

## 5/10.3.1

# Compactage filiforme

---

Nous allons étudier :

- 1) Le principe de compactage d'objets filiformes.
- 2) Le principe de décompactage et d'affichage de fichiers filiformes.

### I. Le compacteur

Les dessins monochromes définis par leurs contours (dessins du genre « fil de fer ») pourront être énormément compactés (jusqu'à un facteur 20 !) par le programme qui va suivre.

Le procédé de compactage est très simple. Il consiste en :

- la définition d'un point quelconque appartenant au dessin,
- le codage de la direction dans laquelle il faut se déplacer pour atteindre le point suivant.

Le programme de compactage filiforme défini est écrit en BASIC et occupe les lignes 1200 à 1450.

Entrez le nom de l'écran à compacter. Cet écran aura été créé par le programme de tracé défini au chapitre 10.1 de la partie 5.

Déplacez ensuite le curseur graphique (grâce aux touches-flèches) jusqu'à rencontrer un point allumé sur le dessin.

Appuyez sur la touche « ENTER ». Le programme trace alors d'une autre couleur (PEN 2) le contour de la forme jusqu'à aboutir à une discontinuité. Arrivé à ce point, le tracé s'arrête. Il faut alors déplacer le curseur pour « sauter » la discontinuité. Dès que le curseur se trouve sur un autre point (voisin) du dessin, appuyez sur la touche « ENTER », et ainsi de suite jusqu'à ce que tout l'objet ait changé de couleur.

Appuyez alors deux fois sur la touche « ESC ». Le programme indique la place occupée par le fichier compacté et propose une sauvegarde magnétique ou un retour au compactage (appui sur « ENTER »).

Le programme de compactage est le suivant :

```
1000 REM *****
1010 REM Codage de formes
1020 REM *****
1030 '
1040 'Initialisation
1050 '
1060 'Prog. ASM Sauvegarde et affichage ecran
1070 FOR I=0 TO &17:READ A:POKE &2F00+I,A:NEXT
1080 DATA &21,0,&C0,&11,0,&40,1,&FF,&3F,&ED,&B0,&C9
1090 DATA &21,0,&40,&11,0,&C0,1,&FF,&3F,&ED,&B0,&C9
1100 '
1110 ON BREAK GOSUB 1670 'Sortie du programme
1120 INK 0,0:INK 1,10:INK 3,6,25:BORDER 0:MODE 1
1130 INPUT "Nom de l'ecran a coder ";N$:LOAD N$,&C000
1140 W=10:AG=&3000 'Adresse graphique
1150 GOTO 1490 'Positionnement en debut de Forme
1160 AG=&3000:V1=INT(X/256):V2=X-V1*256:V3=INT(Y)/256:V4=Y-V3*256:POKE AG,V1:POK
E AG+1,V2:POKE AG+2,V3:POKE AG+3,V4:AG=&3003 'Entete
1170 '
1180 ' Codage de la forme en memoire
1190 '
1200 mx=x:my=y
1210 PLOT X,Y,2
1220 W=9' 'Init du calcul
1230 IF TEST(X+2,Y)=1 THEN U=X+2:V=Y:D=0:GOSUB 1430
1240 IF TEST(X+2,Y+2)=1 THEN U=X+2:V=Y+2:D=1:GOSUB 1430
1250 IF TEST(X,Y+2)=1 THEN U=X:V=Y+2:D=2:GOSUB 1430
1260 IF TEST(X-2,Y+2)=1 THEN U=X-2:V=Y+2:D=3:GOSUB 1430
1270 IF TEST(X-2,Y)=1 THEN U=X-2:V=Y:D=4:GOSUB 1430
1280 IF TEST(X-2,Y-2)=1 THEN U=X-2:V=Y-2:D=5:GOSUB 1430
1290 IF TEST(X,Y-2)=1 THEN U=X:V=Y-2:D=6:GOSUB 1430
1300 IF TEST(X+2,Y-2)=1 THEN U=X+2:V=Y-2:D=7:GOSUB 1430
1310 IF W=9 THEN 1490
1320 X=WX:Y=WY:PT1=0
```

```
1330 IF G=1 THEN G=0 ELSE G=1
1340 IF G=1 THEN OCT=16*WD ELSE OCT=OCT+WD:AG=AG+1:POKE AG,OCT
1350 IF W=0 THEN PLOT x,y,2:GOTO 1370 'Saut de plume
1360 GOTO 1210
1370 IF G=1 THEN OCT=OCT OR &80 ELSE OCT=OCT OR 8
1380 IF g=1 THEN ag=ag+1
1390 POKE AG,OCT:GOTO 1490
1400 '
1410 'Calcul optimal du prochain point
1420 '
1430 W1=- (TEST (U+2,V)=1) - (TEST (U+2,V+2)=1) - (TEST (U,V+2)=1) - (TEST (U-2,V+2)=1) - (TEST (U-2,V)=1) - (TEST (U-2,V-2)=1) - (TEST (U,V-2)=1) - (TEST (U+2,V-2)=1)
1440 IF W1<W THEN W=W1:WX=U:WY=V:WD=D
1450 RETURN
1460 '
1470 'Positionnement du curseur
1480 '
1490 WX=0:WY=0
1500 IF PT1=1 THEN AG=AG+1:POKE AG,&88 'Point unique
1510 cou=TEST(x,y):PLOT x,y,3
1520 A#=INKEY#:IF A#="" THEN 1520
1530 A=ASC(A#)
1540 PLOT X,Y,COU
1550 IF A=240 THEN Y=Y+2:WY=WY+2
1560 IF A=241 THEN Y=Y-2:WY=WY-2
1570 IF A=242 THEN X=X-2:WX=WX-2
1580 IF A=243 THEN X=X+2:WX=WX+2
1590 IF a<>13 THEN 1510
1600 IF WX>=0 THEN AX=WX/2 ELSE AX=(-WX/2) OR &80
1610 IF WY>=0 THEN AY=WY/2 ELSE AY=(-WY/2) OR &80
1620 AG=AG+1:POKE AG,AX:AG=AG+1:POKE AG,AY:g=0:ag=ag+1
1630 PT1=1:GOTO 1200
```

```

1640 '
1650 'Sortie du programme
1660 '
1670 CALL &2F00 'Sauvegarde ecran
1680 CLS:PRINT"La forme occupe la memoire situee"
1690 PRINT"entre &3000 et ";HEX$(AG+1);"."
1700 PRINT:INPUT"Nom de la sauvegarde (ou ENTER) ";R$
1710 IF LEN(R$)=0 THEN CALL &2F0C:RETURN 'Pas de sauvegarde
1720 POKE AG+1,&FF:SAVE R$,B,&3000,AG+2-&3000 'Sauvegarde
1730 END

```

Lignes 1070 à 1090 : Chargement des sous-programmes Assembleur

Lignes 1110 à 1160: Initialisation du programme

Lignes 1200 à 1390: Codage de la forme

Lignes 1430 à 1450: Calcul optimal du prochain point

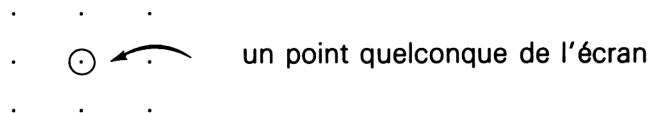
Lignes 1490 à 1630: Positionnement du curseur graphique sur le prochain départ

Ligne 1710 : Retour au compactage si appui sur ENTER

Ligne 1720 : ou sortie avec sauvegarde du fichier compacté

Une technique intéressante employée dans ce programme est *la recherche du prochain point à allumer en créant le moins possible de discontinuités*. Cette technique est basée sur le principe suivant :

Tout point de l'écran peut être entouré de huit façons différentes :

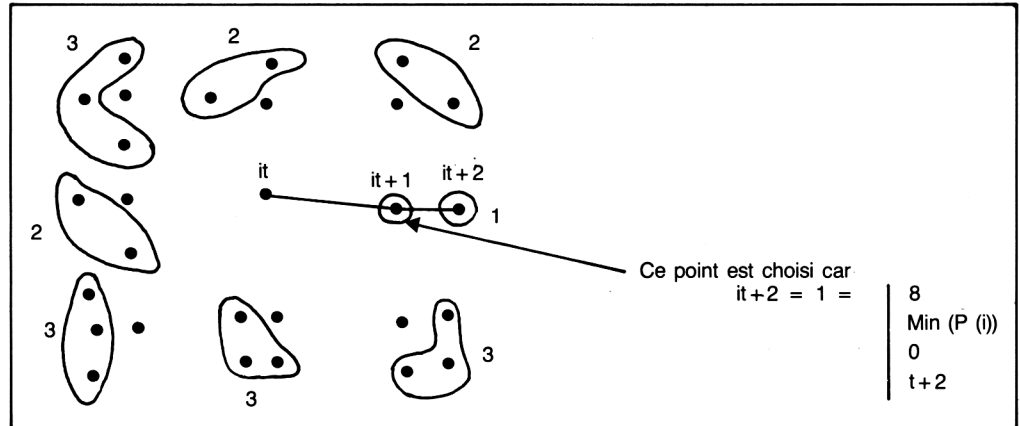


Appelons « t » le contexte actuel, « t + 1 » le contexte après un déplacement, et « t + n » le contexte après n déplacements.

Si, pour chacun de ces huit points, nous calculons le nombre de points immédiatement contigus P(i) (et donc le nombre de déplacements possibles en t + 2), il apparaît que :

$$\min_{i=0}^8 (P(i))$$

donnera le point i ayant le moins de chance de provoquer une discontinuité (Calcul de Min (P(i)) effectué ligne 1440, et calcul de P(i) effectué ligne 1430).

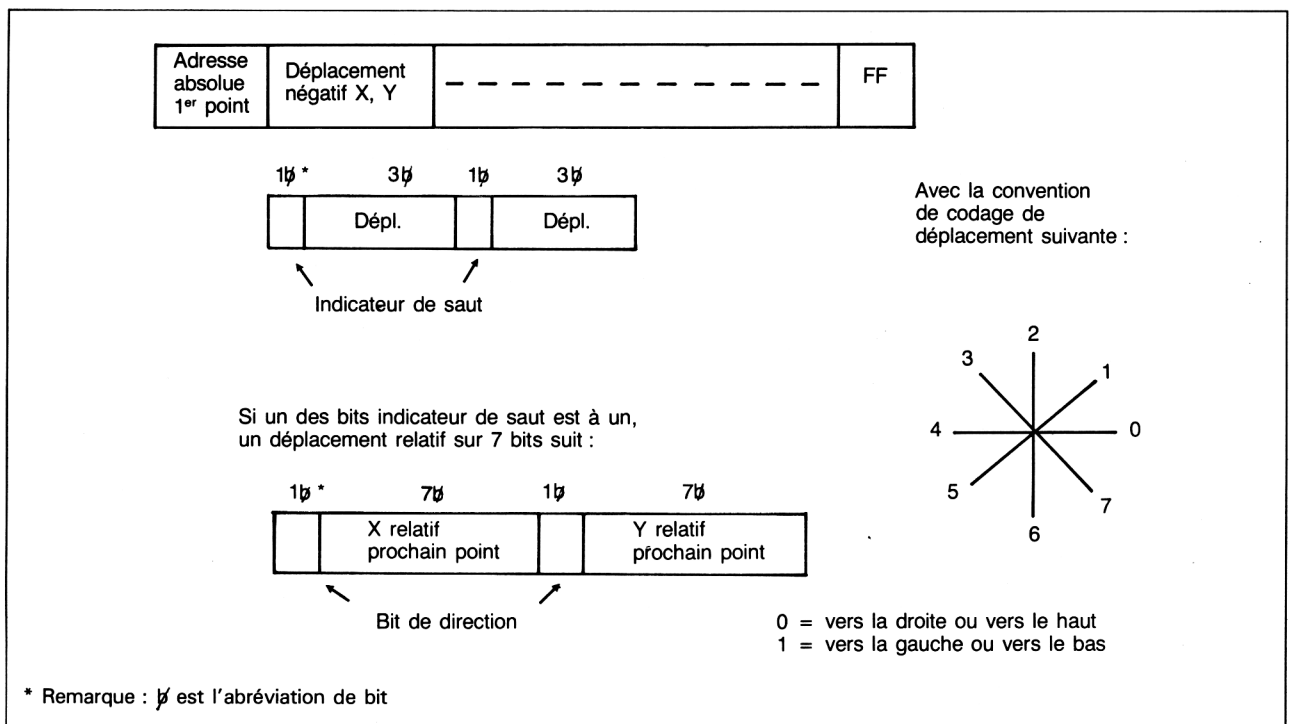


*Remarque importante :*

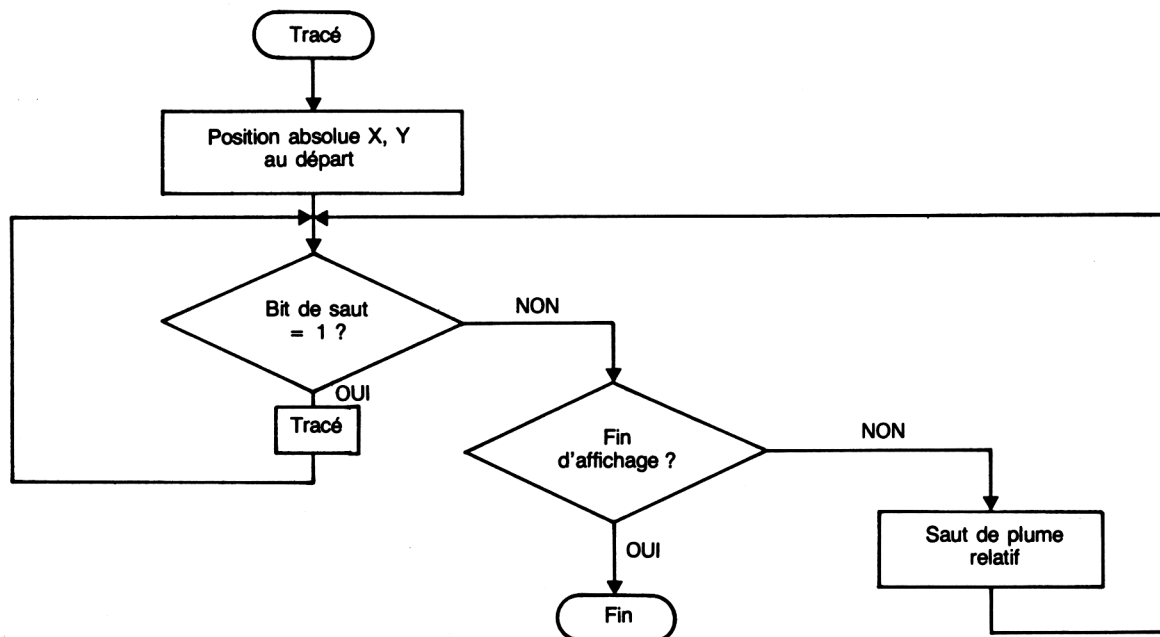
La forme devra OBLIGATOIREMENT être dessinée en PEN 1. Référez-vous aux lignes 1230 à 1300 du listing qui explique le pourquoi de la chose : le calcul du prochain point est effectué en comparant le résultat de la fonction TEST (qui donne la couleur d'un point) à 1.

Le programme défini ci-dessus utilise des sous-programmes écrits en langage d'assemblage pour l'affichage des dessins filiformes.

L'utilisation du langage d'assemblage est quasi obligatoire pour réduire le temps d'affichage qui, malgré tout, n'est pas négligeable (une seconde pour 2 000 points). Le principe est simple. Le fichier compacté généré par le programme précédent possède la structure suivante :



Partant de la structure de ce fichier, voyons comment écrire le programme de tracé.



d'où le programme d'assemblage suivant :

```

1      ;
2      ; Afficheur de formes 4 bits
3      ;
4      ; Entree: HL=à de la forme
5      ; Sortie: Tts registres ecrases
6      ; Pt d'entree: AFD
7      ;
8      ORG 9000H
9      LOAD 9000H
10     ;
11     FORM: DS 2           ; à Forme
12     RT: DS 1
13     TSAUT: EQU $
  
```

```

14 9003 D090          DW  DIR0
15 9005 D890          DW  DIR1
16 9007 E090          DW  DIR2
17 9009 E890          DW  DIR3
18 900B F090          DW  DIR4
19 900D F890          DW  DIR5
20 900F 0091          DW  DIR6
21 9011 0891          DW  DIR7
22                   ;
23                   AFD:    EQU  $                ;Point d'entree
24 9013 220090        LD   (FORM),HL          ;Sauvegarde à forme
25 9016 FD2A0090      LD   IY,(FORM)
26 901A FD5600        LD   D,(IY+0)
27 901D FD5E01        LD   E,(IY+1)
28 9020 FD6602        LD   H,(IY+2)
29 9023 FD6E03        LD   L,(IY+3)
30 9026 CDEABB        CALL OBBEAH          ;Plot absolu
31                   ;
32 9029 DD2A0090      LD   IX,(FORM)
33 902D DF23          INC  IX
34 902F DD23          INC  IX
35 9031 DD23          INC  IX
36 9033 DD23          INC  IX          ;IX=à 1er octet forme
37                   ;
38                   AFD00: EQU  $
39 9035 DD7E00        LD   A,(IX+0)
40                   AFD0:  EQU  $
41 9038 FE88          CP   88H
42 903A 2841          JR   Z,AFD1
43 903C E670          AND  70H
44 903E CB2F          SRA  A
45 9040 CB2F          SRA  A
46 9042 CB2F          SRA  A

```

```
47 9044 CB2F          SRA  A
48 9046 87           ADD  A,A
49 9047 87           ADD  A,A
50 9048 87           ADD  A,A          ;Quartet fort X 8
51 9049 2A0390       LD   HL,(TSAUT)
52 904C 0600         LD   B,0
53 904E 4F           LD   C,A
54 904F 09           ADD  HL,BC
55 9050 3E00         LD   A,0
56 9052 320290       LD   (RT),A        ;Memo retour
57 9055 E9           JP   (HL)          ;Trait quartet fort
58                   AFD2: EQU  $
59 9056 DD7E00       LD   A,(IX+0)
60 9059 CB7F         BIT  7,A
61 905B 2020         JR   NZ,AFD1        ;Saut de plume
62 905D E607         AND  7
63 905F 87           ADD  A,A
64 9060 87           ADD  A,A
65 9061 87           ADD  A,A          ;Quartet faible X 8
66 9062 2A0390       LD   HL,(TSAUT)
67 9065 0600         LD   B,0
68 9067 4F           LD   C,A
69 9068 09           ADD  HL,BC
70 9069 3E01         LD   A,1
71 906B 320290       LD   (RT),A        ;Memo retour
72 906E E9           JP   (HL)          ;Trait quartet faible
73                   ;
74                   AFD3: EQU  $
75 906F DD7E00       LD   A,(IX+0)
76 9072 CB5F         BIT  3,A
77 9074 2007         JR   NZ,AFD1        ;Saut de plume
78 9076 DD23         INC  IX
```



```

79 907B DD7E00      LD  A,(IX+0)
80 907B 18BB        JR  AF00
81                ;
82                AF01: EQU  $
83 907D DD23        INC IX
84 907F DD7E00      LD  A,(IX+0)
85 9082 FEFF        CP  0FFH
86 9084 C8          RET  Z
87 9085 CB7F        BIT  7,A
88 9087 200B        JR  NZ,AF011      ;X negatif
89 9089 6F          LD  L,A
90 908A 2600        LD  H,0
91 908C 37          SCF
92 908D 3F          CCF
93 908E ED6A        ADC HL,HL
94 9090 54          LD  D,H
95 9091 5D          LD  E,L      ;Sauvegarde
96 9092 1812        JR  AF012
97                AF011: EQU  $
98 9094 E67F        AND  7FH
99 9096 6F          LD  L,A
100 9097 2600       LD  H,0
101 9099 37         SCF
102 909A 3F         CCF
103 909B ED6A       ADC HL,HL
104 909D 54         LD  D,H
105 909E 5D         LD  E,L
106 909F 210000     LD  HL,0
107 90A2 ED52       SBC HL,DE
108 90A4 54         LD  D,H
109 90A5 5D         LD  E,L      ;Sauvegarde depl. X rel
110                ;

```

```

111          AFD12:    EQU  $
112 90A6 DD7E01      LD  A,(IX+1)
113 90A9 CB7F        BIT  7,A
114 90AB 2009        JR   NZ,AFD13      ;Y negatif
115 90AD 6F          LD  L,A
116 90AE 2600        LD  H,0
117 90B0 37          SCF
118 90B1 3F          CCF
119 90B2 ED6A        ADC  HL,HL
120 90B4 1810        JR   AFD14
121          AFD13:    EQU  $
122 90B6 E67F        AND  7FH
123 90B8 6F          LD  L,A
124 90B9 2600        LD  H,0
125 90BB 37          SCF
126 90BC 3F          CCF
127 90BD ED6A        ADC  HL,HL
128 90BF 44          LD  B,H
129 90C0 4D          LD  C,L
130 90C1 210000      LD  HL,0
131 90C4 ED42        SBC  HL,BC
132          AFD14:    EQU  $
133 90C6 CDEDBB      CALL OBBEDH      ;Plot relatif
134 90C9 DD23        INC  IX
135 90CB DD23        INC  IX
136 90CD C33590      JP   AFD00      ;Passage a la suite
137          ;
138          DIR0:    EQU  $
139 90D0 110200      LD  DE,2
140 90D3 210000      LD  HL,0
141 90D6 1836        JR   DIR8
142          ;

```

```
143          DIR1:      EQU  $
144 90DB 110200          LD  DE,2
145 90DB 210200          LD  HL,2
146 90DE 182E           JR   DIR8
147          ;
148          DIR2:      EQU  $
149 90E0 110000          LD  DE,0
150 90E3 210200          LD  HL,2
151 90E6 1826           JR   DIR8
152          ;
153          DIR3:      EQU  $
154 90E8 11FEFF          LD  DE,OFFFEH
155 90EB 210200          LD  HL,2
156 90EE 181E           JR   DIR8
157          ;
158          DIR4:      EQU  $
159 90F0 11FEFF          LD  DE,OFFFEH
160 90F3 210000          LD  HL,0
161 90F6 1816           JR   DIR8
162          ;
163          DIR5:      EQU  $
164 90F8 11FEFF          LD  DE,OFFFEH
165 90FB 21FEFF          LD  HL,OFFFEH
166 90FE 180E           JR   DIR8
167          ;
168          DIR6:      EQU  $
169 9100 110000          LD  DE,0
170 9103 21FEFF          LD  HL,OFFFEH
171 9106 1806           JR   DIR8
172          ;
173          DIR7:      EQU  $
174 9108 110200          LD  DE,2
```

```

175 910B 21FEFF          LD  HL,OFFFEH
176                      ;
177                      DIRB: EQU  #
178 910E CDEDBB          CALL OBBEDH
179 9111 3A0290          LD   A,(RT)
180 9114 CB47            BIT  0,A
181 9116 CA5690          JP   Z,AF02
182 9119 C36F90          JP   AF03
183                      END

```

*Remarque :*

Ce programme utilise la routine du FIRMWARE « PLOT RELATIVE ». Reportez-vous en au chapitre 2.7 de la partie 4 pour avoir plus de détails.

## II. Le Décompacteur/Afficheur

Ce programme écrit en BASIC intègre le sous-programme précédent et permet d'afficher simplement une forme à l'écran.

Pour l'utiliser, entrez le nom de la forme à afficher et son implantation en mémoire.

Cette implantation sera toujours &3000 pour une utilisation standard du programme de codage. Pour pouvoir afficher plusieurs formes, il faudra leur donner des adresses d'implantation différentes. Dans ce cas, modifiez le programme de codage lignes 1160, 1690 et 1720 en conséquence. Après avoir chargé la ou les formes, leur dessin apparaît à l'écran.

Le programme d'affichage est le suivant :

```

1000 REM *****
1010 REM Affichage de formes codees
1020 REM *****
1030 '
1040 'Initialisation
1050 '
1060 MEMORY &3000:MODE 1:INK 0,0:INK 1,10:BORDER 0
1070 PRINT"Ce programme permet d'afficher une ou"
1080 PRINT"plusieurs formes a la sortie du"
1090 PRINT"programme 'Compacteur filiforme'"

```

```

1100 '
1110 'Chargement du S/P ASM afficheur de formes
1120 '
1130 FOR I=&9003 TO &911B:READ A:POKE I,A:NEXT I
1140 DATA &D0,&90,&DB,&90,&E0,&90,&E8,&90,&F0,&90,&F8,&90,&0,&91,&8,&91
1150 DATA &22,&0,&90,&FD,&2A,&0,&90,&FD,&56,&0,&FD,&5E,&1,&FD,&66,&2,&FD,&6E,&3,
&CD,&EA,& BB,&DD,&2A,&0,&90,&DD,&23,&DD,&23,&DD,&23,
&DD,&23,&DD,&7E,&0,&FE,&88,&28,&41,&E6,&70,&CB,&2F,&CB,&2F,&CB,&2F,&CB,&2F,&87,&
87,&87,&2A,&3,&90,&6,&0,&4F,&9,&3E,&0,&32
1160 DATA &2,&90,&E9,&DD,&7E,&0,&CB,&7F,&20,&20,&E6,&7,&87,&87,&87,&2A,&3,&90,&6
,&0,&4F,&9,&3E,&1,&32,&2,&90,&E9,&DD,&7E,&0,&CB,&5F,
&20,&7,&DD,&23,&DD,&7E,&0,&18,&BB,&DD,&23,&DD,&7E,&0,&FE,&FF,&CB,&CB,&7F,&20,&B,
&6F,&26,&0,&37,&3F,&ED,&6A,&54,&5D,&18
1170 DATA &12,&E6,&7F,&6F,&26,&0,&37,&3F,&ED,&6A,&54,&5D,&21,&0,&0,&ED,&52,&54,&
5D,&DD,&7E,&1,&CB,&7F,&20,&9,&6F,&26,&0,&37,&3F,&ED,
&6A,&18,&10,&E6,&7F,&6F,&26,&0,&37,&3F,&ED,&6A,&44,&4D,&21,&0,&0,&ED,&42,&CD,&ED
,&BB,&DD,&23,&DD,&23,&C3,&35,&90,&11,&2
1180 DATA &0,&21,&0,&0,&18,&36,&11,&2,&0,&21,&2,&0,&18,&2E,&11,&0,&0,&21,&2,&0,&
18,&26,&11,&FE,&FF,&21,&2,&0,&18,&1E,&11,&FE,&FF,&21
,&0,&0,&18,&16,&11,&FE,&FF,&21,&FE,&FF,&18,&E,&11,&0,&0,&21,&FE,&FF,&18,&6,&11,&
2,&0,&21,&FE,&FF,&CD,&ED,&BB,&3A,&2,&90
1190 DATA &CB,&47,&CA,&56,&90,&C3,&6F,&90
1200 FOR I=0 TO 6:READ A:POKE &9200+I,A:NEXT 'Commande de l'afficheur
1210 DATA &21,0,&30,&CD,&13,&90,&C9
1220 '
1230 'Saisie des formes a afficher
1240 '
1250 NF=NF+1 'Numero de forme
1260 LOCATE 1,10:INPUT"Nom de la forme";N#
1270 INPUT"Implantation memoire";IM(NF)
1280 LOAD N#,IM(NF) 'Chargement forme
1290 INPUT"Position sur l'ecran: X=";X
1300 INPUT"
Y=";Y
1310 A1=INT(X/256):A2=X-A1*256:A3=INT(Y/256):A4=Y-A3*256
1320 POKE IM(NF)+1,A2:POKE IM(NF)+3,A4:POKE IM(NF),A1:POKE IM(NF)+2,A3
1330 PRINT:INPUT"Une autre forme (O/N) ";R#:R#=UPPER#(R#)

```

```
1340 IF R#<>"0" AND R#<>"N" THEN 1330
1350 IF R#="0" THEN 1250
1360 '
1370 'Affichage de la (des) forme(s)
1380 '
1390 CLS:NF=1 'Affichage forme 1
1400 WHILE IM(NF)<>0
1410   MSB=INT(IM(NF)/256):LSB=IM(NF)-MSB*256 'à sur 8 bits
1420   POKE &9201,LSB:POKE &9202,MSB 'Interface afficheur
1430   CALL &9200 'Affichage
1440   NF=NF+1
1450 WEND
1460 END
```

Lignes 1060 à 1090 : Présentation

Lignes 1130 à 1190 : Chargement du sous-programme afficheur

Lignes 1200 à 1210 : Chargement de l'interface  
BASIC/ASSEMBLEUR

Lignes 1250 à 1350 : Saisie des formes à afficher

Lignes 1390 à 1460 : Affichage des formes.

Les sous-programmes écrits en langage d'assemblage utilisés sont :

- le programme d'affichage défini dans le compacteur filiforme,
- un programme d'interfaçage avec le BASIC qui consiste à donner dans HL la première adresse de la forme à afficher. Cette adresse est décomposée en poids fort et poids faible Ligne 1300.