

software MBASIC

Gestione di alberi binari

di Vincenzo Bossi - Torino

Seconda parte

Come promesso nella scorsa puntata, ecco che in questo numero presenteremo un programma applicativo, che sfrutta le subroutine di gestione degli alberi binari.

Il programma in questione, chiamato dall'autore «alberi1», è uno dei tanti esempi di possibile applicazione delle subroutine citate: l'autore confessa che tale programma contiene qualche piccola imprecisione formale, nel senso che ad esempio il secondo File Indice (FI) è a volte chiamato col numero «1» mentre altre con il numero «2»,

STRUTTURA FILE:

Ogni File indice ha la seguente struttura:

Variabili: HFPSx# / HFPDx# / HFFPx# / HFPRx# / HFCHx#
Pos. nel Record: 1 2 / 3 4 / 5 6 / 7 8 / 9 ... 9 + L. Chiave

x e' il Numero di canale su cui e' aperto il File.

(Tutte le variabili HFP.x# contengono variabili NUMERICHE INTERE, ottenute con la funzione MKI\$(). Per avere il valore numerico occorre usare la funzione CVI(HFP.x#))

HFPSx# = Punt. al Figlio Sinistro.
HFPDx# = " " Destro.
HFFPx# = " " Padre.
HFPRx# = " " Record dati.
HFCHx# = Chiave

In ogni File, il primo Record, contiene, nella variabile HFPSx#, il numero di record di cui e' costituito il File (sempre come CVI()).

Tabella 1 - Struttura dei file usati dal programma di gestione di alberi binari.

2 h\$	Chiave Rc. da sost. -Canc.	44 hndc	N. Rc. da cancell. -Canc.
3 h\$(15)	chiave (per trasf. a File)	45 hndel	N. Rc. da annullare -Canc.
4 h(0.hci)	P.Sn. (=HFPSx#)	46 hnfdc	Num. Rc. Figlio Dx. -Canc.
5 h(1.hci)	P.Dx. (=HFPDx#)	47 hnr(0.hci)	num. ult. Rc.
6 h(2.hci)	P.Padre (=HFFPx#)	48 hnr(1.hci)	lung. Ch.
7 h(3.15)	per trasferire dati col File	49 hnr(2.15)	contiene i dati
8 h(3.hci)	P.Rc.Dati (=HFPRx#)	50 hnr(2.hci)	Rc. corrente
9 half\$	Nome F. alfab. -Ott.	51 hnud	N. ultimo Rc. a Dx. -Canc.
10 hca	N.Can. F. alfab. -Ott.	52 hnums	Num. Rc. da sost. -Canc.
11 hcc	registra Num. Rc.	53 hott	O: ott. F.I./1: F.I.+F.D. -Ott
12 hcc\$	Ch. letta	54 hott!	indice 'ottimizz.'
13 hcfdd	N.Can. F.D. dest. -Ott.	55 hp	P. al Rc.
14 hcfds	N.Can. F.D. sorg. -Ott.	56 hpi	P. al Rc. appena letto
15 hch\$	chiave	57 hp2	incremento Rc. -Ott.
16 hcha\$	Chiave in F.alf. -Ott.	58 hpat	Rc. padre
17 hchd\$	chiave da sostituire	59 hpd	P.Dx. Rc. da cancell. -Canc
18 hchi\$	chiave da ins. invece di hchd\$	60 hpdfdc	P.Dx. Figlio Dx. -Canc.
19 hci	Num. Canale	61 hpd\$	P.Dx. Rc. da sost. -Canc.
20 hcid	N.Can. F.I. dest. -Ott.	62 hpp	P. al padre
21 hcis	N.Can. F.I. sorg. -Ott.	63 hppc	P.Padre Rc. da cancell. -Canc.
22 hcon	N. Rc.per controllo	64 hppfdc	P.Padre Figlio Dx. -Canc.
23 hcr	Rc. corrente	65 hpps	P.Padre Rc. da sost. -Canc.
24 hcr1	+ come hcc	66 hprc	P.Dati Rc. da cancell. -Canc.
25 hdc\$	chiave Rc. da cancell. -Canc.	67 hprfdc	P.Dati Figlio Dx. -Canc.
26 hdisk!	K per ott. solo F.I.	68 hprs	P.Dati Rc. da sost. -Canc.
27 hdisco!	k per ott. F.I. e F.D.	69 hpsc	P.Sn. Rc. da cancell. -Canc.
28 hfd1\$	lset x Rc. F.D. sorg. -Ott.	70 hpsfdc	P.Sn. Figlio Dx. -Canc
29 hfd2\$	lset x Rc. F.D. dest. -Ott.	71 hps\$	P.Sn. Rc. da sost. -Canc.
30 hfdc\$	Chiave Figlio Dx. -Canc.	72 hpun	legato a hp2 -Ott.
31 hfd\$	Nome F.D. dest. -Ott.	73 hra	Num. Ult. Rc. Alfab. -Ott.
32 hfddef\$	Nome defin. F.D. -Ott.	74 hrd	Num. Rc. Dati
33 hfd\$	Nome F.D. sorg. -Ott.	75 hrd1	conserva P. Rc. Dati
34 hfi\$	nome F.I.	76 hrd2\$	P.Dati in F.alf. -Ott.
35 hfid\$	Nome F.I. dest. -Ott.	77 hrde	Num. Rc. cancellati
36 hfiddef\$	Nome defin. F.I. -Ott.	78 hro	Num. Rc. occupati
37 hfin	conserva N. ult. Rc. -Ott.	79 hru	Num. Rc. utilizzati
38 hfis\$	Nome F.I. sorg. -Ott.	80 hs	scelta della sub
39 hi	O: P.Sn./1: P.Dx.	81 hsic	0 = salvo sempre/1 = solo fine
40 hic	come hi. dipende da questo	82 ht	0 = non trovato/1 = trovato
41 hlc	Lung. Ch.	83 hur	Num. Ult. Rc.
42 hlfd	lung. Rc. F.D. -Ott.	84 hxy	variabile per loop -Ott.
43 hmd	O: nuovo hrd/1: hrd=hrd1		

Tabella 2 - Indice delle variabili usate dal programma di gestione degli alberi binari.

ma ciò non influenza assolutamente il funzionamento.

Il lettore attento noterà che il listato (vedi pagg. 164, 165) contiene un gran numero di istruzioni di commento, dove al posto di REM, l'autore ha messo l'apostrofo ('): ebbene laddove nella linea di programma ci sono delle assegnazioni ed altre istruzioni eseguibili dall'MBASIC, allora bisogna eliminare tale apostrofo.

Infatti l'autore ha desiderato così evidenziare le istruzioni più importanti, sfruttando set di caratteri alternativi della sua stampante, senz'altro riuscendo nell'intento: dal momento che per fare ciò doveva inserire dei caratteri di controllo che l'MBASIC non avrebbe accettato, allora è stato costretto all'uso di REM, che viceversa accettano all'interno della riga stessa anche caratteri di controllo.

Altre annotazioni

A completamento di quanto detto la scorsa puntata, aggiungiamo tre utili tabelle di riferimento:

- nella tabella 1 troviamo indicata la struttura dei file utilizzati dal programma applicativo e dalle subroutine;

- nella tabella 2 abbiamo invece un elenco delle variabili che compaiono nel programma: tale elenco è stato ottenuto a partire da un altro programma, chiamato «alber2», identico ad «alber1», ma con tutte le REM eliminate (a parte i commenti veri e propri!!) e perciò con tutte le linee rese operative;

- nella tabella 3 infine sono riportate schematicamente le subroutine vere e proprie di gestione degli alberi binari, per ognuna delle quali vengono indicate varie informazioni:

- il nome della routine, indicante la funzione eseguita

- il numero associato alla routine stessa

- eventuali annotazioni sull'uso della routine

- l'elenco delle variabili da settare in input prima della chiamata (riga

contraddistinta da una «freccia in giù»): alcune variabili poste tra parentesi quadra possono essere settate o meno a seconda delle situazioni

— l'elenco delle variabili i cui valori si ottengono in output al termine dell'esecuzione della routine (riga contraddistinta da una «freccia in su»).

CHIAMATE:

INIZIO - 1 -

- Da fare una volta sola all'inizio del programma -
 ↓: HS=1:GOSUB 60000
 ↑: -

APERTURA - 2 -

↓: HCI=:HFI#=:HLC=:HS=2:GOSUB 60000
 ↑: -

INSERIMENTO - 3 -

↓: HCI=:HCH#=:CHSIC=:JHS=3:GOSUB 60000
 ↑: HT

RICERCA - 4 -

↓: HCI=:HCH#=:HS=4:GOSUB 60000
 ↑: HT,HRD,H#(HCI)

CANCELLAZIONE - 5 -

↓: HCI=:HCH#=:HS=5:GOSUB 60000
 ↑: HT,HRD

NEXT - 6 -

↓: HCI=:HI=:HS=6:GOSUB 60000
 ↑: HT,HRD,H#(HCI)

FINE - 7 -

↓: HCI=:HS=7:GOSUB 60000
 ↑: -

MODIFICA - 8 -

↓: HCI=:HCHD#=:HCHI#=:HS=8:GOSUB 60000
 ↑: HT

OTTIMIZZAZIONE - 9 -

↓: HOTT=:HFIS#=:HCIS=:HFID#=:HCID=:HFIDEF#=:HALF#=:HCA=:HLC=:HFDS#=:
 HCFDS#=:HFDD#=:HCFDD#=:HFDDDEF#=:HLFD=:JHS=9:GOSUB 60000

↑: -

- All'uscita dalla subroutine tutti i files sono chiusi -

INIGEN - 10 -

- Da fare una volta sola per ogni file -

↓: HCI=:HFI#=:HLC=:HS=10:GOSUB 60000

↑: -

- All'uscita dalla subroutine il file e' chiuso -

BASALT - 11 -

↓: HCI=:HI=:HS=11:GOSUB 60000

↑: H#(HCI)

- All'uscita dalla Subroutine il valore di HI e' invertito -

STAT - 12 -

↓: HCI=:HLFD=:JHS=12:GOSUB 60000

↑: HRO,HRU,HRDE,HOTT!,HDISK!,HDISKO!

Tabella 3 - Indice delle chiamate alle subroutine di gestione degli alberi binari, con indicati i parametri di input e di output.

```

1 'save "alber1"
10 'ISTRUZIONI RIGUARDANTI LA GESTIONE DEL
LE CHIAVI ("ALBERO")
20 'ISTRUZIONI RIGUARDANTI LA GESTIONE DEI
DATI
10000 REM *** prg ex: ***
10010 AC3=CHR$(27)+"4"+"blank:skip - F:fine"+CHR$(27)+"5"
10020 AC3=CHR$(27)+"4"+"O:ric - 1:dopo - 2:prima - 3:mod - 4:canc - 5:mod ch. -
6: fine"+CHR$(27)+"5"
10100 DATA Inigen, I, Ric. & C., stat, f, fine, ottim.
10200 REM *** inizio generale ***
10210 'HS=1:GOSUB 60000 -inizio
10300 REM *** open ***
10305 CLOSE
10310 'HCI=1:HFIS="ALB1.IND":HLC=6:HS=2:GOS
UB 60000 -apre il primo file indice
10320 'OPEN "R", #2, "ALB1.DAT", IO }
10340 'FIELD 2,10 AS DATIS } -apre il primo file
Dati
10410 'HCI=3:HFIS="ALB2.IND":HLC=15:HS=2:GOS
SUB 60000 -apre il secondo file indice
10420 'OPEN "R", #4, "ALB2.DAT", 30 }
10440 'FIELD 4,30 AS DATIS } -apre il secondo file
Dati
10500 REM *** scelte ***
10510 RESTORE 10100:CLS:FOR X=1 TO 7:READ A$:PRINT X;"-";A$:NEXT X
10520 LOCATE 1,18:INPUT "Scelta":ISC0
10600 ON SCO GOSUB 11000,12000,13000,14000,15000,16000,17000
10700 IF SCO=1 OR SCO=7 THEN 10300
10710 IF SCO=6 THEN 10900
10800 GOTO 10500
10900 END
11000 REM *** inizio generale ***
11100 CLS:PRINT "1- 1":PRINT "2- 2":PRINT "3- 2 & 3":PRINT "4- skip":LOCATE 1,17
:INPUT "Scelta":SC1
11110 ON SC1 GOTO 11200,11300,11400,11999
11200 INPUT "conferma (S):ESC:IF SC1<>"S" AND SC1<>"s" THEN 11000
11210 'HCI=1:HFIS="ALB1.IND":HLC=6:HS=10:
GOSUB 60000 -Inizializza il F. Indice N.1, sul canale 1
11215 CLOSE
11220 RETURN
11300 INPUT "conferma (S):ESC:IF SC1<>"S" AND SC1<>"s" THEN 11000
11310 'HCI=3:HFIS="ALB2.IND":HLC=15:HS=10:
GOSUB 60000 -Inizializza il F. Indice N.2, sul canale 3
11315 CLOSE
11320 RETURN
11400 INPUT "conferma (S):ESC:IF SC1<>"S" AND SC1<>"s" THEN 11000
11410 'HCI=1:HFIS="ALB1.IND":HLC=6:HS=10:
GOSUB 60000 -Inizializza il F. Indice N.1, sul canale 1
11420 'HCI=3:HFIS="ALB2.IND":HLC=15:HS=10:
GOSUB 60000 -Inizializza il F. Indice N.2, sul canale 3
11425 CLOSE
11999 RETURN
12000 REM *** inserimento ***
12100 CLS:PRINT "1- 1":PRINT "2- 2":PRINT "3- 2 & 3":PRINT "4- fine":LOCATE 1,17:
INPUT "Scelta":SC2
12110 CLS:PRINT AC2$
12200 ON SC2 GOTO 12300,12500,12300,12300,12999
12300 REM *** ↓ can. #1 ***
12310 INPUT "chi":CHI$
12320 IF CHI$="" THEN 12380
12330 IF CHI$="f" OR CHI$="F" THEN 12000
12340 'HCI=1:HCHS="f" OR CHI$="F":GOSUB 60000 -inseri-
sco una nuova chiave (HCH$) sul file aperto sul canale HCI (1)
12350 'IF HT THEN PRINT "ESISTE GIA":GOTO 12310 -controllo se esiste gia'
la chiave
12360 'INPUT "data":DI$ }
12365 'LSET DATIS=DI$ }
12370 'PUT #2,HRD } -Inserisco i dati corrispondenti alla chi
ave
12380 IF SC2=3 THEN 12500

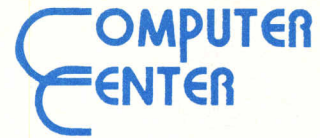
```

3

```

13701 'HS=5:GOSUB 60000 -Cancello la chiave
13720 'IF HCI=1 THEN TRAS19=DAT19:LSET DAT19="*"+T
RAS19:PUT #2,HRD:GOTO 13150 -annullo il Record dati
13730 'IF HCI=1 THEN TRAS29=DAT29:LSET DAT29="*"+T
RAS29:PUT #4,HRD:GOTO 13150 -annullo il Record dati
13799 GOTO 13150
13800 REM *** modifica chiave ***
13801 'HCHD$=CH$ -Chiave da cancellare
13810 INPUT "nuova chiave:NCH$
13820 'HCHIS=NCH$ -Chiave da inserire
13830 'HS=8:GOSUB 60000
13840 GOTO 13150
13999 RETURN
14000 REM *** output ***
14010 DATA raw video,raw stampa,alfa,fine
14090 RESTORE 14010
14200 CLS:LOCATE 1,6:FOR X=1 TO 4:READ A$:PRINT X,A$:NEXT X
14210 LOCATE 1,19:INPUT "Scelta":SCB
14300 ON SCB GOSUB 14400,14400,14600,14999
14310 IF SCB=4 THEN RETURN
14320 GOTO 14000
14400 INPUT "1 o 2":NCON:ON NCON GOTO 14402,14457
14402 'GET #1,1:HCI=1:PRINT CVI(HFPS1$):IF SCB=2
THEN LPRINT CVI(HFPS1$) -Stampa il numero del Record del F.Ind.
14405 FOR X=2 TO HNR(O,1)
14410 'GET #1,X:GET #2,X:GOSUB 63800 -Legge i dati
del F.Ind. e li converte in H(O-1-2-3,HCI) e H(HCI) usando la subroutine 63800
14411 'Attenzione: Ad ogni Chiave non corrispondera' il Record del F.
Dati indicato
14412 'dal puntatore al Record Dati, ma i Record dei 2 File sono semplicemente a
ppaiati.
14420 IF SCB=1 THEN PRINT X;TAB(5);:FOR Y=0 TO 3:PRINT USING "###";H(Y,HCI);:PRI
NT " ";:NEXT Y:PRINT " ";:H(HCI);TAB(45);DAT1$
14430 IF SCB=2 THEN LPRINT X;TAB(5);:FOR Y=0 TO 3:LPRINT USING "###";H(Y,HCI);:L
PRINT " ";:NEXT Y:LPRINT " ";:H(HCI);TAB(45);DAT1$
14450 NEXT X
14455 INPUT "vado":SCF:GOTO 14000
14457 'GET #3,1:HCI=3:PRINT CVI(HFPS3$):IF SCB=2
THEN LPRINT CVI(HFPS1$) -Stampa il numero del Record del F.Ind.
14460 FOR X=2 TO HNR(O,3)
14470 'GET #3,X:GET #4,X:GOSUB 63800 -Legge i dati
del F.Ind. e li converte in H(O-1-2-3,HCI) e H(HCI) usando la subroutine 63800
14480 IF SCB=1 THEN PRINT X;TAB(5);:FOR Y=0 TO 3:PRINT USING "###";H(Y,HCI);:PRI
NT " ";:NEXT Y:PRINT " ";:H(HCI);TAB(45);DAT2$
14490 IF SCB=2 THEN LPRINT X;TAB(5);:FOR Y=0 TO 3:LPRINT USING "###";H(Y,HCI);:L
PRINT " ";:NEXT Y:LPRINT " ";:H(HCI);TAB(45);DAT2$
14495 NEXT X
14496 IF SCB=1 THEN INPUT A
14499 RETURN
14600 REM *** alfa ***
14610 CLS
14620 'INPUT "canale":HCI -scegli canale
14630 IF HCI<>1 AND HCI<>3 THEN 14620
14640 'INPUT "Asc(O) o disc(1)";HI -Ordinamento ascendente o discendente?
14645 IF HI<>0 AND HI<>1 THEN 14640
14650 INPUT "Video(O) o print(1)";VP
14655 IF VP<>0 AND VP<>1 THEN 14650
14670 'HS=11:GOSUB 60000 -Cerca il primo (o l'ultimo) dato
14675 IF HI THEN XX=2 ELSE XX=HNR(O,HCI)
14680 GOSUB 14900 : stampa
14700 IF HI THEN XX=XX+1 ELSE XX=XX-1
14705 'HS=6:GOSUB 60000 -cerca il dato successivo (o precedente)
14710 'IF HT THEN GOSUB 14900 ELSE 14800 -finito?
14720 GOTO 14700
14800 INPUT A
14810 GOTO 14000
14900 REM *** printalfa ***
14905 IF HCI=1 THEN 14950
14910 'GET #4,HRD -Leggo il Record Dati del File N.4
14915 IF VP THEN 14930

```



LE PIÙ GRANDI NOVITÀ DEL 1986
AI PREZZI PIÙ BASSI D'EUROPA

PERIFERICHE PER QL

QINTERAM - Espansione interna di memoria da 256 o da 512 kb con Eprom da 16 kb.

Incrementa la memoria del QL a 384 o 640 kb, si inserisce all'interno del computer senza effettuare nessuna saldatura, dissaldatura o manomissione, non invalida pertanto la garanzia ed il montaggio è velocissimo ed accessibile a tutti. Il connettore di espansione resta libero e si possono inserire contemporaneamente altre periferiche.

Inoltre sulla stessa scheda è presente una Eprom che aggiunge 16 kb di nuovi e potenti comandi al QDOS, fra i quali la gestione del ROM come RAM Disk e il multitasking che permette ad esempio di caricare in memoria i 4 programmi della PSON e passare istantaneamente dall'uno all'altro. Dissipazione termica e assorbimento irrilevanti grazie all'impiego di integrati con tecnologia TTL C-MOS.

QINTERAM 256 kb L. 199.000
QINTERAM 512 kb L. 259.000

QM DISK - Sistema per floppy drives

Il sistema è composto da un'interfaccia per floppy drives e da drives da 3,5" e 1 Mb. L'interfaccia progettata anch'essa dall'autore del QDOS Tony Tebby possiede la piena compatibilità con tutto il software e hardware SINCLAIR.

Drive 3.5" 1 Mb L. 490.000
QINTERCOM - Interfaccia di comunicazione multifunzione.

È munita di una uscita Centronics per stampanti-parallele, più una uscita Cannon a 9 poli per joystick e di una uscita per Modem con possibilità di selezionare la velocità da 75 a 1.200 Baud.

QL CARPROM

Cartuccia porta EPROM completa di EPROM da 128 kb. Si inserisce nella porta ROM esterna.

1 L. 15.000
10 L. 12.000
100 L. 8.900

SOFTWARE PER QL

QL MOON

Superbo e professionale programma sull'astronomia lunare, informazioni e statistiche su oltre 300 punti, banca dati di oltre 50 kb. Gestito da menu con grafica eccezionale. **Su cartuccia** L. 34.000

QL PLAGIO

Professionale copiatore universale per QL. Copia settore per settore qualsiasi tipo di cartuccia. Scritto completamente in L/M. Funziona con qualsiasi tipo di programma e versione ROM. Il programma viene venduto esclusivamente per proprio uso personale.

Su EPROM L. 90.000
Floppy disk 3,5" DD - SD 135 TPI Basf L. 7.000
Floppy disk 3,5 DD - DD - 135 L. 90.000
TPI Panasonic L. 9.000
Coppia integrati versione JS L. 35.000

ATTENZIONE PER TUTTI I POSSESSORI DELLO SPECTRUM

Interfaccia Duplex

Permette di duplicare e di trasferire su: NASTRO - MICRODRIVE - FLOPPY DISK qualsiasi tipo di programma commerciale oggi esistente sul mercato; TURBO - TURBOPULSANTI - MAXI - CON L/M NEL LOADER, ecc.

Semplicissima da usare, si collega l'interfaccia al connettore di espansione, al termine premendo un tasto di break si ottiene una copia a velocità normale che si carica in maniera autonoma senza interfaccia collegata.

I possessori dell'interfaccia 1 potranno scegliere l'opzione microdrive al momento del trasferimento ed ottenere su cartidge una copia del programma preferito.

Il prezzo dell'INTERFACCIA DUPLEX, con il manuale e le spese di spedizione contrassegno è di L. 95.000

ATTENZIONE I PREZZI SOVRAESPONDI SONO IVA COMPRESA

PER INFORMAZIONI E/O ORDINAZIONI

VIA FORZE ARMATE, 260
2052 MILANO
TEL. 02-4890213

4

```

14920 'PRINT XX;TAB(5);H$(3);TAB(30);DAT2$:RETURN -stampo la chiave e
il dato
14930 LPRINT XX;TAB(5);H$(3);TAB(30);DAT2$:RETURN
14950 'GET #2,HRD -Leggo il Record Dati del File N.2
14955 IF VP THEN 14980
14960 'PRINT XX;TAB(5);H$(1);TAB(30);DAT1$:RETURN -stampo la chiave e
il dato
14980 LPRINT XX;TAB(5);H$(1);TAB(30);DAT1$:RETURN
14999 RETURN
15000 REM *** fine ***
15010 'HCI=1;HS=7;GOSUB 60000 -Chiudo Il F.Ind. N.1
15020 'HOI=3;HS=7;GOSUB 60000 -Chiudo Il F.Ind. N.3
15030 CLOSE
15099 RETURN
16000 REM *** stat ***
16100 'INPUT "canale";HCI -sceigo canale
16110 'INPUT "Lunghezza Rc. Dati";HLFD
16120 'HS=12;GOSUB 60000 -Sub. Statistica
16130 'PRINT "cc":HRO;"util";HRU;"canc";HRDE;"ott";HOTT
;HDISK;HDISKO! -Stampo le variabili
16140 INPUT A
16999 RETURN
17000 REM *** ottimizza ***
17010 CLS
17020 'INPUT "solo F.Indice (0) o anche F.dat1 (1)";HOTT
17030 IF HOTT<>0 AND HOTT<>1 THEN 17020
17040 'INPUT "F.Ind. sorgente";HFIS$
17045 'INPUT "sul canale";HCIS
17050 'INPUT "F.Ind. destinazione";HFIDS$
17060 'INPUT "sul canale";HCID
17070 'INPUT "Nome definitivo F.Indice";HFIDEF$
17080 'INPUT "nome file alfabetico";HALF$
17090 'INPUT "sul canale";HCA
17095 'INPUT "Lunghezza chiave";HLC
17100 IF HOTT=0 THEN 17200
17110 'INPUT "F.Dati sorgente";HFDS$
17120 'INPUT "sul canale";HCFDS
17130 'INPUT "F.Dati destinazione";HFDD$
17140 'INPUT "sul canale";HCFDD
17150 'INPUT "Nome definitivo F.Dati";HFDDDEF$
17160 'INPUT "Lunghezza Rc. F.Dati";HLFD
17200 INPUT "conferma (s)";SC7$
17210 IF SC7$<>"s" AND SC7$<>"S" THEN RETURN
17220 'HS=9;GOSUB 60000 -Sub. Ottimizzazione
17230 RETURN
    
```

2

```

12390 GOTO 12310
12500 REM *** ↓ can. #2 ***
12510 INPUT "CH2";CH2$
12520 IF CH2$="" THEN 12580
12530 IF CH2$="f" OR CH2$="F" THEN 12000
12540 'HCI=3;HCH$=CH2$;HS=3;GOSUB 60000 -Inseri-
sco una nuova chiave (HCH$)sul File aperto sul canale HCI (3)
12550 'IF HT THEN PRINT "ESISTE GIA";GOTO 12510 -controllo se esiste gia'
la chiave
12560 'INPUT "data2";D2$ }
12565 'LSET DAT2$=D2$ }
12570 'PUT #4,HRD }
ave
12580 IF SC2=3 THEN 12300
12590 GOTO 12510
12999 RETURN
13000 REM *** ric. & C. ***
13010 CLS:PRINT AC3$
13020 'INPUT "canale";HCI -N. Canale del F.Ind.
13030 IF HCI=6 THEN RETURN
13060 IF HCI<>1 AND HCI<>3
13100 INPUT "chiave";CH$
13110 'HCH$=HCH$;HS=4;GOSUB 60000 -Cerco la chiave
13120 'IF HT=0 THEN PRINT "NON ESISTE";GOTO 13020 -Controllo se esiste la
chiave
13130 'IF HCI=1 THEN GET#2,HRD:PRINT DAT1$ }
13140 'IF HCI=3 THEN GET#4,HRD:PRINT DAT2$ }
13145 '-Stampo i dati
13150 INPUT "indirizzo";SC3
13160 IF CRLIN>15 THEN PRINT AC3$
13200 ON SC3 GOTO 13400,13410,13500,13700,13800,13999
13210 GOTO 13020
13200 REM *** Data successivo o precedente ***
13401 'HI=1;GOTO 13420 -chiave successiva
13410 'HI=0 -chiave precedente
13420 'HS=6;GOSUB 60000 -cerco la chiave successiva (o preceden-
te)
13430 GOTO 13120
12500 REM *** modifica dati ***
13501 INPUT "nuovo dato";ND$
13510 'IF HCI=1 THEN LSET DAT1$=ND$:PUT #2,HRD }
13520 'IF HCI=3 THEN LSET DAT2$=ND$:PUT #4,HRD }
13521 '-Inserisco il nuovo dato
13530 GOTO 13150
13700 REM *** cancellazione ***
    
```