

9/8.6

Récupération d'un fichier effacé par la commande IERA

Vous connaissez sans doute l'instruction **IERA**, "**nomfic.ext**" du Basic Amstrad qui vous permet d'effacer un fichier sur une disquette. Lorsqu'une telle commande est activée, le fichier spécifié n'est pas supprimé physiquement de la disquette: il est seulement marqué « absent » dans le catalogue, et, de ce fait, il n'apparaît plus lorsque vous demandez le répertoire de la disquette.

Partant de cette remarque, il vient tout de suite à l'esprit qu'un fichier supprimé par la commande IERA peut être restitué en effaçant la marque « absent » du répertoire. C'est effectivement le cas si aucune opération d'écriture sur la disquette n'a été faite depuis l'effacement. Effectivement, les commandes d'écriture sur disquettes sont autorisées à écrire sur les fichiers marqués « absent » (sinon, à quoi la commande IERA servirait-elle ?).

Voyons en détails la façon de procéder pour récupérer un fichier effacé.

Définitions

- On appelle *secteur* un certain nombre d'octets consécutifs situés sur la disquette (512 octets pour Amstrad CPC). Un secteur est la quantité minimale d'informations accédée à chaque lecture sur disquette ou disque dur.
- Sur CPC, un secteur est divisé en quatre zones de taille égale appelées *enregistrements* (chaque zone fait donc 128 octets).
- Un *bloc* est un ensemble de secteurs qui occupent une taille de 1 Koctets ou 2 Koctets. Les blocs permettent une gestion plus aisée des fichiers de taille importante. La taille des blocs est fixée sous CP/M.

Dans le répertoire figurent les numéros des blocs occupés par chaque fichier.

Sachant que :

- la taille standard d'un bloc sous CP/M est de 1 Koctets,
- chaque fichier occupe au minimum un bloc,

il est facile de conclure que chaque fichier, même s'il ne contient qu'un caractère, occupera une taille minimale de 1 Koctet sur la disquette.

Structure détaillée du catalogue

Une des fonctions fondamentales d'un système d'exploitation, quel qu'il soit et quelles que soient ses origines, est de faciliter la gestion des fichiers sur les supports de sauvegarde de masse (lecteurs de disquettes ou de disques durs). Pour ce faire, les systèmes d'exploitation font appel à un répertoire (souvent appelé catalogue sur les ordinateurs CPC) qui contient les informations nécessaires pour retrouver rapidement les fichiers sur le support mémoire. Ces informations sont les suivantes :

- nom des fichiers,
- emplacement des fichiers sur le support.

Examinons en détails les données enregistrées dans le répertoire pour chaque fichier. Par la suite, nous appellerons *entrées* l'ensemble des données permettant d'accéder à un fichier dans le répertoire.

Chaque entrée comporte 32 octets dont voici la signification :

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F

Octet 00 : Numéro de USER du fichier (00 à 0F), ou indicateur de fichier « absent » (effacé par une commande IERA). Lorsqu'un fichier a été effacé, cet octet contient la valeur #E5.

Octets 01 à 08 : Nom du fichier, complété par des caractères « Espace » (#20) si nécessaire (si la longueur du nom est inférieure à 8 caractères).

Octets 09 à 0B : Nom de l'extension.

Octet 0C : Numéro de l'extension (Cf. Octet 0F).

Octets 0D à 0E : 0

Octet 0F : Nombre d'enregistrements du fichier.
Si cet octet vaut #80 (128), une « extension » suit car une entrée ne peut référencer que 16 Koctets (effectivement, une entrée ne peut comporter que 16 numéros de blocs, donc 16 Koctets. (Cf. Octets 10 à 1F.)

Octets 10 à 1F : Numéros de blocs occupés par le fichier.

Précisons encore que la position du répertoire dépend du format de la disquette. Une disquette peut être formatée en « Système », « Data » ou « lbm ».

— si la disquette est au format système, le catalogue se trouvera sur la piste 2, secteurs 65 à 69 ;

— si la disquette est au format data, le catalogue se trouvera sur la piste 0, secteurs 193 à 197 ;

— si la disquette est au format lbm, le catalogue se trouvera sur la piste 1, secteurs 1 à 5.

Ainsi l'octet 0 d'une entrée contiendra la valeur #E5 si ce fichier a été effacé, ou un numéro d'USER. Pour qu'un tel fichier soit à nouveau accessible, il suffira de charger l'octet 0 de l'entrée correspondante avec la valeur 0.

Utilisation des instructions « KL FIND COMMAND », « READ SECTOR » et « WRITE SECTOR »

Instruction « READ SECTOR » (&84) du lecteur de disquettes

Une des instructions cachées du lecteur de disquette est « READ SECTOR ».

Cette instruction permet d'accéder à un secteur quelconque de la disquette. Nous allons l'utiliser pour lire la valeur d'une entrée dans le répertoire.

Instruction « WRITE SECTOR » (&85) du lecteur de disquettes

Une autre instruction cachée du lecteur de disquettes est « WRITE SECTOR ».

Nous allons l'utiliser pour réécrire l'entrée lue par READ SECTOR dans le répertoire disquette, en ayant pris le soin de rectifier la valeur contenue dans le premier octet de l'entrée.

Accès aux instructions « READ SECTOR » et « WRITE SECTOR » à travers la macro instruction KL FIND COMMAND

L'instruction « KL FIND COMMAND » du Firmware est située à l'adresse #BCD4. Elle permet de trouver l'adresse d'une instruction de type RSX ou d'une commande externe de la ROM basse de l'Amstrad.

Elle permet, en outre, de trouver les adresses des commandes de lecture et d'écriture « READ SECTOR » et « WRITE SECTOR ». L'instruction « KL FIND COMMAND » combinée à un RESTART 24 permettra d'activer la lecture ou l'écriture d'un secteur de la disquette.

Interfaçage de la routine KL FIND COMMAND

L'adresse en mémoire où se trouve le nom de la commande à accéder doit être fournie en entrée de cette routine dans le registre HL.

En sortie, si la commande indiquée est trouvée :

- l'indicateur « Carry » est positionné à un ;
- le registre C contient l'adresse de ROM SELECT ;
- le registre HL contient l'adresse de la routine.

Si la commande indiquée n'est pas trouvée :

- l'indicateur « Carry » est positionné à zéro ;
- les registres C et HL contiennent des valeurs sans signification.

Dans tous les cas :

- les registres A, B et DE sont effacés ;
- les autres registres sont intacts.

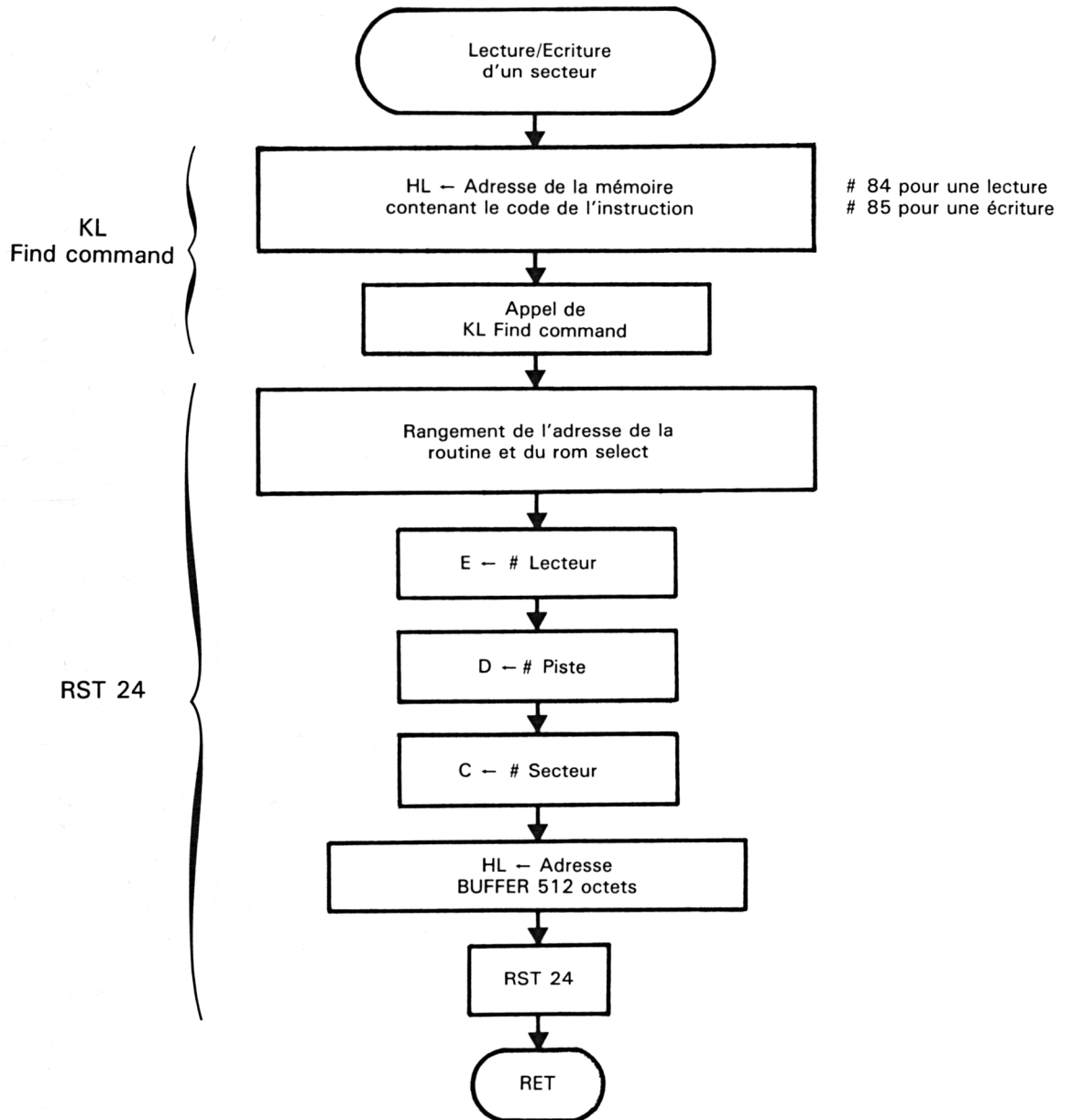
Interfaçage du RESTART 24

Les registres C, D, E et HL sont utilisés en entrée. Ils doivent contenir les informations suivantes :

- E = Numéro du lecteur de disquettes,
- D = Numéro de la piste à accéder,
- C = Numéro du secteur à accéder,
- HL = Adresse d'un buffer de 512 octets nécessaire à la lecture ou à l'écriture d'un secteur.

Nous pouvons maintenant réaliser assez facilement des programmes en assembleur de lecture et d'écriture d'un secteur de la disquette.

Ces deux programmes respecteront la logique suivante :



Algorithme : Lecture / Ecriture d'un secteur sur disquette

Lecture d'un secteur sur disquette

```

1          ORG 8000H
2          LOAD 8000H
3          ;
4          ;-----
5          ; Definition des constantes
6          ;-----
7          ;
8          FIND:      EQU 0BCD4H          ;KL FIND COMMAND
9          ;
10         ;-----
11        ; Definition des variables
12        ;-----
13        ;
14        AD:         DS 3
15        BUFF:       DS 512             ;Buffer lect/ecrit
16        MAX:        DS 1
17        DRIVE:      DS 1              ;Nom du drive
18        PISTE:      DS 1              ;Numero de piste
19        SECT:       DS 1              ;Numero de secteur
20 8207 84          LIT:         DB 84H          ;Instr. lecture
21 8208 85          ECRIT:      DB 85H          ;Instr. ecriture
22        ;
23        ;-----
24        ; Lecture d'un secteur
25        ;-----
26        ;Entree: Lecteur, piste et secteur
27        ;Sortie: Secteur lu dans BUFF
28        ;-----
29        ;
30        LECT:       EQU *              ;Point d'entree

```

```
31 8209 210782      LD   HL,LIT           ;Lecture disque
32 820C CDD4BC      CALL FIND             ;KL FIND COMMAND
33 820F 220080      LD   (AD),HL
34 8212 79          LD   A,C
35 8213 320280      LD   (AD+2),A
36 8216 3A0482      LD   A,(DRIVE)
37 8219 5F          LD   E,A             ;No de drive
38 821A 3A0582      LD   A,(PISTE)
39 821D 57          LD   D,A             ;No de piste
40 821E 3A0682      LD   A,(SECT)
41 8221 4F          LD   C,A             ;No de secteur
42 8222 210380      LD   HL,BUFF
43 8225 DF          RST 24             ;Activ. instruction en RO
44 8226 0080      DW  AD
45 8228 C9          RET
46                END
```

Ecriture d'un secteur sur disquette

```

1          ORG 8000H
2          LOAD 8000H
3          ;
4          ;-----
5          ; Definition des constantes
6          ;-----
7          ;
8          FIND:      EQU 0BCD4H          ;KL FIND COMMAND
9          ;
10         ;-----
11         ; Definition des variables
12         ;-----
13         ;
14         AD:        DS 3
15         BUFF:      DS 512              ;Buffer lect/ecrit
16         MAX:       DS 1
17         DRIVE:     DS 1                ;Nom du drive
18         PISTE:     DS 1                ;Numero de piste
19         SECT:      DS 1                ;Numero de secteur
20 8207 84          LIT:      DB 84H          ;Instr. lecture
21 8208 85          ECRIT:    DB 85H          ;Instr. ecriture
22         ;
23         ;-----
24         ; Ecriture d'un secteur
25         ;-----
26         ;Entree: BUFF = buffer a ecrire
27         ;Sortie: Ecriture BUFF sur disq.
28         ;-----
29         ;

```



```

30          ECR:      EQU  $           ;Point d'entree
31 8209 2108B2      LD  HL,ECRIT       ;Ecriture disque
32 820C CDD4BC      CALL FIND         ;KL FIND COMMAND
33 820F 2200B0      LD  (AD),HL
34 8212 79          LD  A,C
35 8213 3202B0      LD  (AD+2),A
36 8216 3A04B2      LD  A,(DRIVE)
37 8219 5F          LD  E,A           ;No de drive
38 821A 3A05B2      LD  A,(PISTE)
39 821D 57          LD  D,A           ;No de piste
40 821E 3A06B2      LD  A,(SECT)
41 8221 4F          LD  C,A           ;No de secteur
42 8222 2103B0      LD  HL,BUFF
43 8225 DF          RST 24           ;Activ. instruction en RC
44 8226 00B0       DW  AD
45 8228 C9         RET
46          END

```

Ces programmes élémentaires vont être agrémentés d'une interface utilisateur qui permettra d'entrer le nom et l'extension du fichier effacé à retrouver. Ce fichier sera cherché sur la disquette spécifiée, et le premier octet de son entrée dans le répertoire sera modifié si nécessaire. Si le fichier spécifié n'est pas trouvé dans le répertoire, le message 'Fichier non trouvé' sera affiché à l'écran.

Le programme de restitution est présenté dans deux versions : totalement assembleur et basic plus assembleur.

Interface utilisateur

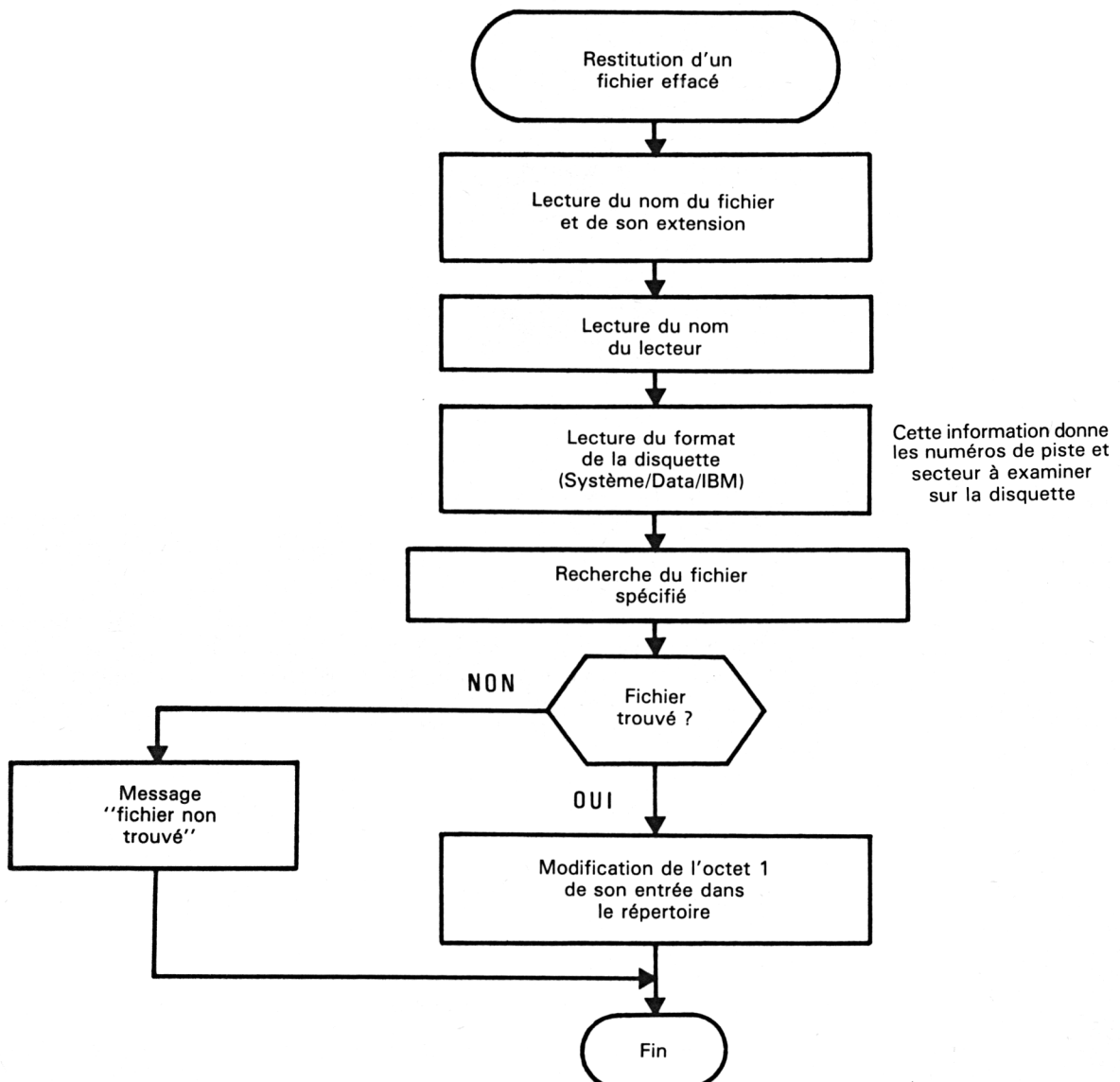
Cette interface consiste à lire :

- le nom et l'extension du fichier à retrouver ;
- le nom du lecteur de disquette (A ou B) ;
- le format de la disquette (Système, donnée ou lbm).

Le fichier est recherché dans la totalité du catalogue. Si le fichier spécifié est rencontré dans le catalogue, le premier octet de l'entrée correspondante est comparé à #E5. S'il est égal à cette valeur, il est mis à 0 pour être accessible sous un numéro d'USER quelconque.

Si le fichier spécifié n'est pas trouvé dans le catalogue ou s'il est trouvé mais avec le premier octet de son entrée différent de #E5, le message 'Fichier non trouvé' est affiché et le répertoire est laissé intact. Ces diverses actions sont résumées dans l'algorithme suivant.

Restitution d'un fichier effacé



Programme de restitution de fichiers en version Basic

- Les ordres « KL FIND COMMAND », « READ SECTOR » et « WRITE SECTOR » ne peuvent pas être utilisés facilement en Basic, c'est pourquoi nous allons développer un petit programme en assembleur pour lire et écrire un secteur quelconque de la disquette.

```

1          ORG  9000H
2          LOAD 9000H
3          ;
4          ;-----
5          ; Definition des constantes
6          ;-----
7          ;
8          FIND:      EQU  0BCD4H          ;KL FIND COMMAND
9          ;
10         ;-----
11         ; Definition des variables
12         ;-----
13         ;
14         AD:        DS    3
15         BUFF:      DS    512           ;Buffer lect/ecrit
16         MAX:       DS    1
17         DRIVE:     DS    1           ;Nom du drive
18         PISTE:     DS    1           ;Numero de piste
19         SECT:      DS    1           ;Numero de secteur
20         ;
21         ORG  9217H
22         LOAD 9217H
23 9217 84          LIT:      DB    84H          ;Instr. lecture
24 9218 85          ECRIT:    DB    85H          ;Instr. ecriture
25         ;
26         ORG  9375H
27         LOAD 9375H

```

```

28      ;
29      ;-----
30      ; Ecriture d'un secteur
31      ;-----
32      ;Entree: BUFF = buffer a ecrire
33      ;Sortie: Ecriture BUFF sur disq.
34      ;-----
35      ;
36      ECR:      EQU  $           ;Point d'entree
37 9375 211892   LD   HL,ECRIT     ;Ecriture disque
38 9378 CDD4BC   CALL FIND       ;KL FIND COMMAND
39 937B 220090   LD   (AD),HL
40 937E 79      LD   A,C
41 937F 320290   LD   (AD+2),A
42 9382 3A0492   LD   A,(DRIVE)
43 9385 5F      LD   E,A         ;No de drive
44 9386 3A0592   LD   A,(PISTE)
45 9389 57      LD   D,A         ;No de piste
46 938A 3A0692   LD   A,(SECT)
47 938D 4F      LD   C,A         ;No de secteur
48 938E 210390   LD   HL,BUFF
49 9391 DF      RST  24         ;Activ. instruction en R0
50 9392 0090   DW   AD
51 9394 C9      RET
52      ;
53      ;-----
54      ; Lecture d'un secteur
55      ;-----
56      ;Entree: PISTE et SECTeur a lire
57      ;Sortie: secteur lu dans BUFF
58      ;-----

```

```
59          ;
60          LECT:      EQU  $          ;Point d'entree
61 9395 211792      LD  HL,LIT          ;Lecture disque
62 9398 CDD4BC      CALL FIND          ;KL FIND COMMAND
63 939B 220090      LD  (AD),HL
64 939E 79          LD  A,C
65 939F 320290      LD  (AD+2),A
66 93A2 3A0492      LD  A,(DRIVE)
67 93A5 5F          LD  E,A          ;No de drive
68 93A6 3A0592      LD  A,(PISTE)
69 93A9 57          LD  D,A          ;No de piste
70 93AA 3A0692      LD  A,(SECT)
71 93AD 4F          LD  C,A          ;No de secteur
72 93AE 210390      LD  HL,BUFF
73 93B1 DF          RST 24          ;Activ. instruction en R0
74 93B2 0090      DW  AD
75 93B4 C9          RET
76          END
```

Ce programme en Assembleur est en effet une concaténation des deux précédents. Il est implanté en #9375 pour être identique au programme de restitution écrit totalement en Assembleur présenté par la suite.

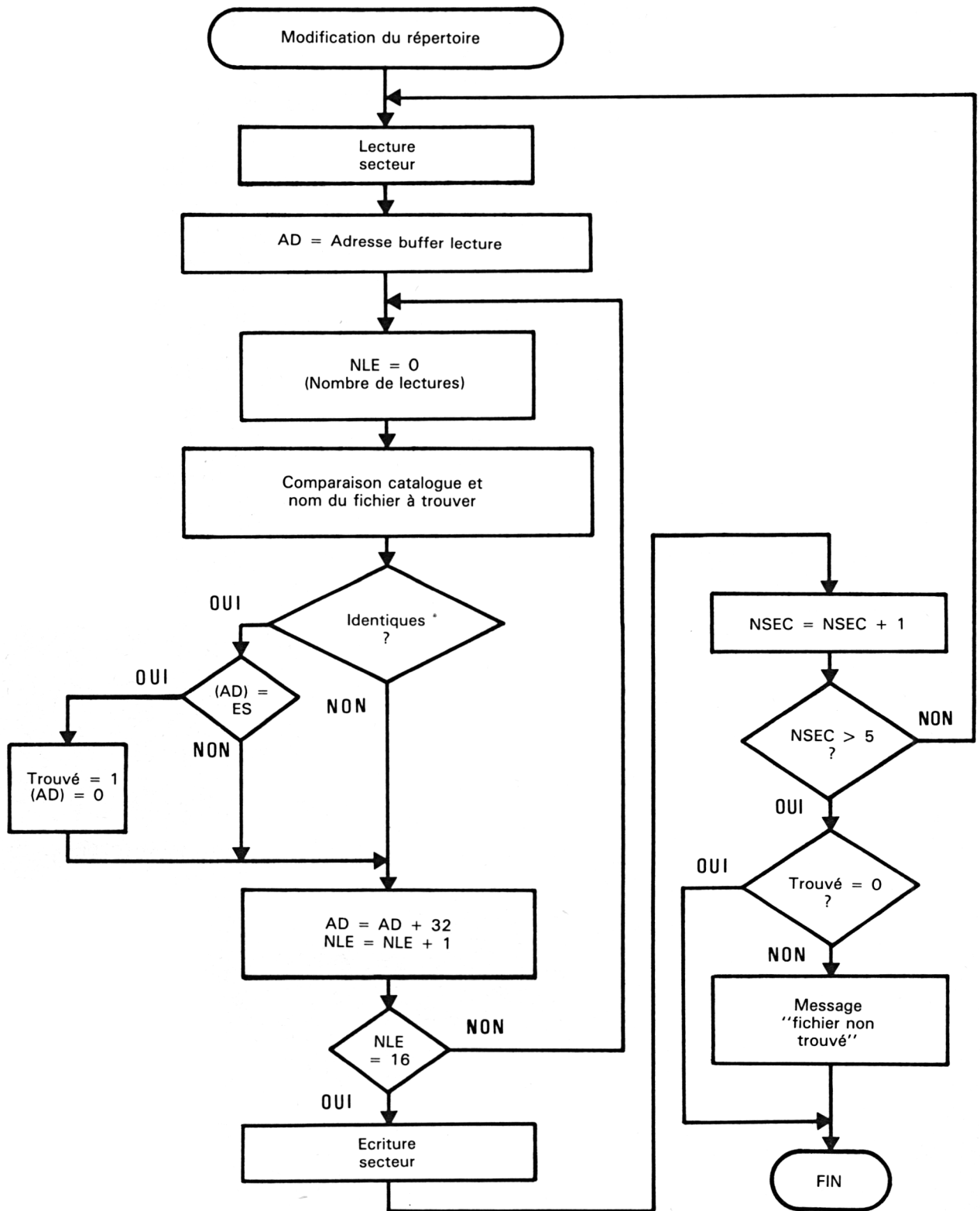
- Le programme Basic de restitution est relativement court, mais assez lent, c'est pourquoi on pourra référer une version assembleur.

```
1000 REM Restitution d'un fichier efface
1010 REM -----
1020 MEMORY &BFFF 'Pour eviter un ecrasement par le Basic
1030 FOR I=&9375 TO &93B4
1040   READ A#
1050   A#="&"+A#
1060   POKE I,VAL(A#)
1070 NEXT I
1080 DATA 21,18,92,CD,D4,BC,22,0,90,79,32,02,90,3A,4,92
1090 DATA 5F,3A,5,92,57,3A,6,92,4F,21,3,90,DF,0,90,C9
1100 DATA 21,17,92,CD,D4,BC,22,0,90,79,32,2,90,3A,4,92
1110 DATA 5F,3A,5,92,57,3A,6,92,4F,21,3,90,DF,0,90,C9
1120 POKE &9217,&84 'Ordre de lecture disque
1130 POKE &9218,&85 'Ordre d'écriture disque
1140 CLS
1150 PRINT "Restitution d'un fichier efface"
1160 PRINT "-----"
1170 PRINT:INPUT"Nom du fichier : ";nf#
1180 PRINT:INPUT"Extension : ";ex#
1190 PRINT:INPUT"Lecteur A ou B : ";le#
1200 le#=UPPER$(le#)
1210 IF le#<>"A" AND le#<>"B" THEN SOUND 1,100,30:GOTO 1190
1220 PRINT:INPUT"Format Systeme Data ou Ibm : ";fo#
1230 fo#=UPPER$(fo#)
1240 IF fo#<>"S" AND fo#<>"D" AND fo#<>"I" THEN SOUND 1,100,30:GOTO 1220
1250
1260 ll=LEN(nf#)
```

```
1270 IF l1<8 THEN FOR i=l1+1 TO 8 : nf#=nf#+ " " : NEXT i
1280 nf#=nf#+ex#
1290 l1=LEN(ex#)
1300 IF l1<3 THEN FOR i=l1+1 TO 3 : nf#=nf#+ " " : NEXT i
1310 '
1320 IF le#="A" THEN POKE &9204,0 ELSE POKE &9204,1
1330 IF fo#="S" THEN POKE &9205,2:POKE &9206,&41
1340 IF fo#="D" THEN POKE &9205,0:POKE &9206,&C1
1350 IF fo#="I" THEN POKE &9205,1:POKE &9206,1
1360 '
1370 nsec=1 'Nombre de secteurs lus
1380 CALL &9395 'Lecture
1390 ad=&9003 'Adresse de lecture nom
1400 nle=0 'Nombre de lectures
1410 lu#=""
1420 FOR i=1 TO l1
1430   ter=PEEK(ad+i)
1440   ter#=CHR$(ter)
1450   lu#=lu#+ter#
1460 NEXT i
1470 IF (lu#=nf#) AND (PEEK(ad)=&E5) THEN trouve=1:POKE ad,0
1480 ad=ad+32 'Passage a l'entree suivante
1490 nle=nle+1 'Nombre de lectures + 1
1500 IF nle<>16 THEN 1410 'Lecture prochaine entree
1510 CALL &9375 'Ecriture
1520 nsec=nsec+1 'Nombre de secteurs lus + 1
1530 IF nsec>5 THEN 1550
1540 POKE &9206,PEEK(&9206)+1:GOTO 1380 'Boucle de lecture
1550 '
1560 IF trouve=0 THEN PRINT:PRINT"Fichier non trouve"
1570 END
```

- Lignes 1000 à 1130 : Implantation du programme Assembleur en mémoire.
- Lignes 1140 à 1250 : Entrée des données pour retrouver le fichier.
- Lignes 1260 à 1310 : Constitution du nom du fichier à retrouver.
- Lignes 1320 à 1350 : Mémorisation des données entrées pour l'interface avec le programme Assembleur.
- Lignes 1370 à 1540 : Lecture, modification éventuelle et écriture des entrées du répertoire.
- Ligne 1560 : Affichage du résultat de la recherche.

Le détail des lignes 1370 à 1540 est donné dans l'algorithme suivant :



Programme de restitution de fichiers en version Assembleur

Le programme Assembleur de restitution reprend les opérations détaillées dans l'algorithme précédent.

A noter que l'affichage du texte est fait par la routine AFALPH, et la saisie formatée de texte par la routine SAISIE. La routine SAISIE demande le nombre maximal de caractères à saisir, et fait un retour chariot automatique lorsque ce nombre est atteint. Une routine d'émission d'un beep sonore (BEEP) est également utilisée. (Voir Partie 6, chap. 5.2).

Les routines de lecture et d'écriture d'un secteur sont implantées aux mêmes adresses que dans le programme Basic.

```

1          ORG 9000H
2          LOAD 9000H
3          ;-----
4          ;Recuperation d'un fichier efface
5          ;-----
6          ;
7          ;-----
8          ;Definition des constantes
9          ;-----
10         ;
11         SOUND:    EQU 0BD34H          ;MC SOUND REG
12         FIND:     EQU 0BCD4H          ;KL FIND COMMAND
13         PRINT:    EQU 0BB5AH          ;TXT OUTPUT
14         READ:     EQU 0BB06H          ; KM WAIT CHAR
15         MODE:     EQU 0BC0EH          ;SCR SET MODE
16         DEL:      EQU 127             ;Caractere DELeTe
17         BS:       EQU 8               ;Caractere Back Space
18         CR:       EQU 13             ;Carriage Return
19         LF:       EQU 10             ;Line Feed
20         CURS:     EQU 95             ;Caractere curseur
21         BLANC:    EQU 32             ;Caractere espace
22         ;

```

```

23      ;-----
24      ;Definition des variables
25      ;-----
26      ;
27      AD:          DS    3          ;Sauv. sortie de FIND
28      BUFF:       DS    512       ;Buffer ecriture
29      MAX:        DS    1
30      DRIVE:     DS    1          ;Nom du drive
31      PISTE:      DS    1          ;Numero de piste
32      SECT:      DS    1          ;Numero de secteur
33      PBUF:      DS    2          ;Pointeur sur un buffer
34      NOMLEC:    DS    1          ;Nom du lecteur
35      FORDIS:    DS    1          ;Format Disquette
36      NOM:       DS    12         ;Nom Fichier+extension
37 9217 84      LIT:      DB    84H   ;Instr. lecture
38 9218 85      ECRIT:   DB    85H   ;Instr. ecriture
39      ALEC:      DS    2          ;@ lecture ds buffer
40      NLE:      DS    1          ;Nbre d'entrees lues
41      TROUVE:   DS    1          ;Fichier trouve
42      NSEC:     DS    1          ;Nbre de secteurs lus
43      ;
44      ;-----
45      ;Definition des messages
46      ;-----
47      ;
48      MES1:     EQU    $
49 921E 52657374      DB    "Restitution d'un
49 9222 69747574
49 9226 696F6E20
49 922A 6427756E
49 922E 20666963

```

```
49 9232 68696572
50 9236 20656666        DB  " efface",CR,LF
50 923A 6163650D
50 923E 0A
51 923F 2D2D2D2D        DB  "-----"
51 9243 2D2D2D2D
51 9247 2D2D2D2D
51 924B 2D2D2D2D
51 924F 2D2D2D2D
51 9253 2D2D2D2D
52 9257 2D2D2D2D        DB  "-----",CR,LF,LF
52 925B 2D2D2D0D
52 925F 0A0AFF
53          ;-----
54          MES2:      EQU  $
55 9262 4E6F6D20        DB  "Nom du fichier :"
55 9266 64752066
55 926A 69636869
55 926E 6572203A
56 9272 5FFF          DB  CURS,OFFH
57          ;-----
58          MES2B:     EQU  $
59 9274 45787465        DB  "Extension :"
59 9278 6E73696F
59 927C 6E203A
60 927F 5FFF          DB  CURS,OFFH
61          ;-----
62          MES3:      EQU  $
63 9281 4C656374        DB  "Lecteur A ou B :"
63 9285 65757220
63 9289 41206F75
```

```
63 928D 2042203A
64 9291 5FFF          DB  CURS,OFFH
65          ;-----
66          MES4:      EQU  $
67 9293 466F726D      DB  "Format Systeme, D
67 9297 61742053
67 929B 79737465
67 929F 6D652C20
67 92A3 44617461
68 92A7 206F7520      DB  " ou Ibm :",CURS,O
68 92AB 49626D20
68 92AF 3A5FFF
69          ;-----
70          MES5:      EQU  $
71 92B2 46696368      DB  "Fichier non trouv
71 92B6 69657220
71 92BA 6E6F6E20
71 92BE 74726F75
71 92C2 7665
72 92C4 FF          DB  OFFH
73          ;-----
74          ALALI:     EQU  $
75 92C5 0D0A0AFF      DB  CR,LF,LF,OFFH
76          ;-----
77          ;
78          ;-----
79          ;Sous-programmes
80          ;-----
81          ;
82          ;-----
83          ;Initialisation
84          ;-----
```

```

85          ;
86          INIT:      EQU  $                ;Point d'entree
87 92C9 210B92      LD   HL,NOM
88 92CC 3E20        LD   A,20H            ;Blanc
89 92CE 060C        LD   B,12
90          IO:       EQU  $
91 92D0 77          LD   (HL),A
92 92D1 23          INC  HL
93 92D2 10FC        DJNZ IO              ;RAZ Buffer Nom
94          ;
95 92D4 AF          XOR  A
96 92D5 321D92      LD   (NSEC),A        ;Nbre secteurs lus
97 92D8 3E01        LD   A,1
98 92DA 321C92      LD   (TROUVE),A      ;Non, trouve
99 92DD 321B92      LD   (NLE),A        ;Nbre entrees lues
100 92E0 C9        RET
101          ;
102          ;-----
103          ;Emission d'un BEEP sonore
104          ;-----
105          ;Entree: aucune
106          ;Sortie: AF, BC et HL effaces
107          ;-----
108          ;
109          BEEP:     EQU  $                ;Point d'entree
110 92E1 3E00        LD   A,0
111 92E3 0E00        LD   C,0
112 92E5 CD34BD      CALL SOUND
113 92E8 3E01        LD   A,1
114 92EA 0E01        LD   C,1
115 92EC CD34BD      CALL SOUND            ;Frequence

```

```

116 92EF 3E08          LD  A,8
117 92F1 0E06          LD  C,6
118 92F3 CD34BD        CALL SOUND             ;Amplitude
119 92F6 3E07          LD  A,7
120 92F8 0E08          LD  C,8
121 92FA CD34BD        CALL SOUND             ;Validation registre A
122                   ;
123 92FD 21FFFF        LD  HL,OFFFH
124                   BOU:  EQU  $
125 9300 2B            DEC  HL
126 9301 7C            LD  A,H
127 9302 B5            OR  L
128 9303 20FB         JR  NZ,BOU
129                   ;
130 9305 3E08          LD  A,8
131 9307 0E00          LD  C,0
132 9309 CD34BD        CALL SOUND             ;Amplitude 0
133 930C C9            RET
134                   ;
135                   ;-----
136                   ;Affichage d'un texte alphanum.
137                   ;-----
138                   ;Entree: @ de depart dans HL
139                   ;Sortie: aucun registre modifie
140                   ;-----
141                   ;
142                   AFALPH: EQU  $             ;point d'entree
143 930D E5            PUSH HL
144 930E F5            PUSH AF
145                   ME1:  EQU  $             ;Boucle d'affichage
146 930F 7E            LD  A,(HL)

```

```

147 9310 FEFF          CP   OFFH
148 9312 2806          JR   Z,ME2          ;Fin d'affichage
149 9314 CD5ABB        CALL PRINT          ;Af. caractere
150 9317 23            INC  HL          ;Caractere suivant
151 9318 18F5          JR   ME1
152                   ME2:   EQU  $
153 931A F1            POP  AF
154 931B E1            POP  HL
155 931C C9            RET
156                   ;
157                   ;-----
158                   ;Saisie de caracteres
159                   ;-----
160                   ;Entree: (MAX)=Nbre de caracteres
161                   ;Sortie: Aucun registre ecrase
162                   ;-----
163                   ;
164                   SAISIE: EQU  $          ;Point d'entree
165 931D 3A0392         LD   A,(MAX)
166 9320 57             LD   D,A          ;Nbre de caract. a lire
167 9321 010000        LD   BC,0          ;Index dans le buffer
168                   S1:   EQU  $
169 9324 2A0792         LD   HL,(PBUF)      ;Buffer de lecture
170 9327 CD06BB        CALL READ          ;Lecture 1 caractere
171 932A FE0D          CP   CR          ;Carriage Return ?
172 932C 283C          JR   Z,S3          ;Oui=> Fin de saisie
173 932E FE7F          CP   DEL          ;DElete ?
174 9330 2818          JR   Z,S2          ;Oui => Traitement DEL
175 9332 F5            PUSH AF
176 9333 3E08          LD   A,BS
177 9335 CD5ABB        CALL PRINT

```



```
178 9338 F1          POP  AF
179 9339 CD5ABB      CALL PRINT
180 933C 09         ADD  HL,BC
181 933D 77         LD   (HL),A          ;Sauvegarde
182 933E 03         INC  BC
183 933F 3E5F       LD   A,CURS
184 9341 CD5ABB      CALL PRINT
185 9344 79         LD   A,C
186 9345 BA         CP   D
187 9346 20DC       JR   NZ,S1          ;Suite de la saisie
188 9348 1820       JR   S3             ;Fin de saisie
189                S2: EQU  #
190 934A 79         LD   A,C
191 934B B7         OR   A
192 934C 28D6       JR   Z,S1           ;DEL non accepte
193 934E 0B         DEC  BC
194 934F 3E08       LD   A,BS
195 9351 CD5ABB      CALL PRINT          ;Retour en arriere
196 9354 3E20       LD   A,BLANC
197 9356 CD5ABB      CALL PRINT          ;Effacement caractere
198 9359 3E08       LD   A,BS
199 935B CD5ABB      CALL PRINT          ;Retour en arriere
200 935E 3E08       LD   A,BS
201 9360 CD5ABB      CALL PRINT          ;Retour en arriere
202 9363 3E5F       LD   A,CURS
203 9365 CD5ABB      CALL PRINT          ;Affichage curseur
204 9368 18BA       JR   S1
205                S3: EQU  #
206 936A 3E08       LD   A,BS
207 936C CD5ABB      CALL PRINT          ;Retour en arriere
208 936F 3E20       LD   A,BLANC
```

```

209 9371 CD5ABB          CALL PRINT          ;Effacement caract.
210 9374 C9              RET
211                      ;
212                      ;-----
213                      ; Ecriture d'un secteur
214                      ;-----
215                      ;Entree:
216                      ;Sortie:
217                      ;-----
218                      ;
219 ECR:                  EQU  $          ;Point d'entree
220 9375 211892          LD   HL,ECRIT          ;Ecriture disque
221 9378 CDD4BC          CALL FIND          ;KL FIND COMMAND
222 937B 220090          LD   (AD),HL
223 937E 79              LD   A,C
224 937F 320290          LD   (AD+2),A
225 9382 3A0492          LD   A,(DRIVE)
226 9385 5F              LD   E,A          ;No de drive
227 9386 3A0592          LD   A,(PISTE)
228 9389 57              LD   D,A          ;No de piste
229 938A 3A0692          LD   A,(SECT)
230 938D 4F              LD   C,A          ;No de secteur
231 938E 210390          LD   HL,BUFF
232 9391 DF              RST  24          ;Activ. instruction en R0
233 9392 0090           DW   AD
234 9394 C9              RET
235                      ;
236                      ;-----
237                      ; Lecture d'un secteur
238                      ;-----
239                      ;Entree:

```

```

240          ;Sortie:
241          ;-----
242          ;
243          LECT:      EQU  $          ;Point d'entree
244 9395 211792      LD   HL,LIT      ;Lecture disque
245 939B CDD4BC      CALL FIND      ;KL FIND COMMAND
246 939B 220090      LD   (AD),HL
247 939E 79          LD   A,C
248 939F 320290      LD   (AD+2),A
249 93A2 3A0492      LD   A,(DRIVE)
250 93A5 5F          LD   E,A          ;No de drive
251 93A6 3A0592      LD   A,(PISTE)
252 93A9 57          LD   D,A          ;No de piste
253 93AA 3A0692      LD   A,(SECT)
254 93AD 4F          LD   C,A          ;No de secteur
255 93AE 210390      LD   HL,BUFF
256 93B1 DF          RST  24          ;Activ. instruction en R0
257 93B2 0090      DW   AD
258 93B4 C9          RET
259          ;
260          ;=====
261          ; PROGRAMME PRINCIPAL
262          ;=====
263          ORG  8000H
264          LOAD 8000H
265 8000 CDC992      CALL INIT      ;des variables
266 8003 3E01      LD   A,1
267 8005 CD0EBC      CALL MODE      ;Mode 1
268 8008 211E92      LD   HL,MES1
269 800B CD0D93      CALL AFALPH      ;Affich 1er message
270 800E 216292      LD   HL,MES2

```

Partie 9 : Programmes

```

271 8011 CD0D93      CALL AFALPH          ;Aff 2eme message
272 8014 3E08        LD    A,B
273 8016 320392      LD    (MAX),A       ;Saisie sur 8 car. max
274 8019 210B92      LD    HL,NOM
275 801C 220792      LD    (PBUF),HL     ;Buffer de lecture
276 801F CD1D93      CALL SAISIE         ;Saisie du nom
277 8022 21C592      LD    HL,ALALI
278 8025 CD0D93      CALL AFALPH         ;A la ligne
279 8028 217492      LD    HL,MES2B
280 802B CD0D93      CALL AFALPH         ;Aff. message 2 bis
281 802E 3E03        LD    A,3
282 8030 320392      LD    (MAX),A       ;3 caract. maxi
283 8033 211392      LD    HL,NOM+8
284 8036 220792      LD    (PBUF),HL
285 8039 CD1D93      CALL SAISIE         ;Saisie extension
286                PPO: EQU  #
287 803C 21C592      LD    HL,ALALI
288 803F CD0D93      CALL AFALPH         ;A la ligne
289 8042 218192      LD    HL,MES3
290 8045 CD0D93      CALL AFALPH         ;Aff 3eme message
291 8048 3E01        LD    A,1
292 804A 320392      LD    (MAX),A       ;Saisie 1 caractere
293 804D 210992      LD    HL,NOMLEC
294 8050 220792      LD    (PBUF),HL
295 8053 CD1D93      CALL SAISIE         ;Lecture nom lecteur
296 8056 3A0992      LD    A,(NOMLEC)
297 8059 E6DF        AND    0DFH         ;Mise en majuscule
298 805B FE41        CP    'A'
299 805D 2809        JR    Z,PP1
300 805F FE42        CP    'B'
301 8061 2805        JR    Z,PP1         ;OK

```

```

302 8063 CDE192      CALL BEEP           ;Emission d'un son
303 8066 18D4        JR    PPO           ;Saisie a nouveau
304                  PP1:   EQU    #
305 8068 21C592      LD    HL,ALALI
306 806B CD0D93      CALL AFALPH        ;A la ligne
307 806E 219392      LD    HL,MES4
308 8071 CD0D93      CALL AFALPH        ;Aff 4eme message
309 8074 3E01        LD    A,1
310 8076 320392      LD    (MAX),A      ;Saisie 1 caractere
311 8079 210A92      LD    HL,FORDIS
312 807C 220792      LD    (PBUF),HL
313 807F CD1D93      CALL SAISIE        ;Lecture format disk
314 8082 3A0A92      LD    A,(FORDIS)
315 8085 E6DF        AND    ODFH        ;Mise en majuscule
316 8087 FE53        CP    'S'
317 8089 280D        JR    Z,PP2
318 808B FE44        CP    'D'
319 808D 2809        JR    Z,PP2
320 808F FE49        CP    'I'
321 8091 2805        JR    Z,PP2        ;OK
322 8093 CDE192      CALL BEEP           ;Emission d'un son
323 8096 18D0        JR    PP1           ;Saisie a nouveau
324                  PP2:   EQU    #
325                  ;
326 8098 3A0992      LD    A,(NOMLEC)
327 809B E6DF        AND    ODFH        ;Mise en majuscule
328 809D FE41        CP    'A'
329 809F 2003        JR    NZ,PP3
330 80A1 AF          XOR    A
331 80A2 1802        JR    PP4
332 80A4 3E01        PP3:   LD    A,1

```

```

333 80A6 320492  PP4:      LD   (DRIVE),A      ;Nom du drive
334                ;
335 80A9 3A0A92      LD   A,(FORDIS)
336 80AC E6DF        AND   ODFH          ;Mise en majuscule
337 80AE FE53        CP   'S'
338 80B0 280E        JR   Z,PP5
339 80B2 FE44        CP   'D'
340 80B4 2816        JR   Z,PP6
341 80B6 3E01        LD   A,1
342 80B8 320592      LD   (PISTE),A
343 80BB 320692      LD   (SECT),A
344 80BE 1815        JR   PP6B
345 80C0 3E02        PP5:      LD   A,2
346 80C2 320592      LD   (PISTE),A
347 80C5 3E41        LD   A,41H
348 80C7 320692      LD   (SECT),A
349 80CA 1809        JR   PP6B
350 80CC AF          PP6:      XOR   A
351 80CD 320592      LD   (PISTE),A
352 80D0 3EC1        LD   A,0C1H
353 80D2 320692      LD   (SECT),A
354                PP6B:      EQU   #
355 80D5 3E01        LD   A,1
356 80D7 321D92      LD   (NSEC),A      ;RAZ nbre sect lus
357                PP7:      EQU   #
358 80DA CD9593      CALL LECT          ;Lecture secteur
359 80DD 210390      LD   HL,BUFF      ;@ buff lect
360 80E0 221992      LD   (ALEC),HL    ;@ de lecture
361 80E3 3E01        LD   A,1
362 80E5 321B92      LD   (NLE),A
363                ;

```

```

364          PP8:      EQU  $           ; Comparaison
365 80EB 2A1992      LD  HL, (ALEC)
366 80EB 23          INC  HL
367 80EC 110B92      LD  DE, NOM
368 80EF 060B        LD  B, 11           ; 11 caract a comparer
369          PP8B:     EQU  $
370 80F1 1A          LD  A, (DE)
371 80F2 BE          CP   (HL)
372 80F3 2015        JR   NZ, PP10
373 80F5 23          INC  HL
374 80F6 13          INC  DE
375 80F7 05          DEC  B
376 80F8 20F7        JR   NZ, PP8B       ; Boucle compar.
377          ; Comparaison reussie
378 80FA 2A1992      LD  HL, (ALEC)
379 80FD 7E          LD  A, (HL)
380 80FE FEE5        CP   OESH
381 8100 200B        JR   NZ, PP10
382 8102 AF          XOR  A
383 8103 321C92      LD  (TROUVE), A       ; Fichier trouve
384 8106 2A1992      LD  HL, (ALEC)
385 8109 77          LD  (HL), A         ; Restitution
386          ;
387          PP10:     EQU  $           ; Comparaison NOK
388 810A 2A1992      LD  HL, (ALEC)
389 810D 112000      LD  DE, 32
390 8110 19          ADD  HL, DE
391 8111 221992      LD  (ALEC), HL
392 8114 3A1B92      LD  A, (NLE)        ; Nbre entrees
393 8117 3C          INC  A
394 8118 321B92      LD  (NLE), A       ; Nbre entrees + 1

```

```
395 811B FE11          CP    17
396 811D 20C9          JR    NZ,PP8
397 811F CD7593        CALL  ECR              ;Ecriture secteur
398 8122 3A0692        LD    A,(SECT)
399 8125 3C            INC    A
400 8126 320692        LD    (SECT),A        ;Prochain secteur
401 8129 3A1D92        LD    A,(NSEC)
402 812C 3C            INC    A
403 812D 321D92        LD    (NSEC),A        ;Nbre secteurs+1
404 8130 FE05          CP    5
405 8132 20A6          JR    NZ,PP7
406 8134 3A1C92        LD    A,(TROUVE)
407 8137 B7            OR    A
408 8138 280C          JR    Z,PP11          ;Fichier pas trouve
409 813A 21C592        LD    HL,ALALI
410 813D CD0D93        CALL  AFALPH          ;A la ligne
411 8140 21B292        LD    HL,MES5
412 8143 CD0D93        CALL  AFALPH          ;Fichier non trouve
413                    PP11: EQU    $
414 8146 C9            RET
415                    END
```


Chargeur Basic du programme Assembleur de restitution

Si vous ne voulez pas entrer le listing assembleur précédent, tout en conservant la vitesse d'exécution de l'assembleur, vous pourrez entrer le chargeur que nous présentons ici.

```

1000 REM Recuperation d'un fichier efface par !ERA
1010 REM =====
1020 REM Mise en memoire du programme assembleur
1030 REM -----
1040 FOR I=&920E TO &93BD
1050   READ A#
1060   A#="&"+A#
1070   POKE I,VAL(A#)
1080 NEXT I
1090 FOR I=&8000 TO &814F
1100   READ a#
1110   A#="&"+A#
1120   POKE I,VAL(A#)
1130 NEXT I
1140 REM -----
1150 REM Appel au programme en &8000
1160 REM -----
1170 CALL &8000
1180 END
2000 REM =====
2010 REM Variables et sous-programmes
2020 REM =====
2030 DATA 0,0,0,0,0,0,0,0,0,0,84,85,0,0,0,0,0
2040 DATA 52,65,73,74,69,74,75,74,69,6F,6E,20,64,27,75,6E
2050 DATA 20,66,69,63,68,69,65,72,20,65,66,66,61,63,65,D
2060 DATA A,2D,2D,2D,2D,2D,2D,2D,2D,2D,2D,2D,2D,2D,2D
2070 DATA 2D,2D,2D,2D,2D,2D,2D,2D,2D,2D,2D,2D,2D,2D,2D

```

Partie 9 : Programmes

```
2080 DATA D,A,A,FF,4E,6F,6D,20,64,75,20,66,69,63,68,69
2090 DATA 65,72,20,3A,5F,FF,45,78,74,65,6E,73,69,6F,6E,20
2100 DATA 3A,5F,FF,4C,65,63,74,65,75,72,20,41,20,6F,75,20
2110 DATA 42,20,3A,5F,FF,46,6F,72,6D,61,74,20,53,79,73,74
2120 DATA 65,6D,65,2C,20,44,61,74,61,20,6F,75,20,49,62,6D
2130 DATA 20,3A,5F,FF,46,69,63,68,69,65,72,20,6E,6F,6E,20
2140 DATA 74,72,6F,75,76,65,FF,D,A,A,FF,21,B,92,3E,20
2150 DATA 6,C,77,23,10,FC,AF,32,1D,92,3E,1,32,1C,92,32
2160 DATA 1B,92,C9,3E,0,E,0,CD,34,BD,3E,1,E,1,CD,34
2170 DATA BD,3E,8,E,6,CD,34,BD,3E,7,E,8,CD,34,BD,21
2180 DATA FF,FF,2B,7C,B5,20,FB,3E,8,E,0,CD,34,BD,C9,E5
2190 DATA F5,7E,FE,FF,28,6,CD,5A,BB,23,18,F5,F1,E1,C9,3A
2200 DATA 3,92,57,1,0,0,2A,7,92,CD,6,BB,FE,D,28,3C
2210 DATA FE,7F,28,18,F5,3E,8,CD,5A,BB,F1,CD,5A,BB,9,77
2220 DATA 3,3E,5F,CD,5A,BB,79,BA,20,DC,18,20,79,B7,28,D6
2230 DATA B,3E,8,CD,5A,BB,3E,20,CD,5A,BB,3E,8,CD,5A,BB
2240 DATA 3E,8,CD,5A,BB,3E,5F,CD,5A,BB,18,BA,3E,8,CD,5A
2250 DATA BB,3E,20,CD,5A,BB,C9,21,18,92,CD,D4,BC,22,0,90
2260 DATA 79,32,2,90,3A,4,92,5F,3A,5,92,57,3A,6,92,4F
2270 DATA 21,3,90,DF,0,90,C9,21,17,92,CD,D4,BC,22,0,90
2280 DATA 79,32,2,90,3A,4,92,5F,3A,5,92,57,3A,6,92,4F
2290 DATA 21,3,90,DF,0,90,C9,0,0,0,0,0,0,0,0,0,0
3000 REM =====
3010 REM Programme principal
3020 REM =====
3030 DATA CD,C9,92,3E,1,CD,E,BC,21,1E,92,CD,D,93,21,62
3040 DATA 92,CD,D,93,3E,8,32,3,92,21,B,92,22,7,92,CD
3050 DATA 1D,93,21,C5,92,CD,D,93,21,74,92,CD,D,93,3E,3
3060 DATA 32,3,92,21,13,92,22,7,92,CD,1D,93,21,C5,92,CD
3070 DATA D,93,21,81,92,CD,D,93,3E,1,32,3,92,21,9,92
3080 DATA 22,7,92,CD,1D,93,3A,9,92,E6,DF,FE,41,28,9,FE
```

```
3090 DATA 42,28,5,CD,E1,92,18,D4,21,C5,92,CD,D,93,21,93
3100 DATA 92,CD,D,93,3E,1,32,3,92,21,A,92,22,7,92,CD
3110 DATA 1D,93,3A,A,92,E6,DF,FE,53,28,D,FE,44,28,9,FE
3120 DATA 49,28,5,CD,E1,92,18,D0,3A,9,92,E6,DF,FE,41,20
3130 DATA 3,AF,1B,2,3E,1,32,4,92,3A,A,92,E6,DF,FE,53
3140 DATA 28,E,FE,44,28,16,3E,1,32,5,92,32,6,92,18,15
3150 DATA 3E,2,32,5,92,3E,41,32,6,92,18,9,AF,32,5,92
3160 DATA 3E,C1,32,6,92,3E,1,32,1D,92,CD,95,93,21,3,90
3170 DATA 22,19,92,3E,1,32,1B,92,2A,19,92,23,11,B,92,6
3180 DATA B,1A,BE,20,15,23,13,5,20,F7,2A,19,92,7E,FE,E5
3190 DATA 20,8,AF,32,1C,92,2A,19,92,77,2A,19,92,11,20,0
3200 DATA 19,22,19,92,3A,1B,92,3C,32,1B,92,FE,11,20,C9,CD
3210 DATA 75,93,3A,6,92,3C,32,6,92,3A,1D,92,3C,32,1D,92
3220 DATA FE,5,20,A6,3A,1C,92,B7,28,C,21,C5,92,CD,D,93
3230 DATA 21,B2,92,CD,D,93,C9,0,0,0,0,0,0,0,0,0,0
```

Pour être sûr que le programme fonctionnera au premier essai, nous vous donnons la liste des codes de vérification à utiliser avec le programme de checksum (Voir Partie 9, chap. 8.4).

```
0A 3E 86 AF D2 6B 72 F6 3C 3E 03 E5 9D D3 14 3D 8E B1 35 1E A1 EC
A5 B9 CB B9 EE C5 57 70 10 08 47 3B 4F 49 9E C4 B8 EE 97 9A A5 0D
B2 EA 87 9E
```

