

LISTINGS ASSEMBLEUR BOOT

```
ORG 9F61H
LOAD 9F61H
```

```
;*****
; 2 routines pour assister le BASIC
```

```
;=====
;remplissage d'ecran avec ASCII 207
```

```
;efface l'ecran
```

```
LD A,0CH
CALL 0BB5AH
```

```
;injecte les parametres
```

```
LD HL,0C000H
LD DE,0C001H
LD C,L
LD A,55H
```

```
FILL: LD B,08
LD (HL),A
LDIR
```

```
;decale motif et continue
```

```
RLCA
LD B,A
LD A,H
OR A
LD A,B
JR NZ,FILL
RET
```

```
;=====
;remplit fenetre de TXT avec ASCII 207
;sans effacer le texte
```

```
TAMPON:DEFS 08
```

```
;reloge X1,X2,Y1,Y2 dans un buffer
PUSH IX
POP HL
```

```
LD DE,TAMPON
LD BC,0008
LDIR
```

```
;corrige valeurs de fin pour tests
;et les insere dans le programme
```

```
LD HL,(TAMPON)
RES 0,L
INC HL
INC HL
LD (Y2+1),HL
LD HL,(TAMPON+06)
RES 0,L
LD (X1+1),HL
LD HL,(TAMPON+04)
RES 0,L
INC HL
INC HL
LD (X2+1),HL
```

```
;met encre de premier plan
```

```
LD A,01
CALL 0BBDEH
```

```
;initialise Y (pair) a Y1
```

```
LD HL,(TAMPON+02)
RES 0,L
DEC HL
DEC HL
```

```
;efface le carry pour la 1ere boucle
```

```
XOR A
```

```
;init. pour boucle sur les Y
```

```
LOOPY: INC HL
INC HL
Y2: LD BC,0000
LD D,H
LD E,L
SBC HL,BC
```

```
;fini si Y depasse Y2
```

```
RET NC
LD H,D
LD L,E
```

```
;decale X1 de 1 si no de ligne/2 impair
```

```
X1: LD DE,0000
BIT 1,L
JR Z,LOOPY
INC DE
```

```
;commence le test sur le point a gauche
```

```
LOOPX: PUSH DE
DEC DE
```

```
;test du pt et des 2 pts adjacents
```

```
CALL TESTPT
JR NZ,NEXTPT
INC DE
CALL TESTPT
JR NZ,NEXTPT
INC DE
```

```

CALL TESTPT
JR NZ, NEXTPT

;3 pts vides: on allume celui du milieu
PUSH HL
DEC DE
CALL #BBEAH
POP HL

;on passe au 2e point a droite
NEXTPT:POP DE
INC DE
INC DE

;preserve nile abscisse pour test
PUSH DE
EX DE, HL
X2: LD BC, #000
XOR A
SBC HL, BC
;restaure X et Y
EX DE, HL
POP DE

;si Carry on continue sur la ligne
JR C, LOOPX
;sinon on passe au dessus
JR LOOPY

;-----
;teste si valeur de point=#

TESTPT:PUSH HL
PUSH DE
CALL #BBF#H
POP DE
POP HL
;teste si point eteint
CP #0
RET

;*****
;*
;* TRANSFIL V1.0 - transfert *
;* entre tous formats *
;* copyright Y. GOUR (c)1991 *
;*
;*****

;adresses de donnees

TYPE: EQU #007H
RANDOS: EQU #AC00H
PTRVAR: EQU #AD64H
DESCRI: EQU #AD66H
BUFFER: EQU #AD69H
LOC: EQU #AD75H
LEN: EQU #AD77H

```

```

EXEC: EQU #AD79H
HEXA: EQU #AD7BH
RDSEC: EQU #BDF1H
WRSEC: EQU #BDF4H
SEEK: EQU #BDF7H
DIR: EQU #BDFAH
ERA: EQU #BDFDH
DPB1: EQU #BE00H
DPB2: EQU #BE19H
BANSUP: EQU #BE32H
TIMENU: EQU #BE60H

;adresses de routines

MENU: EQU #122H
TOUCHE: EQU #12CH
TRANSF: EQU #1C3H

;valeurs et touches

LOMEM: EQU #300H
HIMEM: EQU #ABFBH
LENPK1: EQU #095H
LENPK2: EQU #0B1H

ENTER: EQU #DH
ESC: EQU #FCH
DEL: EQU #7FH

;*****
; INSTALLATION DU PROGRAMME & CO

DI
;reinitialise vecteurs
CALL #BD37H

;reloge packs de routines
LD HL, PACK1
LD DE, SWAP#3
LD BC, LENPK1
LDIR
LD HL, PACK2
LD DE, INPUT
LD BC, LENPK2
LDIR

;reloge programme
LD HL, PGH
LD DE, #040H
LD BC, #300H
LDIR

;installe routine d'affichage en RST#0
LD HL, AFF
LD DE, #000
LD BC, #007
LDIR

;TXT OUTPUT en RST30
LD HL, #BB5AH

```

```

LD DE, #030H
LD BC, #003H
LDIR

;copyright dans bandeau
LD HL, COPYR
RST #0

;installe DOS
LD HL, #B0FFH
LD DE, #040H
LD C, #7
CALL #BCCEH

;trouve RSX 4,5,7, DIR et ERA
LD HL, RSXDOS
LD DE, RDSEC
LD B, #5
FNDRSX:PUSH HL
PUSH BC
PUSH DE
;cherche RSX
CALL #BCD4H
;stop si introuvable
JR NC, STOP
;recupere adr. buffer
POP DE
;echange avec adr. RSX
EX DE, HL
;stocke adr. RSX dans buffer
LD (HL), E
INC HL
LD (HL), D
INC HL
LD (HL), C
INC HL
;remet a jour adr. buffer
EX DE, HL
;restaure compteur et ptr RSX
POP BC
POP HL
;nile RSX suit un zero (cf table)
INCHL: LD A, (HL)
INC HL
CP #0
JR NZ, INCHL
DJNZ FNDRSX

;ptr et descripteur de VAR pour ERA
LD HL, DESCR1
LD (PTRVAR), HL
LD HL, BUFFER
LD (DESCRI+1), HL

;loge chiffres en HEXA
LD HL, NBHEXA
LD DE, HEXA
LD BC, #010H
LDIR

```

```

LD SP,0C000H
EI
JP MENU

STOP: CALL SWAP03
LD HL, TXSTOP
STOP2: RST 00
CALL WETKEY
LD HL, 0000H
LD C, 0FDH
CALL 0BD16H

;routine d'affichage de texte
AFF: LD A, (HL)
RST 30H
INC HL
OR A
JR NZ, AFF
RET

;message de copyright
COPYR: DEFB 1FH, 01, 01, ' TRANSFIL '
DEFB 'V1.0 (C) Y. GOUR'
DEFB ' 1991.', 00

;RSXs du DOS
RSXDOS: DEFB 04+80H, 00
DEFB 05+80H, 00
DEFB 07+80H, 00
DEFB 'DI', 'R'+80H, 00
DEFB 'ER', 'A'+80H, 00

;-----
;chiffres HEXA pour les Users
NBHEXA: DEFB '0123456789ABCDEF'

;pas de ROM DISC!!!
TXSTOP: DEFB 0AH, '
DEFB 'PAS DE ROM DISC!'
DEFB ' -RESET NECESSAIRE-'
DEFB ' FRAPPER UNE TOUCHE'
DEFB 00

PACK1: DEFS LENPK1
PACK2: DEFS LENPK2
PGM: DEFB 00

;-----
; PACK de routines NO1

ORG 0BD38H
LOAD PACK1

;-----
;echange fenetres 0 et 3
SWAP03: LD B, 03

```

```

;echange fenetres 0 et B
SWAPWI: LD C, 00
JP 0BBB7H

;-----
;attend touche
WETKEY: PUSH BC
LD B, 64H
ATTEND: CALL 0BB09H
DJNZ ATTEND
POP BC
JP 0BB06H

;-----
;affiche nouveau Lecteur/User
AFFLU: XOR A
LD (LECUSR+4), A
;AFFLU2
LD HL, LECUSR
RST 00
RET

;-----
;lecture du DRIVE
GETLEC: LD HL, BANSUP
JR GETLU

;-----
;lecture du user
GETUSR: LD HL, BANSUP+4

;-----
;sub pour lire source ou dest.

;preserve source/dest. dans C
GETLU: LD C, A
;lit valeur '1'
LD B, (HL)
;determine sens
LD A, (BANSUP+8)
AND 00H
RLCA
CP C
LD A, B
;retour pour valeur '1'
RET Z
;lit valeur '2'
INC HL
INC HL
LD A, (HL)
RET

;-----
;installe Lecteur et User

;preserve Source/Dest. dans D
SETLU: LD D, A
;installe lecteur
CALL GETLEC

```

```

; preserve DRIVE dans E
LD E, A
LD (RANDOS), A
ADD A, 41H
LD (AOUB), A
;installe user
LD A, D
CALL GETUSR
LD (RANDOS+1), A
;preserve source/dest et lecteur
LD B, D
LD C, E
;determine DPB correspondant 0/1
LD HL, DPB1
LD A, (BANSUP+8)
AND 00H
RLCA
CP B
JR Z, SETLU2
;sens 0 et dest ou sens 1 et source
LD HL, DPB2
;determine adresse du DPB selon A/B
SETLU2: LD DE, RANDOS+0190H
XOR A
CP C
JR Z, SETLU3
;pointe sur DPB B
LD DE, RANDOS+01D0H
;restaure source/dest
SETLU3: LD A, B
;recopie DPB 0/1 sur DPB A/B
LD BC, 0019H
LDIR
RET

;-----
;prepare lect/usr pour CAT/DIR
CATDIR: CALL SETLU
;echange fenetres 0 et source/dest.
LD B, A
INC B
PUSH BC
CALL SWAPWI
;affiche message
LD HL, INSERE
RST 00
LD HL, LECTEU
RST 00
RST 00
;attend touche ou ESC
CALL WETKEY
CP ESC
POP BC
;retour si pas de BREAK
RET NZ
;depile adresse de retour inutile
POP HL
;message d'abandon
LD HL, MESESC

```

```

RST 00
;restaure les fenetres
CALL SWAPW1
JP TOUCHE

;-----
;buffer txt pour bande sup
LECURS:DEFB 1FH,00,01,00,00,00,00,00

;-----
;
PACK de routines N02

ORG 0BBBAH
LOAD PACK2

;-----
;INPUT pour chargt ou effact

;init. pour buffer
INPUT: LD HL,BUFFER
LD B,00

;saisie de caracteres,ENTER,DEL,ESC
INPUT1:CALL WETKEY

CP ESC
;si ESC stopper tout. ESC->NO CARRY
RET Z

CP ENTER
JR NZ,INPUT2
;ENTER licite (nom donne)?
XOR A
CP B
JR Z,INPUT1
;met a jour la longueur de var.
LD A,B
LD (DESCR1),A
;lancer le chargement. RET->CARRY
SCF
RET

INPUT2:CP DEL
JR NZ,INPUT3
;DEL licite (+1 car)?
XOR A
CP B
JR Z,INPUT1
;effacer dernier caratere
LD A,08H
RST 30H
LD A,10H
RST 30H
;remet a jour le buffer
DEC HL
DEC B
JR INPUT1

;accepter car. dans limite 20-5A
INPUT3:CP 20H

```

```

JR C,INPUT1
CP 5BH
JR C,INPUT4
SUB 20H
JR INPUT3

;stocker car. Pas le 13e?
INPUT4:LD (HL),A
LD A,0CH
CP B
JR Z,INPUT1
;mise a jour buffer et affichage
LD A,(HL)
INC B
INC HL
RST 30H
JR INPUT1

;-----
;VERIFL:verifie si HL<=DE

;preserve regs DE et A
VERIFL:PUSH DE
LD B,A
;test
XOR A
EX DE,HL
SBC HL,DE
;C=OK - NC=on stoppe: HL>DE
CCF
;restaure registres
EX DE,HL
LD A,B
POP DE
RET

;-----
;STOP10:abandon pour LOAD/SAVE
STOP10:LD HL,MESESC
RST 00
CALL WETKEY
JP TRANSF

;-----
;OPENIN:ouverture de fichier
OPENIN:LD HL,BUFFER
LD A,(DESCR1)
LD B,A
;disk open in
JP 0BC77H

;-----
;SETIN:prepare pour charge/efface

;met lecteur et user source
SETIN: XOR A
CALL SETLU
;message de procedure
LD HL,INSERE

```

```

RST 00
RST 00
LD HL,LECTEU
RST 00
LD HL,NOMFIC
RST 00
RET

;-----
;SETOUT:prepare pour sauvegarde

;met lecteur et user dest.
SETOUT:LD (BUFF10+1),HL
LD A,01
CALL SETLU
;message pour DISC destination
LD HL,INSERE
RST 00
LD HL,DESTIN
RST 00
RST 00
;attend touche
CALL WETKEY
CP ESC
JR NZ,SETOU2
;abandon (depile adr. de retour)
POP HL
JP TRANSF

;message d'ecriture
SETOU2:LD HL,ECRIT
RST 00
;ouvre fichier en sortie
LD HL,BUFFER
BUFF10:LD DE,0000
LD A,(DESCR1)
LD B,A
CALL 0BC8CH
RET C

;abandon (depile adr. de retour)
POP HL
JP STOP10

;*****
;TXT en bas de RAM

ORG 0040H
LOAD 0040H

INSERE:DEFB 12,'Insere DISC ',00
SOURCE:DEFB 'source ',00
DESTIN:DEFB 'destination ',00
LECTEU:DEFB 'dans le lecteur ',
AOUB: DEFB 00,', puis ',00
PRESSE:DEFB 'presser une touche:',10,00
NOMFIC:DEFB 'entrer le nom du fichier',00
DEFB ': ',10,13,00

```

```

EFFACE:DEFB ' a effacer:',10,13,0
MESESC:DEFB 10,13,' * Abandon *',00
LIT: DEFB 10,13,'lit...',00
ECRIT: DEFB 13,'ecrit...',00,00
EFFIC:DEFB 10,13,'efface...',00

```

END

TX MENU

```

;*****
;* TEXTE MENU pour TRANSFIL *
;* copyright Y. GOUR 1991 *
;*****

```

```

ORG 9E80H
LOAD 9E80H

```

```
TXMENU: EQU 0BE80H
```

```

LD HL, TEXTE
LD DE, TXMENU
LD BC, 00D0H
LDIR

```

RET

```

TEXTE:DEFB 12
DEFB 31,6,1,0F2H,9AH," :lecteur source"
DEFB 31,31,1,9AH,0F3H," :lecteur destination"
DEFB 31,59,1,"TAB :source",0F2H,0F3H,"dest."
DEFB 31,7,2,0F0H," :user source"
DEFB 31,31,2,0F1H," :user destination"
DEFB 31,59,2,"DEL :efface"
DEFB 31,3,3,"F1/F4 :CAT/DIR source"
DEFB 31,28,3,"F2/F5 :CAT/DIR destination"
DEFB 31,59,3,"RET :transfere"

```

END

TRANSFIL

```

;*****
;*
;* TRANSFIL V1.0 - transfert *
;* entre tous formats *
;* copyright Y. GOUR (c)1991 *
;*
;*****

```

;adresses de donnees

```

TYPE: EQU 0007H
RANDOS: EQU 0AC00H

```

```

NOBLOC: EQU RANDOS+0019H
DPBA: EQU RANDOS+0190H
SPT: EQU 00H
BLM: EQU 03H
OFF: EQU 0DH
FSC: EQU 0FH
RPS: EQU 15H

```

```

PTRVAR: EQU 0AD64H
DESCRI: EQU 0AD66H
BUFFER: EQU 0AD69H
LOC: EQU 0AD75H
LEN: EQU 0AD77H
EXEC: EQU 0AD79H
HEXA: EQU 0AD7BH
LECUSR: EQU 0BDC5H
RDSEC: EQU 0BDF1H
WRSEC: EQU 0BDF4H
SEEK: EQU 0BDF7H
DIR: EQU 0BDFAH
ERA: EQU 0BDFDH
DPB1: EQU 0BE00H
DPB2: EQU 0BE19H
BANSUP: EQU 0BE32H
TXMENU: EQU 0BE80H

```

;adresses de routines

```

INPUT: EQU 0BBBAH
VERIFL: EQU 0BBFDH
STOP10: EQU 0BC08H
OPENIN: EQU 0BC12H
SETIN: EQU 0BC1CH
SETOUT: EQU 0BC2EH

```

```

SWAP03: EQU 0BD38H
SWAPW1: EQU 0BD3AH
WETKEY: EQU 0BD3FH
AFFLU: EQU 0BD4BH
AFFLU2: EQU 0BD4FH
GETLEC: EQU 0BD54H
GETUSR: EQU 0BD59H
SETLU: EQU 0BD6BH
CATDIR: EQU 0BDA1H

```

;valeurs et touches

```

LOMEM: EQU 0300H
HIMEM: EQU 0ABFBH
HAUT: EQU 0F0H
BAS: EQU 0F1H
GAUCHE: EQU 0F2H
DROITE: EQU 0F3H
TAB: EQU 09H
ENTER: EQU 0DH
ESC: EQU 0FCH
DEL: EQU 07FH
F1: EQU 31H

```

```

F2: EQU 32H
F4: EQU 34H
F5: EQU 35H

```

```

;*****
; TRANSFEREUR

```

```

ORG 0040H
LOAD 0040H

```

```

INSERE:DEFB 12,'Inserer DISC ',00
SOURCE:DEFB 'source ',00
DESTIN:DEFB 'destination ',00
LECTEU:DEFB 'dans le lecteur ',
AOUB: DEFB 00,' , puis ',00
PRESSE:DEFB 'presser une touche:',10,00
NONFIC:DEFB 'entrer le nom du fichier',00
DEFB ': ',10,13,00
EFFACE:DEFB ' a effacer:',10,13,0
MESESC:DEFB 10,13,' * Abandon *',00
LIT: DEFB 10,13,'lit...',00
ECRIT: DEFB 13,'ecrit...',00,00
EFFIC:DEFB 10,13,'efface...',00

```

```

MENU7: CP F1
JR NZ, MENU8

```

```

;catalogue source
XOR A
LBL4: CALL CATDIR
PUSH BC
LD DE, LOMEM
CALL 0BC9BH
JR LBL6

```

```

MENU8: CP F2
JR NZ, MENU9
;catalogue dest.
LD A, 01
JR LBL4

```

```

MENU9: CP F4
JR NZ, MENUA

```

```

;directory source
XOR A
LBL5: CALL CATDIR
PUSH BC
XOR A
RST 10H
DEFW DIR
LBL6: POP BC
CALL SWAPW1
JR TOUCHE

```

```

MENUA: CP F5
JR NZ, TOUCHE
;directory destination
LD A, 01
JR LBL5

```

```

;affiche le Menu en fenetre 3
MENU: CALL SWAP#3
      LD HL, TXMENU
      RST #0
      CALL SWAP#3
;attend frappe de touche
TOUCHE: CALL WETKEY

      CP GAUCHE
      JR NZ, MENU2
;change lecteur source
      XOR A
;saïsit lecteur
LBL1: CALL GETLEC
;commute lecteur
      XOR #1
      LD (HL), A
;ecrit lettre A/B
      ADD A, 41H
LBL7: LD (LECUSR+3), A
;ecrit position de lettre
      INC HL
      LD A, (HL)
      LD (LECUSR+1), A
;remet a jour affichage
      CALL AFFLU
      JR TOUCHE

MENU2: CP DROITE
      JR NZ, MENU3
;change lecteur dest.
      LD A, #1
      JR LBL1

MENU3: CP HAUT
      JR NZ, MENU4
;change user source
      XOR A
;saïsit user
LBL2: CALL GETUSR
;incremente user
      INC A
      AND #FH
      LD (HL), A
;prepare index BC pour affichage
      LD B, #0
      LD C, A
      EX DE, HL
;pointe caractere pour User
      LD HL, HEXA
      ADD HL, BC
      LD A, (HL)
;restaure HL pour position de User
      EX DE, HL
      JR LBL7

MENU4: CP BAS
      JR NZ, MENU5

```

```

;change user dest.
      LD A, #1
      JR LBL2

MENU5: CP TAB
      JR NZ, MENU6
;change sens de copie
      LD HL, BANSUP+8
      LD A, (HL)
      XOR #0H
      LD (HL), A
      LD B, A
;ecrit position pour affichage
      AND 7FH
      LD (LECUSR+1), A
      INC HL
;determine sens de la fleche
      BIT 7, B
      JR NZ, LBL3
      INC HL
;recopie fleche pour affichage
LBL3: LD DE, LECUSR+3
      LD BC, #004
      LDIR
      CALL AFFLU2
      JR TOUCHE

MENU6: CP ENTER
      JR Z, TRANS#
      CP DEL
      JP NZ, MENU7

;=====
; EFFACEMENT DE FICHIER

;affiche message
DELETE: CALL SWAP#3
        CALL SETIN
        LD HL, EFFACE
        RST #0
;saïsit nom de fichier
        CALL INPUT
        JR NC, FINCOP

;message d'effacement
        LD HL, EFFIC
        RST #0
;efface fichier
        LD A, #1
        LD IX, PTRVAR
        RST 10H
        DEFW ERA

FINCOP: CALL SWAP#3
        JP MENU

;=====
; TRANSFERT DE FICHIER

```

```

;-----
;ferme par default tout fichier ouvert

TRANS#: CALL SWAP#3
TRANSF: CALL #BC7DH
        CALL #BC92H
;affiche message
        CALL SETIN
        RST #0
;saïsit nom de fichier
        CALL INPUT
        JR NC, FINCOP

;-----
;ouvre fichier

;message de lecture
        LD HL, LIT
        RST #0
        LD DE, HIMEM-2#48
        CALL OPENIN
;si NC, laisse beton pour ce fichier
        JR NC, TRANSF

;bifurque selon type de fichier
        CP #4
        JR C, BASBIN

;*-----*
;fichier ASCII

;lit tous les octets
        LD HL, LONEM
        LD DE, HIMEM-2#48
ASCII2: CALL #BC8#H
        JR NC, ASCII3
        LD (HL), A
        INC HL
;verifie si fichier pas trop long
        CALL VERIFL
        JR C, ASCII2
;trop long: abandon
        XOR A

ASCII3: CP 1AH
;erreur imprevue: stop!
        JP NZ, STOPIO
;stocke adr. du dernier octet
        DEC HL
        LD (LOC), HL

;ferme le fichier
        CALL #BC7AH
        JP NC, STOPIO

;-----
;sauvegarde fichier

        LD HL, HIMEM-2#48

```

```

CALL SETOUT
;envoie les octets un par un
LD HL,LOMEM
LD DE,(LOC)
ASCII4:LD A,(HL)
CALL #BC95H
JP NC,STOPI0
INC HL
CALL VERIFL
JR C,ASCII4

;ferme le fichier
CALL #BC8FH
JP NC,STOPI0
JR TRANSF

;-----
;stocke LOC/LEN/TYPE pour BAS/BIN

BASBIN:LD (LOC),DE
LD (LEN),BC
LD (TYPE),A

;fichier BASIC/BINAIRE pas trop long?
LD HL,LOMEM
ADD HL,BC
DEC HL
LD DE,HIMEM
CALL VERIFL
JP NC,TRANSF

;-----
;charge fichier

LD HL,LOMEM
CALL #BC83H
;abandon si NC
JP NC,STOPI0
;stocke adresse d'EXEC
LD (EXEC),HL
CALL #BC7AH
;abandon si NC
JP NC,STOPI0

;-----
;sauvegarde fichier

LD HL,LOMEM
CALL SETOUT
LD HL,LOMEM
LD DE,(LEN)
LD BC,(EXEC)
LD A,(TYPE)
CALL #BC96H
JP NC,STOPI0

;ferme le fichier
CALL #BC8FH

```

```

JP NC,STOPI0

;-----
;correction du header de fichier

;ouvre fichier pour lire 1er bloc
LD DE,LOMEM
CALL OPENIN
JP NC,STOPI0

;calcule adr. DBP (IX) selon lecteur
;IX=adr. DPB A par default
LD IX,DBPA
LD A,(RANDOS)
CP #
JR Z,DDA#

;IX=adr. DPB B
LD IX,DPBA+#4#H

;releve no de bloc
DDA#: LD A,(NOBLOC)
LD B,A

;nb de records (128 octets) par bloc
LD A,(IX+BLM)
INC A
LD D,#
LD E,A

;nb records de 128 octets (=) NOBLOC
LD HL,###
DDA1: ADD HL,DE
DJNZ DDA1

;1er record du bloc recherche
INC HL

;lit nb de records par piste
LD A,(IX+SPT)
LD E,A
XOR A
;compte nb de pistes ->NOBLOC
DDA2: SBC HL,DE
INC A
JR NC,DDA2

;rajoute DE pour calcul de no de secteur
ADD HL,DE

;stocke piste
DEC A
LD D,A

;ajoute decalage de piste
LD A,(IX+OFF)
ADD A,D
LD D,A

;lecture du Drive
LD A,(RANDOS)
LD E,A

;nb de records par secteur
LD A,(IX+RPS)
LD C,A

```

```

;recupere reste de piste dans A
LD A,L
;calcule no logique de secteur (H)
DDA3: INC H
SUB C
JR NC,DDA3
;ajuste H
DEC H
;lit no physique du 1er secteur
LD A,(IX+FSC)
;stocke no de secteur du bloc vise
ADD A,H
LD C,A

;lit secteur du HEADER de fichier
LD HL,LOMEM
PUSH IX
PUSH BC
PUSH DE
PUSH HL
RST 16H
DEFW RDSEC

;ecrit vraie adr. de loc.
LD HL,(LOC)
LD (LOMEM+15H),HL
;calcule la CHECKSUM
LD HL,###
POP DE
PUSH DE
LD B,43H
DDA4: PUSH BC
LD A,(DE)
LD B,#
LD C,A
ADD HL,BC
INC DE
POP BC
DJNZ DDA4
;ecrit la CHECKSUM
LD (LOMEM+43H),HL

;ecrit secteur HEADER modifie
POP HL
POP DE
POP BC
PUSH DE
RST 16H
DEFW WRSEC
;retour en premiere piste
POP DE
POP IX
LD A,(IX+OFF)
LD D,A
RST 16H
DEFW SEEK
JP TRANSF

END ■

```