre à COMPACT apparaisse à l'écran. Il vous faudra vous reporter à la rubrique «les messages d'erreurs de COMPACT». En effet, les programmes «RSX» et «CALL» sont dotés de messages d'erreurs qui, ajoutés aux messages du BASIC, permettent une parfaite utilisation des commandes. Ces messages peuvent être activés ou désactivés à volonté. Il suffit pour les désactiver de mettre à 0 l'octet situé à l'adresse &9F63 et de le mettre à 1 pour les activer.

EX : POKE &9F63,0 ou POKE &9F63,1

Si les messages sont désactivés et qu'une erreur a été détectée, aucun message ne sera envoyé mais la commande ne sera pas exécutée. Je vous propose donc de les activer avant de passer aux commandes suivantes.

Nous allons maintenant étudier la terminologie et la syntaxe de chaque commande. Après initialisation de votre ordinateur, introduisez votre disquette et tapez :

```
MEMORY &4000:LOAD «RSX.BIN»:CALL &9F24:POKE &9F63,1
```

### La commande NETTOIE

```
|NETTOIE, <adresse de début>, <taille>, <bank>  
CALL &A30C, <adresse de début>,- <taille>, <bank>
```

COMMANDE : efface une partie de la mémoire.

Paramètre 1 : <adresse de début>

Spécifie l'adresse à laquelle le nettoyage doit débuter.

Paramètre 2 : <taille>

Spécifie la longueur sur laquelle l'effacement doit être effectué. Si le «nettoyage» se situe en BANK 1 la taille de l'effacement ne doit pas être supérieure à 16384 octets ou &4000 en hexadécimal (voir rappel «l'organisation des CPC») sous peine de débordement en BANK 0 (voir exemple).

Paramètre 3 <bank>

Spécifie sur quelle BANK le «nettoyage» doit être effectué.

Exemple de ce qu'il ne faut pas faire :

```
|NETTOIE, &9000, &1000, 0 (1)  
|NETTOIE, &0, 20000, 1 (2)  
|NETTOIE, &8000, &4010, 1 (2)
```

(1) = Effacement du programme «RSX»  
(2) = Débordement en BANK 0

Exemple de ce qu'il faut faire, tapez ces lignes :

```
|NETTOIE, &4000, &5F00, 0  
|NETTOIE, &0, 16384, 1  
|NETTOIE, 16384, &4000, 1  
|NETTOIE, &8000, &4000, 1
```

Cette commande s'utilise donc pour effacer les blocs de la BANK 1, après une initialisation et pour effacer les fichiers se trouvant à un emplacement mémoire sur lequel on désire charger un autre fichier.

### La commande CHARGE

```
|CHARGE, <nom du fichier>, <adresse de relogement>, <bank>  
CALL &A28A, <nom du fichier>, <adresse de relogement>, <bank>
```

COMMANDE : charge en mémoire un fichier binaire se trouvant sur la disquette. Cette commande a la même fonction que l'instruction BASIC LOAD, mais possède un plus très important : elle permet de charger des fichiers et des programmes binaires dans la BANK 1.

Paramètre 1 : nom du fichier

Spécifie le nom du fichier à charger. Le nom peut être accompagné de son indicateur de type ou remplacé par une variable alphanumérique.

Paramètre 2 : adresse de relogement

Spécifie à quelle adresse doit être chargé le fichier.

Paramètre 3 : bank

Spécifie dans quelle BANK doit être chargé le fichier. Si la BANK dans laquelle doit être effectué le chargement est la BANK 1, le fichier à charger ne doit pas dépasser la longueur de 16384 octets (voir précédemment «Rappel sur l'organisation des CPC» et «La commande NETTOIE»).



### La commande PAGE

**| PAGE**, <adresse du fichier>, <bank>  
**CALL &9FA2**, <adresse du fichier>, <bank>

**COMMANDE** : «décompacte» et affiche à l'écran les fichiers traités avec COMPACT.

**Paramètre 1** : <adresse du fichier>

Spécifie l'adresse à laquelle la commande doit chercher le fichier.

**Paramètre 2** : <bank>

Spécifie dans quelle BANK se situe le fichier.

Remarquez que lorsque deux instructions PAGE se suivent, il est nécessaire de mettre une boucle de temporisation ou un arrêt de déroulement du programme entre les deux sinon le décompactage est trop rapide et la première page reste invisible. Tapez ces lignes :

```
|PAGE, &4868, 0:FOR I=0 TO 1000:|PAGE,&1308,1 |PAGE, &5DE9, 0:PRINT «APPUYEZ SUR UNE TOUCHE»:CALL &BB06:|PAGE, &9675,0
```

### La commande FENETRE

**| FENETRE**, <adresse écran de relogement>, <adresse du fichier>, <bank>  
**CALL &9FAC**, <adresse écran de relogement>, <adresse du fichier>, <bank>

**COMMANDE** : «décompacte» et affiche à l'écran les fichiers traités avec COMPACT.

**Paramètre 1** : <adresse écran de relogement>

Spécifie l'adresse écran de coin supérieur gauche sur lequel la fenêtre doit apparaître. La valeur 0 permet de faire apparaître la fenêtre à l'emplacement sur lequel la fenêtre a été compactée.

**Paramètre 2** : <adresse du fichier>

Spécifie l'adresse à laquelle la commande doit chercher le fichier.

**ATTENTION** : toute adresse dépassant le cadre de la mémoire écran serait susceptible d'effacer un fichier en mémoire et pourrait même provoquer le plantage de votre ordinateur. Si cela

se produisait, le seul remède serait l'initialisation complète de votre machine par RESET.

**Paramètre 3** : bank

Spécifie dans quelle BANK se situe le fichier.

### La commande ADR

**|ADR**, <adresse écran>, <variable numérique entière>, <direction>  
**CALL &A16D**, <adresse écran>, <variable numérique entière>, <direction>

**COMMANDE** : calcule la valeur de l'adresse écran configuré à l'adresse spécifiée par le premier paramètre dans la direction déterminée par le troisième paramètre et charge le résultat dans la variable numérique entière nommée par le deuxième paramètre. La structure de la mémoire écran étant ce quelle est, il est très difficile de calculer l'adresse d'un octet périphérique à un autre. Cette commande s'en charge pour vous.

**Paramètre 1** : <adresse écran>

Spécifie l'adresse écran par rapport à laquelle le calcul devra être effectué.

**Paramètre 2** : <variable numérique entière>

Spécifie dans quelle variable le résultat doit être chargé, cette variable doit être une variable numérique entière.

Elle doit être précédée par l'instruction BASIC VARPTR (@) et doit obligatoirement être suivie de l'indicateur de type (%) ou alors l'instruction BASIC DEFINT doit figurer au début de votre programme (voir manuel de l'utilisateur).

**Paramètre 3** : indicateur de direction

Spécifie dans quelle direction se trouve l'octet dont l'adresse est à déterminer.

- 0 : Haut
- 1 : Haut droite
- 2 : Droite
- 3 : Droite bas
- 4 : Bas
- 5 : Bas gauche

- 6 : Gauche
- 7 : Gauche haut

Pour une meilleure compréhension, tapez ce programme :

```
10 MODE 1
20 b=&DBE8
30 a%=0
40 FOR I=0 TO 7
50 |ADR, &DBE8,@a%,I
60 POKE a%, 15
70 LOCATE 1,1:PRINT «Indicateur de direction N» ; I
80 LOCATE 1,3:PRINT «Adresse écran du point => &»:HEX$(a%)
90 LOCATE 1,5:PRINT «APPUYEZ SUR UNE TOUCHE»
100 CALL &BB06
110 NEXT I
RUN
```

Maintenant tapez la ligne ci-dessous, remplacez sur tout le programme la variable (%) par (a) et relancez le programme.

```
25 DEFINT a
```

## LES MESSAGES D'ERREURS DE COMPACT

COMPACT est doté d'un jeu de quatre messages destinés à signaler une erreur commise sur une instruction de COMPACT. Ces messages ont été créés dans le but d'éviter pratiquement tout plantage de vos programmes.

### - ADRESSE ERRONEE

Une adresse mémoire accompagnant l'instruction PAGE ou FENETRE ne correspond pas à un fichier.

### - FICHIER FENETRE

Une adresse mémoire accompagnant l'instruction PAGE correspond à un fichier FENETRE.

### - FICHIER PAGE

Une adresse mémoire accompagnant l'instruction FENETRE correspond à un fichier PAGE.

### - ARGUMENT INCORRECT

Le paramètre d'une commande n'est pas acceptable.

Eric CINGET



# DATA 1

```
1 ' ***** Generateur du fichier COMPA.BIN
2 FOR i=&9AB0 TO &A35B
3 READ a$:a=VAL("&"a$)
4 POKE i,a:NEXT i
5 SAVE "COMPA",B,&9AB0,&A35B-&9AB0
6 DATA 3E,01,32,AF,9A,3E,20,32
7 DATA AE,9A,3E,31,32,AD,9A,21
8 DATA 00,80,06,08,3E,20,CD,5A
9 DATA BB,3E,31,CD,5A,BB,3E,20
10 DATA CD,5A,BB,3E,18,CD,5A,BB
11 DATA 7E,CD,5A,BB,23,05,20,F8
12 DATA 06,38,23,05,20,FC,3E,18
13 DATA CD,5A,BB,3E,3A,CD,5A,BB
14 DATA 7E,FE,00,C8,3A,AD,9A,3C
15 DATA FE,3A,20,17,3A,AE,9A,FE
16 DATA 20,20,07,3E,31,32,AE,9A
17 DATA 18,07,3A,AE,9A,3C,32,AE
18 DATA 9A,3E,30,32,AD,9A,3A,AE
19 DATA 9A,CD,5A,BB,3A,AD,9A,CD
20 DATA 5A,BB,3E,20,CD,5A,BB,3E
21 DATA 18,CD,5A,BB,3A,AF,9A,3C
22 DATA 32,AF,9A,06,08,18,A1,00
```

```
23 DATA 00,DD,7E,00,21,00,C0,01
24 DATA 00,40,ED,A1,E2,50,9B,CC
25 DATA 4C,9B,18,F6,32,38,9B,C9
26 DATA CC,4C,9B,C9,00,00,00,CD
27 DATA 11,BC,32,56,9B,C9,00,00
28 DATA 00,00,00,00,00,00,00,00
29 DATA 00,00,00,00,00,00,00,00
30 DATA 00,00,00,00,00,00,00,00
31 DATA 00,00,00,00,00,00,00,DD
32 DATA 7E,00,32,71,9B,DD,7E,02
33 DATA 32,70,9B,DD,7E,04,32,6F
34 DATA 9B,DD,7E,06,32,6E,9B,DD
35 DATA 7E,08,32,6D,9B,DD,7E,0A
36 DATA 32,6C,9B,DD,7E,0C,32,6B
37 DATA 9B,DD,7E,0E,32,6A,9B,DD
38 DATA 7E,10,32,69,9B,DD,7E,12
39 DATA 32,68,9B,DD,7E,14,32,67
40 DATA 9B,DD,7E,16,32,66,9B,DD
41 DATA 7E,18,32,65,9B,DD,7E,1A
42 DATA 32,64,9B,DD,7E,1C,32,63
43 DATA 9B,DD,7E,1E,32,62,9B,DD
44 DATA 6E,20,DD,66,21,22,72,9B
45 DATA DD,6E,22,DD,66,23,22,74
46 DATA 9B,DD,6E,24,DD,66,25,22
47 DATA 7B,9B,22,77,9B,DD,6E,26
48 DATA DD,66,27,22,7D,9B,22,79
49 DATA 9B,CD,11,BC,06,00,88,28
```

```
50 DATA 09,06,01,B8,28,08,3E,01
51 DATA 18,06,3E,04,18,02,3E,02
52 DATA 32,76,9B,ED,5B,77,9B,2A
53 DATA 79,9B,CD,F0,BB,06,00,B8
54 DATA 28,4B,06,01,B8,28,4B,06
55 DATA 02,B8,28,4B,06,03,B8,28
56 DATA 4B,06,04,B8,28,4B,06,05
57 DATA B8,28,4B,06,06,B8,28,4B
58 DATA 06,07,B8,28,4B,06,08,B8
59 DATA 28,4B,06,09,B8,28,4B,06
60 DATA 0A,B8,28,4B,06,0B,B8,28
61 DATA 4B,06,0C,B8,28,4B,06,0D
62 DATA B8,28,4B,06,0E,B8,28,4B
63 DATA 06,0F,B8,28,4B,3A,62,9B
64 DATA 18,49,3A,63,9B,18,44,3A
65 DATA 64,9B,18,3F,3A,65,9B,18
66 DATA 3A,3A,66,9B,18,35,3A,67
67 DATA 9B,18,30,3A,68,9B,18,2B
68 DATA 3A,69,9B,18,26,3A,6A,9B
69 DATA 18,21,3A,6B,9B,18,1C,3A
70 DATA 6C,9B,18,17,3A,6D,9B,18
71 DATA 12,3A,6E,9B,18,0D,3A,6F
72 DATA 9B,18,08,3A,70,9B,18,03
73 DATA 3A,71,9B,CD,DE,BB,ED,5B
74 DATA 77,9B,2A,79,9B,CD,EA,BB
75 DATA ED,5B,77,9B,3A,76,9B,67
76 DATA 13,ED,53,77,9B,ED,4B,72
```



77 DATA 9B, 7A, B8, 28, 02, 18, 04, 7B  
78 DATA B9, 28, 06, 25, 20, EA, C3, 23  
79 DATA 9C, ED, 5B, 7B, 9B, ED, 53, 77  
80 DATA 9B, 2A, 79, 9B, 23, 23, 22, 79  
81 DATA 9B, ED, 4B, 74, 9B, 7C, B8, 28  
82 DATA 03, C3, 23, 9C, 7D, B9, 28, 0B  
83 DATA 2A, 79, 9B, 7D, 3D, B9, 28, 03  
84 DATA C3, 23, 9C, C9, 00, 00, 00, 00  
85 DATA 00, 00, 3E, 01, CD, 59, BC, 26  
86 DATA 01, 2E, 01, CD, 3F, BB, 21, 40  
87 DATA 01, 22, 26, 9D, 21, C8, 00, 22  
88 DATA 28, 9D, CD, 4A, 9D, CD, 5D, 9D  
89 DATA 18, 28, 11, 00, 00, 2A, 28, 9D  
90 DATA CD, C0, BB, 11, 80, 02, 2A, 28  
91 DATA 9D, CD, F6, BB, C9, ED, 5B, 26  
92 DATA 9D, 21, 00, 00, CD, C0, BB, ED  
93 DATA 5B, 26, 9D, 21, 90, 01, CD, F6  
94 DATA BB, C9, 3E, 01, CD, 1E, BB, 28  
95 DATA 0D, CD, 5D, 9D, 2A, 26, 9D, 23  
96 DATA 22, 26, 9D, CD, 5D, 9D, 3E, 08  
97 DATA CD, 1E, BB, 28, 0D, CD, 5D, 9D  
98 DATA 2A, 26, 9D, 2B, 22, 26, 9D, CD  
99 DATA 5D, 9D, 3E, 00, CD, 1E, BB, 28  
100 DATA 0D, CD, 4A, 9D, 2A, 28, 9D, 23  
101 DATA 22, 28, 9D, CD, 4A, 9D, 3E, 02  
102 DATA CD, 1E, BB, 28, 0D, CD, 4A, 9D  
103 DATA 2A, 28, 9D, 2B, 22, 28, 9D, CD  
104 DATA 4A, 9D, 3E, 2F, CD, 1E, BB, CA  
105 DATA 72, 9D, 26, 1E, 2E, 02, CD, 3F  
106 DATA BB, 3E, 00, CD, 59, BC, C9, 00  
107 DATA 00, 00, 00, 00, 00, 00, 00, 00  
108 DATA 00, 00, 00, 00, 00, 00, 00, 00  
109 DATA DD, 46, 05, DD, 4E, 04, ED, 43  
110 DATA DC, 9D, DD, 46, 03, DD, 4E, 02  
111 DATA ED, 43, DA, 9D, DD, 7E, 00, 32  
112 DATA E0, 9D, ED, 5B, DA, 9D, 21, FF  
113 DATA BF, 3E, 00, 32, E7, 9D, 01, 00  
114 DATA 00, ED, 43, DE, 9D, 23, 7E, 32  
115 DATA E3, 9D, 23, 7E, 32, E4, 9D, 23  
116 DATA 7E, 32, E5, 9D, 23, 7E, 32, E6  
117 DATA 9D, 2B, 2B, 2B, C3, AB, 9E, 2A  
118 DATA E1, 9D, 3A, E3, 9D, 47, 3A, E4  
119 DATA 9D, B8, 28, 02, 18, 16, 3A, E4  
120 DATA 9D, 47, 3A, E5, 9D, B8, 28, 02  
121 DATA 18, 0A, 3A, E5, 9D, 47, 3A, E6  
122 DATA 9D, B8, 28, 10, 3A, E3, 9D, 13  
123 DATA 12, ED, 4B, DE, 9D, 03, ED, 43  
124 DATA DE, 9D, 18, B1, 3E, 01, 32, E7  
125 DATA 9D, 06, FF, 3A, E7, 9D, B8, 28  
126 DATA 0F, 23, 18, 4A, 2A, E1, 9D, 7E  
127 DATA 47, 3A, E3, 9D, B8, 28, 23, 2B  
128 DATA 3A, E0, 9D, 13, 12, 3A, E7, 9D  
129 DATA 13, 12, 3A, E3, 9D, 13, 12, 3E  
130 DATA 00, 32, E7, 9D, ED, 4B, DE, 9D  
131 DATA 03, 03, 03, ED, 43, DE, 9D, C3  
132 DATA 15, 9E, 3A, E7, 9D, 3C, 32, E7  
133 DATA 9D, 18, BE, 22, E1, 9D, 3E, FF  
134 DATA BC, 28, 03, C3, 2F, 9E, 3E, FF  
135 DATA BD, 28, 14, C3, 2F, 9E, 22, E1

136 DATA 9D, 3E, FF, BC, 28, 02, 18, AC  
137 DATA 3E, FF, BD, 28, 12, 18, A5, 3A  
138 DATA E3, 9D, 13, 12, ED, 4B, DE, 9D  
139 DATA 03, ED, 43, DE, 9D, 18, 1A, 3A  
140 DATA E0, 9D, 13, 12, 3A, E7, 9D, 13  
141 DATA 12, 3A, E3, 9D, 13, 12, ED, 4B  
142 DATA DE, 9D, 03, 03, 03, ED, 43, DE  
143 DATA 9D, ED, 4B, DE, 9D, 03, ED, 5B  
144 DATA DC, 9D, 79, 12, 13, 78, 12, C9  
145 DATA FF, 42, 00, 00, 00, 00, 00, 00  
146 DATA 00, 00, 00, 00, 00, 00, 00, 00  
147 DATA 00, 00, 00, 00, 00, 00, DD, 7E  
148 DATA 0A, 32, 13, 9F, 32, 14, 9F, DD  
149 DATA 46, 09, DD, 4E, 08, ED, 43, 15  
150 DATA 9F, DD, 46, 07, DD, 4E, 06, ED  
151 DATA 43, 17, 9F, DD, 46, 05, DD, 4E  
152 DATA 04, ED, 43, 0C, 9F, DD, 46, 03  
153 DATA DD, 4E, 02, ED, 43, 0A, 9F, DD  
154 DATA 7E, 00, 32, 10, 9F, ED, 5B, 0A  
155 DATA 9F, 2A, 15, 9F, 3E, 00, 32, 1D  
156 DATA 9F, 01, 00, 00, ED, 43, 0E, 9F  
157 DATA 7E, 32, 19, 9F, 23, 7E, 32, 1A  
158 DATA 9F, 23, 7E, 32, 18, 9F, 23, 7E  
159 DATA 32, 1C, 9F, 2B, 2B, 2B, C3, 0B  
160 DATA A0, 3A, 19, 9F, 47, 3A, 1A, 9F  
161 DATA B8, 28, 02, 18, 16, 3A, 1A, 9F  
162 DATA 47, 3A, 1B, 9F, B8, 28, 02, 18  
163 DATA 0A, 3A, 1B, 9F, 47, 3A, 1C, 9F  
164 DATA B8, 28, 11, 3A, 19, 9F, 13, 12  
165 DATA ED, 4B, 0E, 9F, 03, ED, 43, 0E  
166 DATA 9F, 23, 18, B4, 3E, 01, 32, 1D  
167 DATA 9F, 23, 06, FF, 3A, 1D, 9F, B8  
168 DATA 2B, 0A, 18, 65, 7E, 47, 3A, 19  
169 DATA 9F, B8, 28, 22, 3A, 10, 9F, 13  
170 DATA 12, 3A, 1D, 9F, 13, 12, 3A, 19  
171 DATA 9F, 13, 12, 3E, 00, 32, 1D, 9F  
172 DATA ED, 4B, 0E, 9F, 03, 03, 03, ED  
173 DATA 43, 0E, 9F, C3, 68, 9F, 3A, 1D  
174 DATA 9F, 3C, 32, 1D, 9F, 22, 11, 9F  
175 DATA 3A, 14, 9F, 3D, CC, 3F, A0, 32  
176 DATA 14, 9F, 2A, 11, 9F, 23, 18, B2  
177 DATA 22, 11, 9F, ED, 4B, 17, 9F, 79  
178 DATA BD, 28, 02, 18, 04, 78, BC, 28  
179 DATA 3A, 3A, 14, 9F, 3D, CC, 3F, A0  
180 DATA 32, 14, 9F, 2A, 11, 9F, C3, 81  
181 DATA 9F, 22, 11, 9F, ED, 4B, 17, 9F  
182 DATA 79, BD, 28, 02, 18, 04, 78, BC  
183 DATA 28, 29, 2A, 11, 9F, 18, 85, 2A  
184 DATA 15, 9F, CD, 26, BC, 22, 15, 9F  
185 DATA 2B, 22, 11, 9F, 3A, 13, 9F, 32  
186 DATA 14, 9F, C9, 3A, 19, 9F, 13, 12  
187 DATA ED, 4B, 0E, 9F, 03, ED, 43, 0E  
188 DATA 9F, 18, 1E, 3A, 10, 9F, 13, 12  
189 DATA 3A, 1D, 9F, 13, 12, 3A, 19, 9F  
190 DATA 13, 12, 7E, 13, 12, ED, 4B, 0E  
191 DATA 9F, 03, 03, 03, 03, ED, 43, 0E  
192 DATA 9F, ED, 4B, 0E, 9F, 03, ED, 5B  
193 DATA 0C, 9F, 79, 12, 13, 78, 12, C9  
194 DATA FF, 56, 41, 49, DD, 4E, 00, DD

195 DATA 46, 01, DD, 6E, 02, DD, 66, 03  
196 DATA 3E, 00, ED, B1, E2, AC, A0, 28  
197 DATA 00, 77, 18, F6, C9, 00, 00, 00  
198 DATA 00, 00, 00, 00, 01, BE, A0, 21  
199 DATA B0, A0, CD, D1, BC, C9, C6, A0  
200 DATA C3, 09, A1, C3, 10, A1, 50, 41  
201 DATA 47, C5, 46, 45, 4E, 45, 54, 52  
202 DATA C5, 00, 00, 00, 00, 00, 00, 00  
203 DATA 41, 44, 52, 45, 53, 53, 45, 20  
204 DATA 45, 52, 52, 4F, 4E, 45, 45, 45  
205 DATA 52, 2E, 46, 48, 43, 48, 49, 45  
206 DATA 52, 20, 46, 45, 4E, 45, 54, 52  
207 DATA 45, 45, 52, 2E, 20, 46, 49, 43  
208 DATA 48, 49, 45, 52, 20, 50, 41, 47  
209 DATA 45, 3E, 01, 32, D2, A0, 18, 05  
210 DATA 3E, 00, 32, D2, A0, DD, 7E, 00  
211 DATA 32, D3, A0, 06, 00, B8, 28, 03  
212 DATA C3, AB, A1, DD, 6E, 02, DD, 66  
213 DATA 03, 22, D4, A0, 7E, 06, 67, B8  
214 DATA 20, 1A, 23, 7E, 06, 31, B8, 20  
215 DATA 13, 23, 7E, 06, C9, B8, 20, 0C  
216 DATA 23, 7E, 06, 7E, B8, 28, 32, 06  
217 DATA D9, B8, 28, 2D, 06, 0F, 21, D8  
218 DATA A0, 78, 32, D7, A0, 3A, D6, A0  
219 DATA 06, 00, B8, 28, 0B, 3A, D7, A0  
220 DATA 47, 7E, CD, 5A, BB, 23, 10, F9  
221 DATA 3A, D3, A0, 06, 00, B8, C8, 0E  
222 DATA C0, 3E, C0, B1, 06, 7F, ED, 49  
223 DATA C9, 3A, D2, A0, 06, 00, B8, 28  
224 DATA 0D, 7E, 06, 7E, B8, 28, 14, 06  
225 DATA 12, 21, E7, A0, 18, C3, 7E, 06  
226 DATA D9, B8, 28, 07, 06, 10, 21, F9  
227 DATA A0, 18, B6, 23, 22, D4, A0, 3A  
228 DATA D2, A0, 06, 00, B8, CA, C5, A2  
229 DATA C3, FD, A1, DD, 7E, 03, 06, 40  
230 DATA B8, DA, B8, A1, 06, 80, 18, 04  
231 DATA 0E, C4, 18, 28, B8, DA, C4, A1  
232 DATA 06, C0, 18, 04, 0E, C5, 18, 13  
233 DATA B8, DA, D0, A1, 06, FF, 18, 06  
234 DATA 0E, C6, 06, 40, 18, 04, 0E, C7  
235 DATA 06, 80, 90, 67, DD, 6E, 02, 22  
236 DATA D4, A0, 18, 09, C6, 40, 67, DD  
237 DATA 6E, 02, 22, D4, A0, 2A, D4, A0  
238 DATA 3E, C0, B1, 06, 7F, ED, 49, C3  
239 DATA 2C, A1, 00, 00, 00, 2A, D4, A0  
240 DATA 7E, 23, 4E, 23, 46, 22, D4, A0  
241 DATA ED, 43, FA, A1, 32, FC, A1, 23  
242 DATA 7E, E5, F5, CD, 0E, BC, F1, E1  
243 DATA 06, 00, B8, 28, 09, 06, 01, B8  
244 DATA 28, 08, 06, 03, 18, 06, 06, 1F  
245 DATA 18, 02, 06, 07, 23, 05, 20, FC  
246 DATA 46, 23, 4E, 3E, 00, C5, E5, CD  
247 DATA 32, BC, E1, 23, 46, 23, 4E, E5  
248 DATA CD, 38, BC, 3E, 0F, E1, C1, E5  
249 DATA C5, F5, CD, 32, BC, F1, C1, 3D  
250 DATA 20, F6, E1, 11, 00, C0, ED, 4B  
251 DATA FA, A1, 3A, FC, A1, 23, 23, 23  
252 DATA ED, A1, 28, 09, E2, 7E, A2, 03  
253 DATA 2B, ED, A0, 18, F3, C5, F5, 46



254 DATA 23,7E,12,13,05,20,FB,F1  
 255 DATA C1,23,0B,0B,18,E2,2A,D4  
 256 DATA A0,23,7E,06,00,B8,28,09  
 257 DATA 06,01,B8,28,08,3E,01,18  
 258 DATA 06,3E,0F,18,02,3E,03,23  
 259 DATA 46,23,4E,F5,E5,CD,32,BC  
 260 DATA E1,F1,3D,20,F2,23,23,23  
 261 DATA 23,E5,C1,03,0A,67,03,0A  
 262 DATA 6F,CD,3E,BC,3A,D3,A0,06  
 263 DATA 00,B8,C8,C3,6F,A1,00,00  
 264 DATA 00,00,00,00,00,2A,D4,A0  
 265 DATA DD,5E,04,DD,56,05,ED,53  
 266 DATA C1,A2,7E,F5,23,7E,32,BE  
 267 DATA A2,3C,32,BF,A2,23,4E,23  
 268 DATA 46,23,3E,00,BA,28,07,ED  
 269 DATA 5B,C1,A2,23,18,06,BA,20  
 270 DATA F6,5E,23,56,ED,53,C1,A2  
 271 DATA F1,23,ED,A1,28,0C,E2,4C  
 272 DATA A3,03,2B,CD,1F,A3,ED,A0  
 273 DATA 18,F0,C5,F5,46,23,7E,7F  
 274 DATA CD,1F,A3,12,13,05,20,F7  
 275 DATA F1,C1,23,0B,0B,18,DB,32  
 276 DATA C0,A2,3A,BF,A2,3D,28,07  
 277 DATA 32,BF,A2,3A,C0,A2,C9,3A  
 278 DATA BE,A2,32,BF,A2,22,C3,A2  
 279 DATA 2A,C1,A2,CD,26,BC,22,C1  
 280 DATA A2,ED,5B,C1,A2,2A,C3,A2  
 281 DATA 3A,C0,A2,C9,3A,D3,A0,06  
 282 DATA 00,C8,0E,C0,3E,02,32,D2  
 283 DATA A0,C3,6F,A1,00,00,00,00

26 DATA 12,32,65,9F,3E,00,32,60  
 27 DATA 9F,21,90,9F,18,2D,DD,6E  
 28 DATA 02,DD,66,03,22,61,9F,7E  
 29 DATA FE,67,20,16,23,7E,FE,31  
 30 DATA 20,10,23,7E,FE,C9,20,0A  
 31 DATA 23,7E,FE,7E,28,2F,FE,D9  
 32 DATA 28,2B,06,0F,21,66,9F,78  
 33 DATA 32,65,9F,3A,63,9F,FE,00  
 34 DATA 28,0B,3A,65,9F,47,7E,CD  
 35 DATA 5A,BB,23,10,F9,3A,60,9F  
 36 DATA FE,00,C8,0E,C0,3E,C0,B1  
 37 DATA 06,7F,ED,49,C9,3A,5F,9F  
 38 DATA FE,00,28,0C,7E,FE,7E,28  
 39 DATA 13,06,0F,21,75,9F,18,C7  
 40 DATA 7E,FE,D9,28,07,06,0C,21  
 41 DATA 84,9F,18,BB,23,22,61,9F  
 42 DATA 3A,64,9F,FE,00,28,03,CD  
 43 DATA 19,BD,3A,5F,9F,FE,00,CA  
 44 DATA EC,A1,C3,B3,A0,DD,7E,03  
 45 DATA 47,DD,4E,02,ED,43,61,9F  
 46 DATA 69,FE,40,DA,6C,A0,18,04  
 47 DATA 0E,C4,18,20,FE,80,DA,77  
 48 DATA A0,18,04,0E,C5,18,12,FE  
 49 DATA C0,DA,82,A0,18,06,0E,C6  
 50 DATA 06,40,18,04,0E,C7,06,80  
 51 DATA 90,67,18,03,C6,40,67,22  
 52 DATA 61,9F,3E,C0,B1,06,7F,ED  
 53 DATA 49,3A,5F,9F,FE,0A,CA,ED  
 54 DATA A2,FE,14,2A,61,9F,CA,53  
 55 DATA A3,C3,DB,9F,00,00,00,2A  
 56 DATA 61,9F,7E,23,4E,23,46,22  
 57 DATA 61,9F,ED,43,B0,A0,32,B2  
 58 DATA A0,23,7E,E5,F5,CD,0E,BC  
 59 DATA F1,E1,FE,00,28,08,FE,01  
 60 DATA 28,08,06,03,18,06,06,1F  
 61 DATA 18,02,06,07,23,10,FD,46  
 62 DATA 23,4E,3E,00,C5,E5,CD,32  
 63 DATA BC,E1,23,46,23,4E,E5,CD  
 64 DATA 38,BC,3E,0F,E1,C1,E5,C5  
 65 DATA F5,CD,32,BC,F1,C1,3D,20  
 66 DATA F6,E1,11,00,C0,ED,4B,80  
 67 DATA A0,3A,B2,A0,23,23,23,ED  
 68 DATA A1,28,09,E2,30,A1,03,2B  
 69 DATA ED,A0,18,F3,C5,F5,46,23  
 70 DATA 7E,12,13,10,FC,F1,C1,23  
 71 DATA 0B,0B,18,E3,2A,61,9F,23  
 72 DATA 7E,FE,00,28,08,FE,01,28  
 73 DATA 08,3E,01,18,06,3E,0F,18  
 74 DATA 02,3E,03,23,46,23,4E,F5  
 75 DATA E5,CD,32,BC,E1,F1,3D,20  
 76 DATA F2,23,23,23,23,E5,C1,03  
 77 DATA 0A,67,03,0A,6F,CD,3E,BC  
 78 DATA 3A,60,9F,FE,00,C8,C3,17  
 79 DATA A0,FE,03,C0,DD,66,05,DD  
 80 DATA 6E,04,DD,56,03,DD,5E,02  
 81 DATA ED,53,61,9F,DD,7E,00,FE  
 82 DATA 00,28,1F,FE,01,28,20,FE  
 83 DATA 02,28,24,FE,03,28,25,FE  
 84 DATA 04,28,29,FE,05,28,2A,FE

85 DATA 06,28,2E,FE,07,28,2F,C3  
 86 DATA C3,9F,CD,29,BC,18,2D,CD  
 87 DATA 29,BC,CD,20,BC,18,25,CD  
 88 DATA 20,BC,18,20,CD,20,BC,CD  
 89 DATA 26,BC,18,18,CD,26,BC,18  
 90 DATA 13,CD,26,BC,CD,23,BC,18  
 91 DATA 0B,CD,23,BC,18,06,CD,23  
 92 DATA BC,CD,29,BC,ED,5B,61,9F  
 93 DATA 7D,12,7C,13,12,C9,00,00  
 94 DATA 00,00,00,00,00,00,00,00  
 95 DATA 2A,61,9F,DD,5E,04,DD,56  
 96 DATA 05,ED,53,E8,A1,7E,F5,23  
 97 DATA 7E,32,E5,A1,3C,32,E6,A1  
 98 DATA 23,4E,23,46,23,3E,00,BA  
 99 DATA 28,07,ED,5B,E8,A1,23,18  
 100 DATA 06,BA,20,F6,5E,23,56,ED  
 101 DATA 53,E8,A1,F1,23,ED,A1,28  
 102 DATA 0C,E2,72,A2,03,2B,CD,45  
 103 DATA A2,ED,A0,18,F0,C5,F5,46  
 104 DATA 23,7E,7F,CD,45,A2,12,13  
 105 DATA 10,F8,F1,C1,23,0B,0B,18  
 106 DATA DC,32,E7,A1,3A,E6,A1,3D  
 107 DATA 28,07,32,E6,A1,3A,E7,A1  
 108 DATA C9,3A,E5,A1,32,E6,A1,22  
 109 DATA EA,A1,2A,E8,A1,CD,26,BC  
 110 DATA 22,E8,A1,ED,5B,E8,A1,2A  
 111 DATA EA,A1,3A,E7,A1,C9,3A,60  
 112 DATA 9F,FE,00,C8,C3,17,A0,00  
 113 DATA 00,00,00,00,00,00,00,00  
 114 DATA 00,00,00,00,00,00,FE,03  
 115 DATA C0,DD,7E,00,32,7C,A2,FE  
 116 DATA 02,DA,9B,A2,C3,C3,9F,DD  
 117 DATA 46,03,DD,4E,02,ED,43,61  
 118 DATA 9F,DD,46,05,DD,4E,04,0A  
 119 DATA 32,7B,A2,FE,00,CA,C3,9F  
 120 DATA FE,0D,38,03,C3,C3,9F,03  
 121 DATA 0A,6F,03,0A,67,3A,7B,A2  
 122 DATA 4F,06,00,11,7E,A2,ED,B0  
 123 DATA 3A,7C,A2,FE,00,28,0D,3E  
 124 DATA 0A,32,5F,9F,ED,4B,61,9F  
 125 DATA 78,C3,60,A0,21,7E,A2,3A  
 126 DATA 7B,A2,47,11,00,C0,CD,77  
 127 DATA BC,28,0D,2A,61,9F,CD,83  
 128 DATA BC,28,05,CD,7A,BC,18,03  
 129 DATA CD,7D,BC,3A,7C,A2,FE,00  
 130 DATA C8,C3,17,A0,01,00,00,10  
 131 DATA FE,03,C0,DD,7E,00,32,0B  
 132 DATA A3,FE,02,DA,1D,A3,C3,C3  
 133 DATA 9F,DD,4E,02,DD,46,03,ED  
 134 DATA 43,0A,A3,3E,00,B8,38,06  
 135 DATA B9,38,03,C3,C3,9F,ED,4B  
 136 DATA 0A,A3,DD,6E,04,DD,66,05  
 137 DATA 22,61,9F,3A,08,A3,FE,00  
 138 DATA 28,0D,3E,14,32,5F,9F,ED  
 139 DATA 4B,61,9F,7B,C3,60,A0,ED  
 140 DATA 4B,0A,A3,3E,00,11,6E,A3  
 141 DATA EB,ED,A0,2B,E2,65,A3,18  
 142 DATA FB,3A,08,A3,FE,00,C8,C3  
 143 DATA 17,A0,00,00,00,00,00,00

# DATA2

```
1 * ***** Generateur du programme RSX
2 FOR I=&9F24 TO &A36D
3 READ a$:a=VAL("&"*a$)
4 POKE I,a:NEXT I
5 SAVE "RSX",B,&9F24,&A36D-&9F24
6 DATA FC,A6,32,9F,01,32,9F,21
7 DATA 24,9F,CD,D1,BC,C9,43,9F
8 DATA C3,A2,9F,C3,AC,9F,C3,8A
9 DATA A2,C3,0C,A3,C3,6D,A1,50
10 DATA 41,47,C5,46,45,4E,45,54
11 DATA 52,C5,43,48,41,52,47,C5
12 DATA 4E,45,54,54,4F,49,C5,41
13 DATA 44,D2,00,14,00,00,40,01
14 DATA 01,12,41,44,52,45,53,53
15 DATA 45,20,45,52,52,4F,4E,45
16 DATA 45,46,49,43,48,49,45,52
17 DATA 20,46,45,4E,45,54,52,45
18 DATA 46,49,43,48,49,45,52,20
19 DATA 50,41,47,45,41,52,47,55
20 DATA 4D,45,4E,54,20,49,4E,43
21 DATA 4F,52,52,45,43,54,FE,02
22 DATA C0,3E,01,32,5F,9F,18,0B
23 DATA FE,03,C0,3E,00,32,5F,9F
24 DATA DD,7E,00,32,60,9F,FE,00
25 DATA 28,14,FE,01,CA,59,A0,3E
```