

## 9/10.2

# Gestion de logiciels

---

### I - Présentation

Votre bibliothèque de logiciels s'enrichit de jour en jour, et le nombre de disquettes que vous possédez frôle la centaine... Quoi de plus exaspérant que de devoir lire le catalogue de 30 disquettes avant de retrouver le logiciel « Trucmuch », que vous avez écrit il y a un mois, pour en faire une démonstration à un ami !

Une seule solution : un répertoire de logiciels. Sur papier ? A quoi bon posséder un outil de programmation tel que l'Amstrad ? Vous pourrez créer ce répertoire grâce à dBase II, mais le prix de revient serait relativement élevé pour la maigre utilisation que vous feriez face à la puissance de ce logiciel. Une autre solution, la bonne : suivre les conseils qui suivent afin d'écrire soi-même sa gestion de logiciel.

Nous vous proposons donc un programme qui vous permettra de gérer votre stock de logiciels avec quelques similitudes envers dBase II, et de retrouver du premier coup un programme parmi vos disquettes. Nous vous proposerons ensuite la méthode d'adaptation à pratiquement tout ce que vous désirez classer.

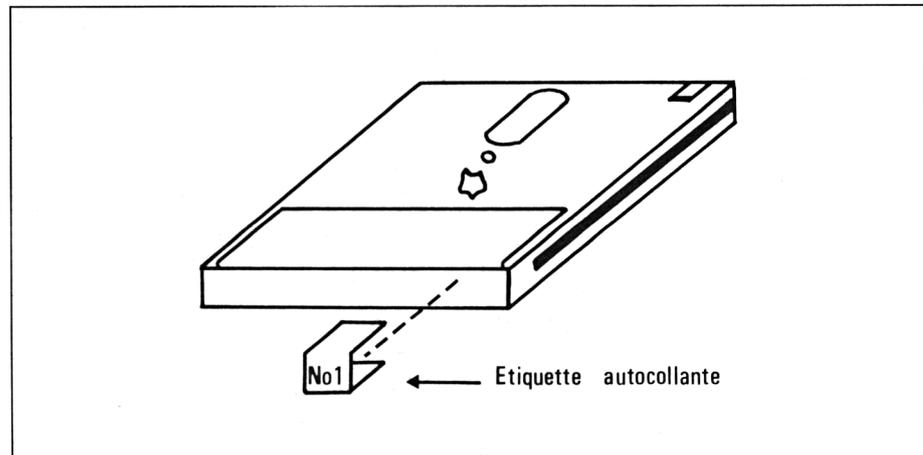
Il nous permettra, de plus, d'étudier une méthode de tri intéressante : le tri par SHELL-METZNER.

Ce programme développé sur Amstrad CPC6128 tourne sur CPC664 et s'adapte sans problème sur CPC464 muni d'un lecteur de disquette (rapelons que la gestion de fichiers se justifie surtout par l'utilisation d'un lecteur de disquette, le lecteur de cassettes, de par sa fiabilité et sa relative lenteur, ne permettant pas un chargement ou une sauvegarde rapide des fichiers gérés).

### II - Organisation matérielle et logiciel

#### LE SUPPORT : LES DISQUETTES

Si vous voulez retrouver rapidement une disquette : une seule solution : le marquage. Vous serez libre du marquage, mais nous vous proposons une méthode simple et rapide : la numérotation algébrique, sur le côté supérieur de la disquette.



#### Numérotation de disquette.

Vous pouvez maintenant ranger vos disquettes sans leur boîtier, dans une boîte prévue à cet effet (ne craignez pas de vous passer de vos boîtiers, les disquettes sont protégées par une lumière qui s'ouvre uniquement lors de l'insertion dans le lecteur).

### ORGANISATION DU LOGICIEL

#### Les caractéristiques d'un logiciel stocké

Il nous faut d'abord définir les caractéristiques à retenir pour chaque logiciel, afin de définir les différentes zones de chacun.

Les variables seront définies en même temps :

DESI\$ : désignation, ou nom, du logiciel

CODE\$ : codage personnel

REMA\$ : remarque concernant le logiciel

DK\$ : référence de la disquette

NBKO\$ : taille du logiciel

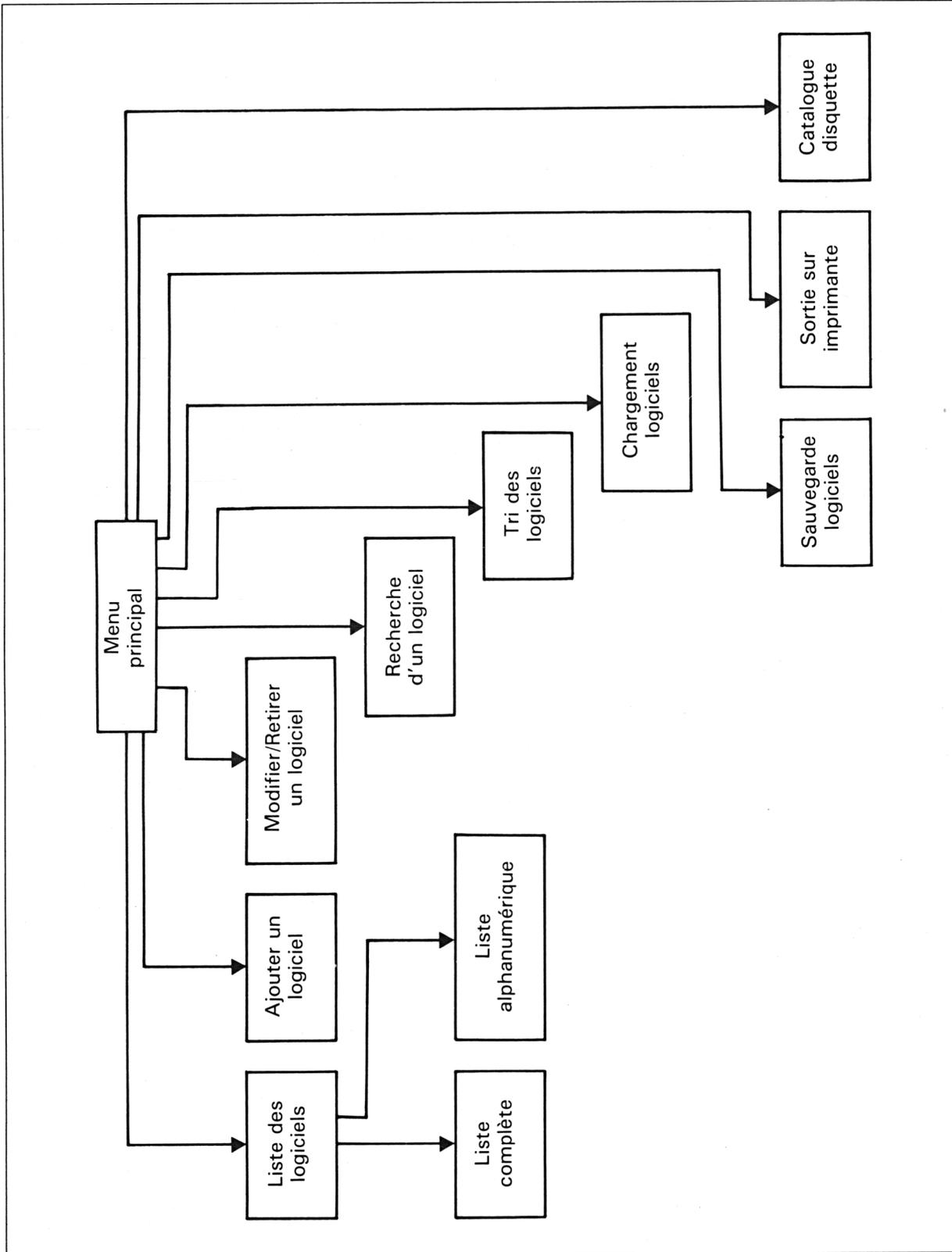
#### Le menu

Nous vous proposons un menu de neuf options, organisées selon le schéma suivant :

- \* Liste des logiciels qui sera décomposée en deux options :
  - liste entière
  - liste à partir d'une position alphabétique.
- \* Ajouter un logiciel.
- \* Corriger/Supprimer un logiciel (si vous désirez supprimer un logiciel il vous suffira d'entrer le caractère ! à la place de la désignation — le logiciel sera sélectionné par son numéro d'ordre).

- \* Recherche d'un logiciel par caractéristique.
- \* Tri alphabétique des logiciels.
- \* Chargement de la liste.
- \* Sauvegarde de la liste.
- \* Sortie sur imprimante.
- \* Catalogue disquette.

Toutes ces opérations étant définies, nous pouvons dire que le programme fonctionnera sous une structure de type arborescente.



Organisation logicielle.

Vous retrouverez ce menu dans la copie d'écran suivante :

```

PROGRAMME GESTION DE LOGICIELS                                MEMOIRE LIBRE : 24
                                                                NB DE LOGICIEL : 0
                                                                MENU
( 1 ) LISTE DES LOGICIELS .....
( 2 ) AJOUTER UN LOGICIEL .....
( 3 ) CORRIGER/ELIMINER UN LOGICIEL ....
( 4 ) RECHERCHE PAR CARACTERISTIQUE ....
( 5 ) TRI DES DONNEE .....
( 6 ) CHARGEMENT DU FICHER LOGICIELS ..
( 7 ) SAUVEGARDE DU FICHER LOGICIELS...
( 8 ) SORTIE SUR IMPRIMANTE .....
( 9 ) CATALOGUE DISQUETTE .....
-----> 6

```

Copie d'écran menu.

Nous imposerons de plus un retour au menu principal dès qu'une erreur de saisie surviendra.

La saisie ou la correction sera présentée de la même façon que celle de dBase II, c'est-à-dire par champs à remplir, telle la copie d'écran donnée ci-dessous :

```

ENTRER LE NO DU LOGICIEL A CORRIGER : ? 5

DESIGNATION ..... : CCP..... LOGICIEL No ..... : 5
CODE ..... : .. PLACE LIBRE ..... : 20 KO
REMARQUE ..... : GESTION DE COMPTE.....
NOMBRE DE KO ..... : 12
DISQUETTE No ..... : 1.....

```

Les champs de saisie.

Vous pourrez aussi garder une trace écrite de tous vos logiciels, pour vérifier ou tout simplement pour donner à un de vos amis qui pourra vous demander une copie d'un logiciel particulier, après avoir choisi sans semer la zizanie dans le classement de vos disquettes.

	Designation	Cod	Remarques	Ko	No dk
* 1	* AWARI	*	* BASIC	* 2	* 6
* 2	* BIORITM	*	* BASIC	* 5	* 2
* 3	* CALEND	*	* WEKA	* 6	* 3
* 4	* CASSBRIK	*	* BASIC	* 3	* 1
* 5	* CCP	*	* GESTION DE COMPTE	* 12	* 1
* 6	* CHAUFFAG	*	* REGULATION CHAUFFAGE	* 3	* 8
* 7	* DUMP	*	* BASIC WEKA	* 2	* 8
* 8	* HANDI	*	* WEKA	* 6	* 8
* 9	* MORSE	*	* WEKA	* 1	* 6
* 10	* OTHELLO	*	* JEU	* 33	* 6
* 11	* PIP	*	* PIP BASIC WEKA	* 2	* 3
* 12	* PROGRAM	*	* PROGRAMMATEUR MEMOIRE	* 3	* 12
* 13	* RECEP232	*	* RECEP. FICHER RS232	* 1	* 4
* 14	* SONNERIE	*	* DETECTEUR SONNERIE	* 1	* 7
* 15	* TAQUIN	*	* WEKA	* 4	* 8
* 16	* WEKASERV	*	* SERVEUR	* 27	* 2

Exemple de liste de logiciels sur imprimante.

### III - Le programme

Vous trouverez ci-après le listing du programme qui ne devrait poser aucun problème à la saisie, si ce n'est de ne pas confondre les caractères zéro et O majuscule.

Dans un premier temps, nous vous conseillons de ne pas taper la ligne 110 qui branche le programme à une routine de traitement d'erreur. Vous essayez votre programme (après l'avoir sauvegardé) et, après avoir corrigé les éventuelles erreurs de frappe (*Syntax Error*), vérifié sur un fichier d'essai que tout fonctionne correctement, vous pourrez insérer cette ligne et sauvegarder de nouveau le programme.

```

10 REM *****
20 REM ***** GESTION DE LOGICIELS *****
30 REM *****
40 OPENOUT"essai."
50 CLOSEOUT
60 MODE 2
70 INK 0,0:INK 1,26:PEN 1:PAPER 0:BORDER
  0
80 DIM desi$(500),code$(500),rema$(500),
  dk$(500),NBK0$(500)
90 sursauv=1
100 REM ***** MENU *****
110 ON ERROR GOTO 2940
120 CLS
130 LOCATE 1,2:PRINT"PROGRAMME GESTION D
  E LOGICIELS"
140 LOCATE 50,2:PRINT"MEMOIRE LIBRE : "
150 LOCATE 67,2:PRINT(INT(FRE(0)/1024))
160 LOCATE 50,4:PRINT"NB DE LOGICIEL : "
170 FOR I=1 TO 500
180 IF DESI$(I)="" THEN GOTO 200
190 NEXT I
200 LOCATE 67,4:PRINT I-1:tri=(i-1)
210 LOCATE 28,5:PRINT"          MENU "
220 LOCATE 22,7:PRINT"( 1 ) LISTE DES LO
  GICIELS ....."
230 LOCATE 22,9:PRINT"( 2 ) AJOUTER UN L
  OGICIEL ....."
240 LOCATE 22,11:PRINT"( 3 ) CORRIGER/EL
  IMINER UN LOGICIEL ...."
250 LOCATE 22,13:PRINT"( 4 ) RECHERCHE P
  AR CARACTERISTIQUE ...."
260 LOCATE 22,15:PRINT"( 5 ) TRI DES DON
  NEES....."
270 LOCATE 22,17:PRINT"( 6 ) CHARGEMENT
  DU FICHER LOGICIELS .."
280 LOCATE 22,19:PRINT"( 7 ) SAUVEGARDE
  DU FICHER LOGICIELS .."
290 LOCATE 22,21:PRINT"( 8 ) SORTIE SUR
  IMPRIMANTE ....."
300 LOCATE 22,23:PRINT"( 9 ) CATALOGUE D
  ISQUETTE ....."
310 CLEAR INPUT
320 A$=INKEY$
330 LOCATE 10,2:PRINT "          "
340 LOCATE 11,2:PRINT CHR$(24);"GESTION"
  ;CHR$(24):IF A$="" THEN 320
350 LOCATE 28,25:PRINT "-----> ";A$
360 IF A$="+" OR A$="&" OR A$="-" OR A$="
  ." THEN GOTO 100
370 A=VAL(A$)

```

```

380 IF A<1 OR A>9 THEN 100
390 ON a GOTO 400,880,1170,1640,2710,192
0,2230,2400,2520
400 REM ***** LISTE DES LOGICIEL
S *****
410 CLS
420 LOCATE 28,5:PRINT"          MENU LIS
TE"
430 LOCATE 22,7:PRINT"( 1 ) LISTE ENTIER
E ....."
440 LOCATE 22,9:PRINT"( 2 ) LISTE PAR LE
TTRE ALPHANUMERIQUE .."
450 CLEAR INPUT
460 A$=INKEY$:IF A$="" THEN 460
470 LOCATE 28,25:PRINT "-----> ";A$
480 IF A$="+" OR A$="&" OR A$="-" OR A$=
"." THEN GOTO 100
490 A=VAL(A$)
500 IF A<1 OR A>2 THEN 100
510 ON a GOTO 520,680
520 REM ***** LISTE ENTIERE *****
530 CLS:GOSUB 2690:A=1
540 FOR I=1 TO 500
550 IF DESI$(I)="" THEN PRINT "*****
***** FIN *****
*****":CALL &
BB06:GOTO 10
0
560 IF A=21 THEN PRINT"          Ta
per <enter> pour suite, <S> pour stopper
"
570 IF A=21 THEN u$=INKEY$:IF u$="s" OR
u$="S" THEN 100 ELSE IF u$<>CHR$(13) THE
N 570 ELSE A=1:CLS:GOSUB 2690
580 LOCATE 2,A+4:PRINT"*"
590 LOCATE 4,A+4:PRINT DESI$(I)
600 LOCATE 28,A+4:PRINT"*"
610 LOCATE 30,A+4:PRINT NBK0$(I)
620 LOCATE 48,A+4:PRINT"*"
630 LOCATE 50,A+4:PRINT DK$(I)
640 LOCATE 73,A+4:PRINT"*"
650 LOCATE 75,A+4:PRINT I
660 A=A+1
670 NEXT I
680 REM ***** LISTE ALPHANUMERIQUE **
*****
690 CLS:INPUT"ENTREZ LA OU LES LETTRES D
U LOGICIEL A RECHERCHER : ",L$
700 L$=UPPER$(L$)
710 CLS:GOSUB 2690

```

```

720 A=1
730 FOR I=1 TO 500
740 IF DESI$(I)="" THEN PRINT "*****
***** FIN *****
*****":CALL &
BB06:GOTO 10
O
750 IF A=21 THEN CALL &BB06:A=1:CLS:GOSU
B 2690
760 IF LEFT$(DESI$(I),LEN(L$)) <> L$ THE
N 860
770 LOCATE 2,A+4:PRINT"*"
780 LOCATE 4,A+4:PRINT DESI$(I)
790 LOCATE 28,A+4:PRINT"*"
800 LOCATE 30,A+4:PRINT NBKO$(I)
810 LOCATE 48,A+4:PRINT"*"
820 LOCATE 50,A+4:PRINT DK$(I)
830 LOCATE 73,A+4:PRINT"*"
840 LOCATE 75,A+4:PRINT I
850 A=A+1
860 NEXT I
870 CALL &BB06:GOTO 100
880 REM ***** AJOUT D'UN LOGICIEL *
*****
890 sursauv=0
900 FOR I=1 TO 500:IF DESI$(I)="" THEN G
OTO 910 ELSE NEXT I
910 CLS
920 LOCATE 30,3:PRINT"AJOUTER UN LOGICIE
L"
930 GOSUB 2540'GRILLE 1
940 LOCATE 72,7:PRINT I
950 LOCATE 72,10:X=INT((FRE(0)/1000)):PR
INT X;" KO"
960 LOCATE 25,7:LINE INPUT "",DESI$(I)
970 DESI$(I)=UPPER$(DESI$(I))
980 IF DESI$(I)="" THEN GOTO 100
990 IF LEN(DESI$(I))>24 THEN DESI$(I)=""
:GOTO 2630
1000 LOCATE 25,10:LINE INPUT "",CODE$(I)
1010 CODE$(I)=UPPER$(CODE$(I))
1020 IF LEN(CODE$(I))>3 THEN DESI$(I)=""
:GOTO 2660
1030 LOCATE 25,13:LINE INPUT "",REMA$(I)
1040 REMA$(I)=UPPER$(REMA$(I))
1050 IF LEN(REMA$(I))>24 THEN DESI$(I)=""
:GOTO 2630
1060 LOCATE 25,16:LINE INPUT "",NBKO$(I)
1070 NBKO$(I)=UPPER$(NBKO$(I))
1080 IF LEN(NBKO$(I))>3 THEN DESI$(I)=""

```

```

:GOTO 2660
1090 LOCATE 25,19:LINE INPUT "",DK$(I)
1100 DK$(I)=UPPER$(DK$(I))
1110 IF LEN(DK$(I))>24 THEN DESI$(I)="":
GOTO 2630
1120 LOCATE 18,23:PRINT"VALIDATION OUI =
>°ENTERS, NON => °NS"
1130 A$=INKEY$
1140 IF A$=CHR$(13) THEN I=I+1:GOTO 910
1150 IF A$="N" OR A$="n" THEN DESI$(I)="
":a$="ANNULATION":GOSUB 2920:GOTO 910
1160 LOCATE 18,23:PRINT"
":GOTO 1120
1170 REM ***** CORRECTION / ANNULAT
ION D'UN LOGICIEL *****"
1180 sursauv=0
1190 CLS :INPUT "ENTRER LE NO DU LOGICIE
L A CORRIGER : ";NUM
1200 GOSUB 2540:GOSUB 1210:GOTO 1280
1210 LOCATE 72,7:PRINT NUM
1220 LOCATE 72,10:X=INT((FRE(0)/1024)):P
RINT X;" KO"
1230 LOCATE 25,7:PRINT DESI$(NUM)
1240 LOCATE 25,10:PRINT CODE$(NUM)
1250 LOCATE 25,13:PRINT REMA$(NUM)
1260 LOCATE 25,16:PRINT NBKO$(NUM)
1270 LOCATE 25,19:PRINT DK$(NUM):RETURN
1280 D$=DESI$(NUM)
1290 C$=CODE$(NUM)
1300 R$=REMA$(NUM)
1310 N$=NBKO$(NUM)
1320 NK$=NBKO$(NUM)
1330 DK$=DK$(NUM)
1340 'CORRECTION EFFECTIVE
1350 LOCATE 25,7:INPUT "",DESI$(NUM)
1360 DESI$(NUM)=UPPER$(DESI$(NUM))
1370 IF DESI$(NUM)="" THEN DESI$(NUM)=D$
1380 IF DESI$(NUM)="!" THEN GOTO 1600
1390 IF LEN(DESI$(NUM))>24 THEN GOTO 263
0
1400 LOCATE 25,10:LINE INPUT "",CODE$(NU
M)
1410 CODE$(NUM)=UPPER$(CODE$(NUM))
1420 IF CODE$(NUM)="" THEN CODE$(NUM)=C$
1430 IF LEN(CODE$(NUM))>3 THEN CODE$(NUM
)="":GOTO 2660
1440 LOCATE 25,13:LINE INPUT "",REMA$(NU
M)
1450 REMA$(NUM)=UPPER$(REMA$(NUM))
1460 IF REMA$(NUM)="" THEN REMA$(NUM)=R$
1470 IF LEN(REMA$(NUM))>24 THEN REMA$(NU

```

```

M="":GOTO 2630
1480 LOCATE 25,16:LINE INPUT "",NBKO$(NUM)
1490 NBKO$(NUM)=UPPER$(NBKO$(NUM))
1500 IF NBKO$(NUM)="" THEN NBKO$(NUM)=NK$
1510 IF LEN(NBKO$(NUM))>3 THEN NBKO$(NUM)="" :GOTO 2660
1520 LOCATE 25,19:LINE INPUT "",DK$(NUM)
1530 DK$(NUM)=UPPER$(DK$(NUM))
1540 IF DK$(NUM)="" THEN DK$(NUM)=DK$
1550 IF LEN(DK$(NUM))>24 THEN DK$(NUM)="" :GOTO 2630
1560 CLS:LOCATE 30,3:PRINT"VERIFICATION"
1570 GOSUB 2540:GOSUB 1210
1580 CALL &BBO6::GOTO 100
1590 REM *** SUPPRESSION D'UN LOGICIEL **
*
1600 FOR J = NUM+1 TO I
1610 DESI$(J-1) = DESI$(J)
1620 NEXT J
1630 GOTO 100
1640 REM ***** CARACTERISTIQUES D'UN LOGICIEL *****"
1650 CLS:LOCATE 30,3:PRINT"RECHERCHE PAR
:"
1660 LOCATE 22,9:PRINT"( 1 ) LA DESIGNATION ....."
1670 LOCATE 22,11:PRINT"( 2 ) LE NUMERO ....."
1680 LOCATE 22,13:PRINT"( 3 ) LE NOMBRE DE KD ....."
1690 LOCATE 22,15:PRINT"( 4 ) RETOUR AU MENU PRINCIPAL ....."
1700 FOR I=1 TO 100:A$=INKEY$:NEXT I
1710 A$=INKEY$:IF A$="" THEN 1710
1720 LOCATE 28,17:PRINT "----> ";A$
1730 IF A$="+" OR A$="&" OR A$="-" OR A$="." THEN GOTO 1640
1740 A=VAL(A$)
1750 IF A<1 OR A>4 THEN 1640
1760 ON A GOTO 1770,1840,1870,100
1770 REM ***** RECHERCHE PAR DESIGNATION *****
1780 CLS:INPUT "ENTRER LE NOM DU LOGICIEL A TROUVER : ",RECHERCHE$
1790 RECHERCHE$=UPPER$(RECHERCHE$)
1800 IF RECHERCHE$="" THEN 100
1810 FOR I=1 TO 500
1820 IF RECHERCHE$=DESI$(I) THEN NUM=I:G

```

```

OSUB 2540:GOSUB 1210:CALL &BBO6
1830 NEXT I:GOTO 100
1840 CLS:INPUT "ENTRER LE NUMERO DU LOGI
CIEL A TROUVER : ",RECHERCHE#
1850 I=VAL(RECHERCHE#)
1860 NUM=I:GOSUB 2540:GOSUB 1210:CALL &B
BO6:GOTO 100
1870 CLS:INPUT "ENTRER LE NOMBRE DE KO :
",RECHERCHE#
1880 FOR I=1 TO 500
1890 IF RECHERCHE#=NBKO#(I) THEN NUM=I:G
OSUB 2540:GOSUB 1210:CALL &BBO6
1900 NEXT I
1910 GOTO 100
1920 REM***** CHARGEMENT DU FICHIER
*****
1930 IF SURSAUV=0 THEN GOSUB 2190
1940 IF SURSAUV=1 THEN 2030 ELSE IF UPPE
R#(REP#) <> "0" THEN GOTO 100
1950 FOR I=1 TO 500
1960 DESI#(I)="
1970 CODE#(I)="
1980 REMA#(I)="
1990 NBKO#(I)="
2000 DK#(I)="
2010 NEXT I
2020 PRINT FRE("")
2030 CLS
2040 PRINT"NOM DU FICHIER A ";CHR$(24);"
CHARGER";CHR$(24)
2050 LOCATE 54,1:LINE INPUT"",REP#
2060 IF REP#="" THEN 100
2070 CLS:LOCATE 10,10:PRINT"CHARGEMENT E
N COURS."
2080 OPENIN REP#
2090 FOR I=1 TO 500:LOCATE 20,20:PRINT i
2100 INPUT#9,DESI#(I):IF DESI#(I)=" THE
N 2160
2110 INPUT#9,CODE#(I)
2120 INPUT#9,REMA#(I)
2130 INPUT#9,DK#(I)
2140 INPUT#9,NBKO#(I)
2150 NEXT I
2160 CLOSEIN
2170 SURSAUV=1
2180 GOTO 100
2190 CLS:PRINT"FICHIER EN MEMOIRE MODIFI
E!"
2200 PRINT:PRINT"EFFACE LE FICHIER?(O/N)
"

```

```

2210 LINE INPUT "",REP#
2220 RETURN
2230 REM ***** SAUVEGARDE DE FICHIER
*****
2240 CLS
2250 PRINT"NOM DU FICHIER A ";CHR$(24);"
SAUVEGARDER";CHR$(24)
2260 LOCATE 54,1:LINE INPUT "",REP#
2270 IF REP#="" THEN 100
2280 CLS:LOCATE 10,10:PRINT"SAUVEGARDE E
N COURS."
2290 OPENOUT REP#
2300 FOR I=1 TO 500:LOCATE 20,20:PRINT i
2310 WRITE#9,DESI$(I)
2320 WRITE#9,CODE$(I)
2330 WRITE#9,REMA$(I)
2340 WRITE#9,DK$(I)
2350 WRITE#9,NBKO$(I):IF DESI$(I)="" THE
N 2370
2360 NEXT I
2370 CLOSEOUT
2380 SURSAUV = 1
2390 GOTO 100
2400 REM ***** SORTIE IMPRIMANTE *
*****
2410 CLS:LOCATE 10,10:PRINT"SORTIE SUR I
MPRIMANTE EN COURS."
2420 REP = INF(&F500)
2430 IF REP <> 30 THEN 3000
2440 PRINT#8,CHR$(27);CHR$(64);CHR$(27);
CHR$(15)
2450 PRINT #8,TAB(10);"Designation";TAB
(36);"Cod";TAB (42);"Remarques";TAB (69)
;"Ko";TAB(75);"No dk":PRINT #8
2460 FOR I=1 TO 500
2470 IF DESI$(I)="" THEN LOCATE 15,12:PR
INT"FIN D'IMPRESSION ...":FOR XX=1 TO 20
00:NEXT XX:PRINT #8,CHR$(12):GOTO 100
2480 PRINT #8,TAB(1);"*";TAB(2);i;TAB(7)
" * ";TAB(10);DESI$(I);TAB (33);" * ";CO
DE$(I);TAB (39);" * ";REMA$(I);TAB (66);
" * ";NBKO$(
I);TAB(72);" * ";DK$(I)
2490 NEXT I
2500 PRINT#8, CHR$(12);CHR$(27);CHR$(64)
2510 GOTO 100
2520 '***** CATALOGUE
DISQUETTE *****
2530 MODE 2:CAT:CALL &BBO6:GOTO 100
2540 '***** GRILLE ECR

```

```

AN *****
2550 LOCATE 3,7:PRINT"DESIGNATION .....
   : ";CHR$(24);".....":
PRINT CHR$(24)
2560 LOCATE 3,10:PRINT"CODE .....
   . : ";CHR$(24);"...":PRINT CHR$(24)
2570 LOCATE 3,13:PRINT"REMARQUE .....
   . : ";CHR$(24);"....."
   :PRINT CHR$(24)
2580 LOCATE 3,16:PRINT"NOMBRE DE KO ....
   . : ";CHR$(24);"...":PRINT CHR$(24)
2590 LOCATE 3,19:PRINT"DISQUETTE No ....
   . : ";CHR$(24);"....."
   :PRINT CHR$(24)
2600 LOCATE 50,7:PRINT"LOGICIEL No .....
   . : ";CHR$(24);"."":PRINT CHR$(24)
2610 LOCATE 50,10:PRINT"PLACE LIBRE ....
   .. : ";CHR$(24);"."":PRINT CHR$(24)
2620 RETURN
2630 '***** LONGUEUR D
ES DONNEES TROP GRANDES *****
2640 CLS:PRINT"***** TROP LONG,
PAS PLUS DE 24 CARACTERES !!! *****
****"
2650 CALL &BB06:GOTO 100
2660 '***** LONGUEUR D
ES DONNEES TROP GRANDES *****
2670 CLS:PRINT"***** TROP LONG,
PAS PLUS DE 3 CARACTERES !!! *****
***"
2680 CALL &BB06:GOTO 100
2690 '***** ENTETE *****
2700 LOCATE 1,1:PRINT"LISTE DES LOGICIEL
S":LOCATE 3,3:PRINT "DESIGNATION":LOCATE
   30,3:PRINT "NOMBRE DE K.O":LOCATE 50,3:
PRINT "No DE
   LA DISQUETTE":RETURN
2710 '***** tri
   par shell-metzner *****
2720 SURSAUV=0
2730 m=tri
2740 garde=tri
2750 CLS:LOCATE 10,10:PRINT"TRI EN COURS
, PATIENCE ..."
2760 p=m
2770 p=INT(p/2)
2780 IF p<1 THEN GOTO 100
2790 deb=1:fin=m-p
2800 r=deb
2810 c=r+p
2820 IF c<garde THEN LOCATE 20,20:PRINT
c:garde=c

```

```

2830 IF desi$(r)<=desi$(c) GOTO 2890
2840 aa$=desi$(r):desi$(r)=desi$(c):desi
$(c)=aa$
2850 sauv2$=code$(r):sauve3$=rema$(r):sa
uve4$=dk$(r):sauve5$=nbko$(r)
2860 code$(r)=code$(c):rema$(r)=rema$(c)
:dk$(R)=dk$(c):nbko$(r)=nbko$(c)
2870 code$(c)=sauv2$:rema$(c)=sauve3$:dk
$(c)=sauve4$:nbko$(c)=sauve5$
2880 r=r-p:IF r>0 GOTO 2810
2890 deb=deb+1:IF deb>fin THEN GOTO 2770
ELSE GOTO 2800
2900 GOTO 100
2910 REM ***** ANNULATION SONORE ***
*****
2920 FOR i=1 TO LEN(a$):PRINT MID$(a$,i,
1);CHR$(143);CHR$(8);:SOUND 1,20,3,15,,
10:FOR x=1 TO 30:NEXT:NEXT:PRINT" ":RETU
RN
2930 '*****
*****
2940 REM ***** ERREUR SUR L'UNITE DE D
ISQUETTE *****
2950 PRINT:PRINT:LOCATE 30,20
2960 PRINT CHR$(7);CHR$(24);"ERREUR DISQ
UETTE !!!";CHR$(24)
2970 FOR I=1 TO 2000 : NEXT I
2980 PRINT ERR;ERL
2990 RESUME 100
3000 REM ***** TESTE LA PRESENCE DE L'
IMPRIMANTE SUR LE PORT D'ADRESSE &F500 (
ON LINE) *****
3010 PRINT:PRINT:PRINT
3020 PRINT CHR$(7);CHR$(24);"IMPRIMANTE
NON CONNECTEE";CHR$(24);" !!!"
3030 PRINT CHR$(7);CHR$(24);" OU PAS
DE PAPIER ";CHR$(24);" !!!"
3040 FOR I = 1 TO 2000:NEXT I
3050 GOTO 100

```

- Lignes 40 et 50 : ces lignes permettent de réserver le buffer de l'unité de disquette.
- Ligne 80 : réserve un tableau possible de 500 logiciels à gérer (si vous n'en avez pas assez, faites donc un nettoyage par le vide des logiciels que vous n'utilisez certainement plus).
- Ligne 90 : création d'une variable de vérification de la sauvegarde d'un fichier logiciel avant le chargement d'un nouveau.
- Lignes 100 à 350 : affichage du MENU et acquisition à la volée du choix.
- Ligne 110 : permet d'appeler un sous-programme de traitement d'erreur sur l'unité de disquette (mauvais format du nom de fichier, fichier inexistant, oubli de disquette...).
- Lignes 360 à 390 : testent la validité du choix et effectuent le branchement.
- Lignes 400 à 470 : affichage du menu LISTE et attente du choix LISTE ENTIERE ou LISTE ALPHANUMERIQUE : pour effectuer un choix par liste alphanumérique, il vous suffira de frapper la ou les premières lettres du ou des logiciels recherchés et le programme gèrera la recherche.
- Lignes 520 à 670 : affichage de la liste complète des logiciels, avec possibilité de sortir de la liste à la fin de l'affichage d'une page (vous vous laisserez guider par les instructions).
- Lignes 680 à 870 : recherche par ordre alphanumérique (les logiciels dont le nom commence par un chiffre sont aussi pris en compte).
- Lignes 880 à 1160 : cette partie de programme vous propose un écran composé de champs que vous remplirez un à un (la validation d'un champ s'effectue par la touche **<RETURN>**). Lorsque vous n'aurez plus de logiciels à insérer, il vous suffira de valider par **<RETURN>** sur le champ DESIGNATION qui restera vide.
- Lignes 1170 à 1630 : permettent la modification d'une caractéristique sur un logiciel. Si vous entrez le caractère ! sur le champ DESIGNATION, le logiciel sera irrémédiablement éliminé du fichier. Par contre lors de la modification d'une caractéristique, le simple fait de frapper **<RETURN>** sur le champ où se trouve le curseur laisse ce champ inchangé. Si vous voulez effectuer une modification, il faut le faire sur tout le champ.
- Lignes 1640 à 1910 : cette portion de programme permet d'effectuer des recherches selon trois caractéristiques :
  - la désignation,
  - le numéro d'ordre,
  - la capacité du logiciel (*exemple* : si vous voulez optimiser les contenus de vos disquettes, et qu'il vous reste 15 K-octets sur une disquette, vous pouvez rechercher tous les logiciels ayant cette taille).

— Lignes 1920 à 2220 : permettent de charger un de vos fichiers logiciels en vous demandant le nom (vous pourrez séparer le fichier utilitaire du fichier jeu, nous ne saurons que trop vous le recommander). Une vérification est auparavant effectuée sur la variable **SURSAUV**, afin de savoir si le logiciel précédemment en mémoire a été modifié, a été sauvegardé. Si ce n'est pas le cas, le logiciel vous demande confirmation (saut à la routine placée en 2190 à 2220) de l'effacement du fichier actuellement en mémoire.

Par contre si le fichier doit être effacé, la place mémoire libérée doit être récupérée pour être affichée (ceci est fait habituellement par le programme et n'a pas besoin d'être géré par le programmeur, sauf dans notre cas car nous affichons la mémoire restant), ce qu'effectue la ligne 2020 par l'instruction **PRINT FRE (" ")**.

— Lignes 2230 à 2390 : sauvegarde du fichier en demandant à l'utilisateur le nom.

— Lignes 2400 à 2510 : possibilité d'imprimer le fichier sur imprimante. De plus, lorsque vous effectuez une sortie sur imprimante, et que celle-ci n'est pas connectée, il est très déconcertant d'attendre, et être obligé d'effectuer un « **Break** ». Le programme teste donc en ligne 2430 si l'imprimante est bien connectée par lecture en ligne 2420 de l'état du port Centronics placé à l'adresse &F500 (si le contenu de ce port est différent de 30 décimal alors l'imprimante ne répond pas présente — bouton « **ON LINE** » non appuyé — imprimante non allumée ou non connectée...). Si tout est correct, l'impression est réalisée en caractères condensés (CHR\$(27);CHR\$(15) ligne 2440), après réinitialisation de l'imprimante (CHR\$(27);CHR\$(64)). En fin d'impression, l'imprimante est de nouveau ré-initialisée pour une utilisation sur un autre logiciel par exemple (ligne 2500).

— Lignes 2520 à 2530 : affichage du catalogue disquette sans avoir à retourner sous Basic.

— Lignes 2540 à 2620 : ce sous-programme permet d'afficher à l'écran les champs de caractéristiques des logiciels utilisés par les choix **CORRECTION** et **AJOUTER** (appels en lignes 930 et 1200).

— Lignes 2630 à 2650 : traitement d'une erreur d'entrée sur un champ de 24 caractères.

— Lignes 2660 à 2680 : traitement d'une erreur d'entrée sur un champ de trois caractères.

— Lignes 2690 et 2700 : affichage de l'en-tête de présentation lors de l'affichage d'une liste de logiciels.

— Lignes 2710 à 2900 : tri des logiciels par ordre alphanumérique sur la désignation. Le principe de ce tri sera expliqué au paragraphe suivant.

— Lignes 2910 et 2920 : annulation sonore d'une entrée non confirmée (appel en ligne 1150).

— Lignes 2930 à 2990 : traitement de l'erreur sur l'unité de disquette par affichage en vidéo-inversée, et retour au menu (par l'instruction **<RESUME 100>**).

— Lignes 3000 à 3050 : traitement de la non-présence de l'imprimante et retour au menu.

*Remarque :*

Vous aurez certainement aperçu l'instruction CALL &BBO6, qui effectue un branchement à un vecteur de la RAM (placé à cette adresse). Ce vecteur envoie le programme à une routine qui attend la frappe d'une touche sur le clavier. Dès que l'on appuie sur une touche, le Basic a de nouveau la main et continue son traitement. Cette routine n'affecte aucune variable du programme et peut être utilisée sans risque dans n'importe quel programme nécessitant un arrêt provisoire, redémarrant par l'appui d'une touche. Vous pouvez essayer cette commande en mode direct, vous verrez alors le curseur disparaître et ne réapparaître que lorsque vous appuierez sur une touche quelconque.

#### **IV - Le principe de tri**

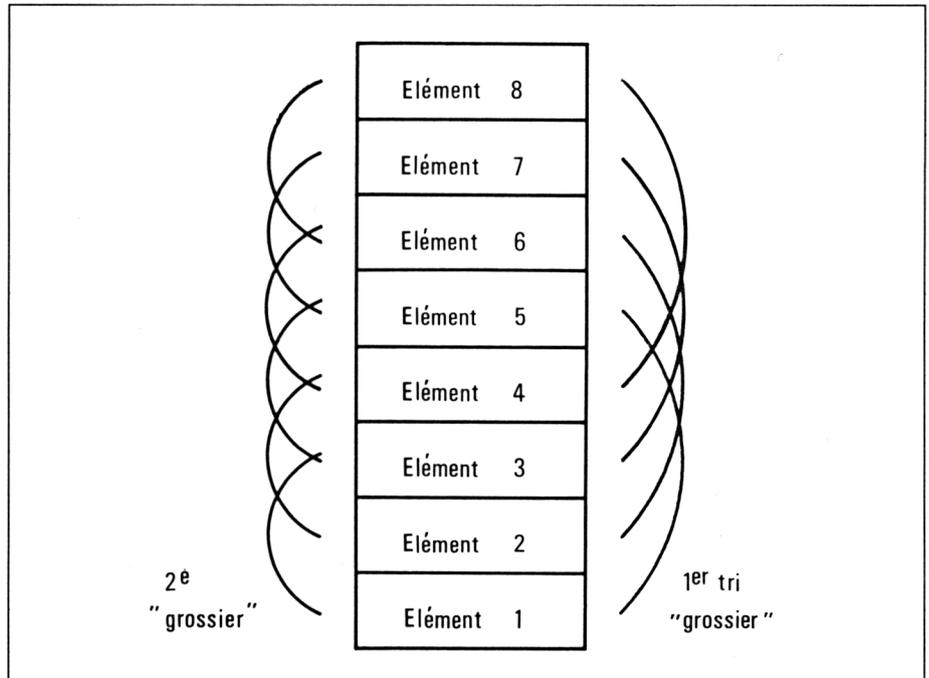
Ce sous-programme est celui du tri par la méthode de Shell améliorée.

Le principe du tri de Shell consiste à diviser une première fois l'ensemble à trier en deux, puis à comparer les éléments distants de la moitié du nombre d'éléments jusqu'à atteindre le dernier élément de la liste, et à les permuter si le cas se présente (selon un tri croissant ou décroissant).

Quand les éléments ont été une première fois triés grossièrement de cette façon, on divise de nouveau le nombre d'éléments et on effectue le tri des éléments distants de ce nouveau nombre. Et ainsi de suite jusqu'à ne plus pouvoir diviser l'ensemble des éléments à trier.

Si le nombre trouvé lors de la division par deux ne tombe pas juste, on prendra la valeur directement inférieure comme distance des éléments à trier.

Prenons exemple sur un ensemble de huit éléments que l'on veut classer par ordre croissant de bas en haut.



Exemple de tri « Shell-Metzner » sur 8 éléments.

Une première division par deux nous donne 4 comme distance entre les éléments. Nous comparerons d'abord l'élément 1 et l'élément 5 ( $1 + 4 = 5$ ). Si l'élément 5 est inférieur à l'élément 1, ils seront permutés. Puis on compare l'élément 2 et l'élément 6, une permutation sera effectuée si nécessaire. On fera de même pour les éléments 3 et 7 ainsi que pour les éléments 4 et 8.

Ensuite la valeur 4 est divisée en deux, ce qui va permettre de trier les éléments distants de 2.

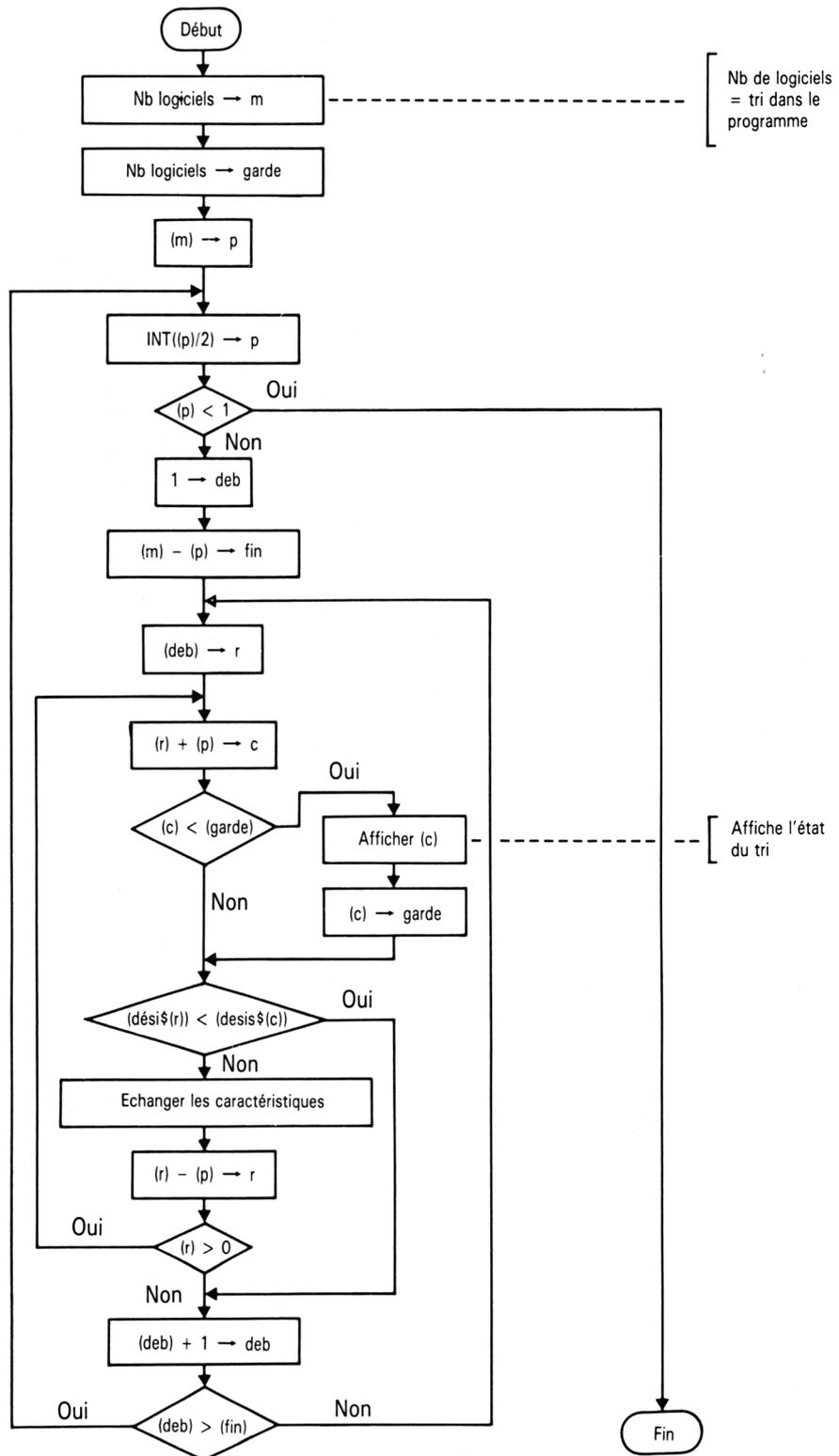
L'élément 1 est comparé avec l'élément 3, permuté si nécessaire, puis l'élément 2 et l'élément 4, ... etc.

Une nouvelle division par deux du précédent écart nous donne la distance UN (dernier écart possible) des éléments à trier.

Ce dernier tri effectué, tous les éléments de l'ensemble seront triés comme il l'était désiré.

Cette méthode de tri est de loin beaucoup plus rapide que la méthode du tri classique dit à *bulle*, que l'on trouvera décrite dans le chapitre concernant les applications du lecteur de disquette Vortex 5 pouces 1/4 (Voir Partie 8, chapitre 6.1).

L'algorithme du programme de tri utilisé dans la gestion de logiciels vous est donné ci-dessous :



Nous vous signalons que la notation (**m**) signifie que l'on utilise le contenu de la variable **m** pour effectuer le traitement.

De plus le signe  $\rightarrow$  est le signe d'affectation, c'est-à-dire, dans l'exemple (**m**) - (**p**)  $\rightarrow$  **fin**, que l'on affecte à la variable **fin** la valeur de la différence des contenus de **m** et de **p**. Cette formulation s'écrit plus simplement en Basic, mais moins rigoureusement, de la façon suivante : **fin = m - p**.

Vous retrouverez le programme transcrivant cet algorithme en Basic entre les lignes 2730 et 2890 (voir pages 14 et 15).

— La ligne 2750 n'a pas d'intérêt majeur dans le tri, ainsi que la ligne 2820 qui permet d'afficher l'état du tri (la distance entre les éléments triés), qui permet de faire patienter l'utilisateur en le rassurant sur un non-ratage du tri.

— Les lignes 2840 à 2870 assurent les permutations des éléments ne répondant pas au critère de comparaison de la ligne 2830.

## V. Modification du programme

Nous vous proposons par exemple de modifier le programme pour qu'il convienne à la gestion de votre discothèque ou de votre bibliothèque.

Votre imagination aidant, vous pourrez élaborer les caractéristiques intéressantes de vos livres ou disques (nom de l'auteur, titre, nombre de pages...), pour remplir la liste décrite au paragraphe sur les caractéristiques d'un logiciel stocké.

Pour modifier le programme, il suffira de remplacer les caractéristiques, et le mot **LOGICIEL** par les noms que vous aurez choisis dans les lignes suivantes :

— 13 - 160 - 220 à 280 - 690 - 920 -  
 — 1190 - 1680 - 1780 - 1840 - 1870 -  
 — 2450 - 2550 à 2600 - 2700 -

