

# software

## C-64

a cura di Tommaso Pantuso

### Music 64

di Paolo Verrocchio - Pescara

Questo programma consente di far suonare semplici melodie al vostro Commodore 64, scrivendole direttamente su un pentagramma disegnato sullo schermo. Il programma ridefinisce alcuni caratteri per rappresentare le note e le pause; i dati corrispondenti alle sagome dei nuovi caratteri occupano 2 Kb a partire dalla locazione 2048. Questo però è solitamente il byte iniziale della RAM Basic, cioè di quella parte di memoria RAM in cui sono memorizzati i programmi Basic.

Per evitare quindi una disastrosa sovrapposizione tra dati e programma bisogna spostare la RAM Basic, facendola partire alla locazione 4096. In sostanza prima di iniziare a battere il programma dovrete digitare  
POKE 44,16:POKE 4096,0  
e ogni volta che vorrete ricaricare il programma battete  
POKE 44,16:POKE 4096,0:LOAD «MUSIC»

### Impiego del programma

Dato il RUN si dovrà attendere qualche istante, il tempo necessario per leggere i dati relativi ai nuovi ca-

ratteri, allo Sprite e alle note da suonare. Subito dopo apparirà un pentagramma dall'estensione di 2 ottave sul quale si può muovere uno Sprite di forma quadrata tramite i tasti-cursore (quelli con le frecce, per intenderci) sia in senso verticale, per scegliere la nota, sia in senso orizzontale, per passare ad un'altra nota. Sopra il pentagramma vi è un rigo in cui è indicato che valore avrà la nota o la pausa che si sta per scrivere; naturalmente questo valore può essere modificato come vedremo in seguito. Oltre a queste informazioni vi è indicato anche il numero della pagina su cui si sta lavorando. Le pagine disponibili sono 30, numerate da 0 a 29, ognuna contenente 13 note o pause, per cui la melodia non può essere più lunga di 390 note. Sotto il pentagramma troviamo un primo menu che comprende le seguenti opzioni:

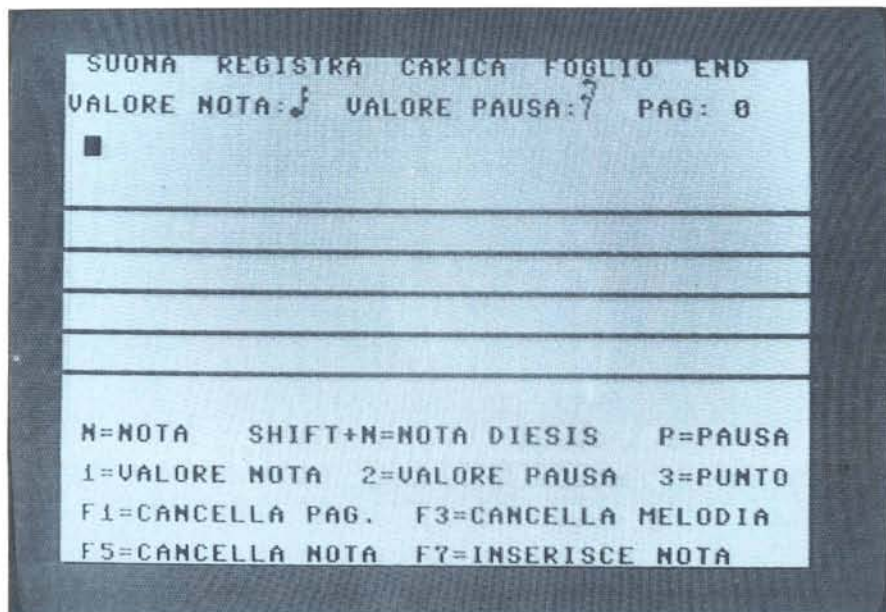
**NOTA:** si abilita premendo «N» e si ottiene la nota specificata dalla posizione del cursore, questa nota avrà come valore quello specificato nel rigo sovrastante il pentagramma.

**NOTA DIESIS:** si abilita con Shift + «N», è uguale all'opzione precedente, solo che la nota sarà diesis, cioè aumentata di un semitono.

**PAUSA:** abilitata tramite il tasto «P» serve per ottenere una pausa del valore specificato nel rigo sovrastante il pentagramma.

**VALORE NOTA:** si ottiene col tasto «1», incrementa il valore della nota indicato nell'ormai famoso rigo delle informazioni.

**VALORE PAUSA:** ottenuta tramite il tasto «2», funziona come la prece-



dente ma invece del valore della nota incrementa il valore della pausa.

**PUNTO:** si abilita premendo «3» e aggiunge ai valori della nota e della pausa un punto; cioè il valore di una nota o di una pausa viene aumentato della metà del valore stesso.

(Ad esempio una nota del valore di 2/4 seguita da un punto varrà 3/4).

**CANCELLA PAGINA:** con questa opzione, ottenuta tramite F1, si cancella la pagina su cui si sta lavorando.

**CANCELLA MELODIA:** questa opzione invece, ottenibile con F3, serve per cancellare l'intera melodia.

**CANCELLA NOTA:** si abilita premendo F5 e serve per cancellare la nota o la pausa su cui si trova il cursore.

**INSERISCE NOTA:** si abilita con F7 e si usa per aggiungere delle note o delle pause di cui ci si è dimenticati in un primo momento. Grazie a questa opzione si crea uno spazio nel punto in cui è posizionato il cursore, e tutte le note successive scalano di un posto.

Sopra il pentagramma e sopra il rigo delle informazioni troviamo un altro menu composto dalle seguenti opzioni:

**SUONA:** si abilita con «S» e dopo aver fatto questa scelta ci apparirà un altro menu formato dalle seguenti vo-

ci: 1) STRUMENTI 2) METRONOMO 3) ESECUZIONE 4) FINE.

Premendo 1 si sceglie con quale strumento suonare la melodia. Gli strumenti disponibili sono 6: pianoforte, flauto, violino, cembalo, xilofono, organo.

Premendo 2 si sceglie la velocità con cui suonare la melodia: essa varia da 1 a 9 e per default è uguale a 4.

Premendo 3 si passa all'esecuzione vera e propria. Il computer ci rivolgerà alcune domande per sapere quante sono le pagine su cui sono scritte le note (o le pause) da suonare e qual è la prima; immessi questi dati esso suonerà la melodia. L'esecuzione è ciclica, nel senso che dopo aver finito, il computer ricomincia da capo; per fermarla basta premere lo spazio.

premeendo 4 si ritornerà al pentagramma.

**REGISTRA:** si abilita con «R» e serve per registrare la melodia su un registratore a cassette.

**CARICA:** si abilita con «C» e serve per caricare una melodia precedentemente salvata sul registratore.

**FOGLIO:** si abilita con «F» e si usa per passare ad un'altra pagina.

**END:** abilitata con «E» serve per uscire dal programma.

### Per chi vuole il listato

Il listato di questo programma è lungo alcuni metri. In conseguenza di ciò si è ritenuto opportuno non pubblicarlo, sia perché avrebbe occupato troppo spazio sulla rivista sottraendone ad altri argomenti, sia perché una digitazione senza errori di un listato così lungo appare poco probabile.

Chi è interessato al programma può ordinare, secondo il solito sistema, il disco o la cassetta in redazione. È anche possibile «pescare» direttamente (e gratuitamente) il programma per via telematica, dal nostro servizio MC-Link; questo ovviamente vale per chi è attrezzato in tal senso. Ricordiamo che per ottenere una casella su MC-Link è sufficiente telefonare (con un modem e un programma di comunicazione) al numero 06/4510211.

Questo programma è disponibile su disco presso la redazione. Vedere l'elenco dei programmi disponibili e le istruzioni per l'acquisto a pag. 169.

## String Search

di Bert Bunse - Quartu Sant'Elena

String Search è un programma per il C-64 corredato di disk drive e la sua funzione è quella di cercare una sequenza di stringhe o codici ASCII a scelta dell'utente, sul disco, fornendo come risultato finale il numero di traccia e settore nel quale o nei quali essi sono registrati.

La caratteristica principale è che, per eseguire tale ricerca, il programma non trasferisce tutti i dati dal disco alla memoria del computer (il che richiederebbe un tempo notevole sia in Basic come pure in L.M.), ma lavora mediante una routine in L.M. situata nel buffer del drive per cui l'esame di tutti i settori del disco, compresi quelli situati sulla traccia 18, richiede un tempo medio di circa 3 minuti.

Esso può quindi essere utilizzato per le numerose applicazioni che richiedono il ritrovamento sul disco di un testo o di una serie di comandi di un linguaggio di programmazione (per esempio «jsr ffd5») evitando le noiose ricerche «manuali» con un disk monitor.

Passo ora ad una descrizione più dettagliata del programma, rimandando al commento allegato ai listati per ulteriori informazioni. Le note principali del programma sono:

— ricerca in tutti i settori del disco di al massimo sei stringhe fino alla lunghezza di 10 caratteri ciascuna o di al massimo sei sequenze di codici ASCII fino ad una lunghezza di 10 codici per ciascuna sequenza;

— l'esame di tutto il disco richiede un tempo di circa 3 minuti che può però allungarsi se il disco contiene molte stringhe corrispondenti a quelle cercate in quanto l'output del risultato allunga i tempi;

— come risultato finale vengono indicati: 1) il numero di traccia e settore nel quale è stata trovata una determinata stringa; 2) un numero tra 0 e 255 (esso viene fornito sia in decimale che in esadecimale) che punta l'ultimo byte della stringa trovata nell'ambito del settore, la quale può essere quindi facilmente reperita mediante un disk monitor.

Il programma consiste in due parti: una in Basic che carica la parte in L.M., controlla l'input dei parametri di ricerca definiti dall'utente ed infine avvia e guida la ricerca stessa; una in L.M., situata nel buffer numero 2 del drive, che esamina il contenuto di ciascun settore caricato nel buffer numero 1 da parte del programma Basic e scrive il risultato della ricerca nel buffer numero 0 dove viene controllato dal programma Basic. In caso di esito positivo sarà trasferito nella memoria

del computer. In questo modo si evita infatti il trasferimento di tutti i dati dal drive al computer attraverso il lento bus seriale.

Lo svolgimento del programma comprende:

— il caricamento nel buffer del drive della parte in L.M.;

— l'utente decide poi se cercare delle stringhe o sequenze di codici ASCII e quante di esse devono essere ricercate in ciascun settore (da 1 a 6);

— segue l'input delle stringhe o codici ASCII controllato da parte del programma;

— ad input concluso i parametri di ricerca sono trasferiti nel buffer 0 del drive dove vengono utilizzati dalla routine in L.M.;

— inizia la ricerca vera e propria; se in un settore vengono trovate delle stringhe corrispondenti a quelle cercate, il risultato viene indicato sul video;

— a ricerca conclusa, sarà chiesto all'utente se esaminare un altro disco, nel qual caso la ricerca sarà ripetuta con gli stessi parametri di prima senza la necessità di ripetere l'input.

Vorrei infine segnalare che secondo il mio parere il programma contiene una serie di routine piuttosto interessanti per quanto riguarda il controllo sull'input che, per essere scritte in basic, sono abbastanza efficaci.

**Lista variabili**

**SS, B%, HS, H** = variabili per la trasformazione di numeri decimali in numeri esadecimali; H contiene il numero decimale al momento dell'ingresso nella rispettiva subroutine, HS il risultato finale

**CS** = decide se vengono cercate stringhe o sequenze ASCII

**NS, N** = numero delle stringhe - codici ASCII cercate (da 1 a S)

**ST\$(x) o ST\$(x,10)** = stringhe - sequenze ASCII cercate

**LS(x)** = lunghezza delle sequenze di codici ASCII (1-10)

**P** = posiziona il cursore nell'input delle stringhe - codici ASCII

**ST\$** = controllo sull'input

**T1, T2, S** = numeri di traccia e settore

**AS** = risultato della lettura di una locazione di memoria del buffer del drive

**L, L1** = posizionano il buffer-pointer

**A** = numero delle stringhe trovate in un de-

terminato settore (max. 110)

**NOS** = nome della routine in L.M.

**X\$** = input s/n, return

**v, w, x, y, z** = variabili per cicli for - next

**Commento al listato Basic**

**90 - 160** = controllo sulle decisioni si/no dell'utente; 120 accende il cursore e svuota

il buffer della tastiera; 130, 140 input; 150 stampa N = no sul video; 160 spegne il cursore

**460 - 700** = cerca nella directory del disco la routine in L.M. che serve per eseguire la ricerca delle stringhe nei singoli settori e la carica nel buffer #2 (\$0500 - 05ff) del drive

**710 - 1130** = input delle stringhe o sequenze ASCII da cercare

*Disassemblato del programma in L.M.*

```

B#
PC SR AC KR VR SP
.;C037 70 37 00 4B F6
.
5004 EA          NOP
5005 EA          NOP
5006 A9 00      LDA #000
5008 BD 40 03   STA $0340
500B BD 01 03   STA $0301
500E AE 01 03   LDX $0301
5011 EE 01 03   INC $0301
5014 EC 00 03   CPX $0300
5017 D0 01     BNE $501A
5019 60        RTS
501A BD 10 03   LDA $0310,X
501D BD 20 03   STA $0320
5020 BD 16 03   LDA $0316,X
5023 8D 21 03   STA $0321
5026 A2 00     LDX #000
5028 AC 20 03   LDY $0320
502B BD 00 04   LDA $0400,X
502E D9 00 03   CMP $0300,Y
5031 F0 06     BEQ $5039
5033 E8        INX
5034 D0 F5     BNE $502B
5036 4C 0E 05   JMP $050E
5039 BE 30 03   STX $0330
503C CC 21 03   CPY $0321
503F F0 1A     BEQ $505B
5041 E0 FF     CPX #0FF
5043 F0 C9     BEQ $500E
5045 E8        INX
5046 CB        INY
5047 BD 00 04   LDA $0400,X
504A D9 00 03   CMP $0300,Y
504D F0 ED     BEQ $503C
504F EE 30 03   INC $0330
5052 AE 30 03   LDX $0330
5055 AC 20 03   LDY $0320
5058 4C 2B 05   JMP $052B
505B AC 40 03   LDY $0340
505E C0 6E     CPY #06E
5060 F0 B7     BEQ $5019
5062 EE 40 03   INC $0340
5065 0A        TXA
5066 99 90 03   STA $0390,Y
5069 E0 FF     CPX #0FF
506B F0 A1     BEQ $500E
506D 4C 4F 05   JMP $054F
5070 EA          NOP
    
```

```

1 REM *****
2 REM *
3 REM *   S T R I N G   S E A R C H   *
4 REM *
5 REM *           (C) 1986           *
6 REM *
7 REM *           BERT BUNSE         *
8 REM *
9 REM *****
10 POKE53280,6:POKE53281,0:GOSUB280
11 DIMHS,Cs,Ns,ST$,As,Nos,X$,B%,H,N,P,L,L1,A,T2,V,W,X,Y,Z
12 S$="0123456789ABCDEF":T1=18:S=1
13 DATA1,17,20,18,24,18,25,30,17,31,35,16
14 GOTO430
15 REM *****
16 REM * SUBROUTINES *
17 REM *****
18 REM *
19 REM * INPUT S/N
20 REM *
21 POKE204,0:POKE198,0
22 GETX$:IFX$<>"N"ANDX$<>"S"ANDX$<>CHR$(13)GOTO130
23 IFX$=CHR$(13)THENX$="S"
24 IFX$="N"THENPOKE1024+PEEK(214)*40+PEEK(211),142
25 POKE204,1:RETURN
26 REM *
27 REM * LETTURA BUFFER DRIVE #2
28 REM *
29 GET#5,As:IFAs=""THENAs=CHR$(0)
30 RETURN
31 REM *
32 REM * INPUT RETURN
33 REM *
34 POKE198,0
35 GETX$:IFX$<>CHR$(13)GOTO260
36 RETURN
37 REM *
38 REM * INTESTAZIONE VIDEO
39 REM *
40 PRINT"(CLR)(PUR)":FORX=1TO40:PRINT"-":NEXT
41 PRINTSPC(12)"(WHT)(RVS) STRING SEARCH (PUR)"
42 FORX=1TO40:PRINT"-":NEXT
43 PRINTSPC(10)"(YEL)(DOWN)(C) BERT BUNSE 1986 (CYN)":RETURN
44 REM *
45 REM * CONVERSIONE DEC.-ESA.
46 REM *
47 B%=0:H$=""
48 FORV=1TOOSTEP-1
49 H=H-B%*16+(V+1):B%=H/16+V:H$=H$+MID$(S$,B%+1,1)
50 NEXTV
51 RETURN
52 REM *****
53 REM * INIZIO PRG PRINCIPALE *
54 REM *****
55 REM *
56 REM * CARICAMENTO ROUTINE L.M.
57 REM *
58 PRINTSPC(5)"(DOWN)(DOWN)INSERISCI IL DISCO ORIGINALE E"
59 PRINTSPC(13)"(DOWN)PREMI 'RETURN':GOSUB220
60 OPEN15,8,15:OPEN5,8,5,"#2"
61 PRINT#15,"U1";5:0:T1;S
62 PRINT#15,"B-P";5:0
63 GOSUB200:T1=ASC(As)
64 GOSUB200:S=ASC(As)
65 FORX=0TO7
66 PRINT#15,"B-P";5:X*32+5
67 NOS=""
68 FORY=0TO15
69 GOSUB200:IFASC(As)=160GOTO630
70 NOS=NOS+As
71 NEXTY
72 NEXTX
73 IFNOS="ROUTINE L.M."GOTO660
74 NEXTX
75 GOTO520
76 PRINT#15,"B-P";5:X*32+3
77 GOSUB200:T1=ASC(As)
78 GOSUB200:S=ASC(As)
79 PRINT#15,"U1";5:0:T1;S
80 CLOSE15:CLOSE5
81 REM *
82 REM * INPUT STRINGHE
83 REM *
84 PRINTSPC(6)"(DOWN)(DOWN)VUOI CERCARE UNA SEQUENZA DI:"
85 PRINTSPC(8)"(DOWN)(RVS) A (OFF) ASCII-CODES"
86 PRINTSPC(8)"(DOWN)(RVS) C (OFF) CARATTERI DA TASTIERA"
87 GETC$:IFC$<>"C"ANDC$<>"A"GOTO770
88 PRINT"(DOWN)(DOWN)(RGHT)NUMERO DELLE STRINGHE CERCATE (MAX.6)?"
    
```

**850, 940** = aprono la tastiera come file logico; «input #1» = «input» senza punto interrogativo

**880, 990, 1110** = posiziona il cursore nella desiderata riga e colonna del video

**1000 - 1040** = controllo sull'input delle sequenze di codici ASCII; devono essere compresi tra 0 e 255; il primo deve essere <> X (cioè di ciascuna sequenza scelta si

deve immettere almeno un codice); non possono contenere altro che cifre o X per indicarne la fine

**1060 - 1090** = idem

**1140 - 1430** = inizializza il buffer #0 (\$0300 - 03ff) del drive; in esso vengono infatti trasferiti tutti i parametri di ricerca (numero di stringhe cercate, la loro lunghezza ecc.) che servono poi alla routine in L.M., situata

nel buffer #2, per la ricerca nei singoli settori (che del resto verranno caricati nel buffer #1 (\$0400 - 04ff))

**1210 - 1220** = carica il numero delle stringhe o sequenze ASCII cercate nella posizione 0 (\$0300)

**1230 - 1240** = carica l'inizio delle stringhe (80 (\$0350), 90, 100 ecc.) in 16 (\$0310), 17, 18 ecc.

**1260 - 1330** = carica la fine delle stringhe (dipendente dalla loro lunghezza) in 22 (\$0316), 23 24 ecc. ed infine le stringhe stesse nelle rispettive posizioni (per esempio: prima stringa = \$0350, \$0351, \$0352 -< fine stringa)

**1340 - 1430** = idem per le sequenze ASCII

**1440 - 1568** = ricerca delle stringhe - sequenze ASCII nei singoli settori ed output

**1480 - 1490** = caricano settore per settore nel buffer #1 (\$0400 - 04ff) ed eseguono la ricerca chiamando la routine L.M. che inizia in \$0506 (= U5)

**1500** = legge il risultato della ricerca nell'ultimo settore che viene depositato nel buffer #0, locazione 64 (\$0340); essa contiene infatti il numero delle stringhe trovate in quel settore (max 110); se 0, viene caricato direttamente il prossimo settore

**1510 - 1540** = carica la posizione dell'ultimo byte nell'ambito del settore per ciascuna stringa trovata che viene depositata a partire da 144 (\$0390); output su video (traccia, settore, posizione ultimo byte) per ciascuna stringa.

```

790 GETN$:IFN$<"1"ORN$>"6"GOTO790
800 N=VAL(N$):IFC$="A"GOTO930
810 REM * INPUT CARATTERI DA TASTIERA
820 DIMST$(N):GOSUB280:PRINT
830 FORX=1TON:PRINTSPC(6)"(DOWN)INPUT"X"(LEFT).STRINGA:"NEXT
840 PRINTSPC(5)"(DOWN)(MAX. 10 CARATTERI A STRINGA)"
850 OPEN1,0
860 P=6:POKE647,4:PRINT"(PUR)"
870 FORX=1TON:P=P+2
880 POKE211,24:POKE214,P:SYS58640:INPUT#1,ST$(X):IFLEN(ST$(X))>10GOTO880
890 NEXT
900 PRINT:PRINTSPC(15)"(DOWN)(DOWN)(DOWN)(CYN)VA BENE ? S(UP)":POKE211,25:GOSUB9
910 IFX$="N"GOTO860
920 CLOSE1:GOTO1140
930 REM * INPUT ASCII-CODES
940 DIMST$(N,10):LS(N)=OPEN1,0
950 FORY=1TON:GOSUB280:PRINTSPC(9)"(DOWN)(DOWN)IMMETTI LA"Y"(LEFT).SEQUENZA:"
960 FORZ=OTOSTEP2:PRINT"(DOWN)(RIGHT)(RIGHT)ASCII CODE"Z":ASCII CODE"Z"
1"":"NEXTZ
970 PRINTSPC(9)"(DOWN)(X - PER FINE SEQUENZA)"
980 P=6:POKE647,4:PRINT"(PUR)":FORZ=1TO9STEP2:P=P+2
990 POKE211,17:POKE214,P:SYS58640:INPUT#1,ST$(Y,Z)
1000 IFVAL(ST$(Y,Z))<ORVAL(ST$(Y,Z))>255GOTO990
1010 IFLEFT$(ST$(Y,Z),1)="-"X"GOTO990
1020 ST$=LEFT$(ST$(Y,Z),1)
1030 IFST$<"0"ORST$>"9"ANDST$<"X"GOTO990
1040 IFST$="X"GOTO1110
1050 POKE211,35:INPUT#1,ST$(Y,Z+1)
1060 IFVAL(ST$(Y,Z+1))<ORVAL(ST$(Y,Z+1))>255GOTO1050
1070 ST$=LEFT$(ST$(Y,Z+1),1)
1080 IFST$<"0"ORST$>"9"ANDST$<"X"GOTO1050
1090 IFST$="X"GOTO1110
1100 NEXTZ
1110 POKE211,15:POKE214,22:SYS58640:PRINT"(CYN)VA BENE ? S(UP)":POKE211,25:GOSUB
90
1120 IFX$="N"GOTO980
1130 NEXTY:CLOSE1
1140 REM *
1150 REM * INIZIALIZZ. BUFFER DRIVE #0
1160 REM *
1170 PRINT:PRINT"(UP)(UP)(UP)(RIGHT)(RIGHT)INSERISCI IL DISCO SUL QUALE LAVORARE"
1180 PRINTSPC(13)"(DOWN)E PREMI 'RETURN'":GOSUB220
1190 PRINT"(CLR)TRACCIA: SETTORE: BYTE NUMERO:(PUR)"
1200 OPEN15,8,15:OPEN3,8,3,"#0":OPEN4,8,4,"#1"
1210 PRINT#15,"B-P":3;0
1220 PRINT#3,CHR$(N)
1230 PRINT#15,"B-P":3;16
1240 L=70:FORX=1TON:L=L+10:PRINT#3,CHR$(L):NEXT
1250 IFC$="A"GOTO1340
1260 L=70:PRINT#15,"B-P":3;22
1270 FORX=1TON
1280 L=L+10:L1=L+LEN(ST$(X))-1:PRINT#3,CHR$(L1)
1290 NEXT
1300 L=70:FORX=1TON:L=L+10
1310 PRINT#15,"B-P":3;L
1320 PRINT#3,ST$(X)
1330 NEXTX:GOTO1440
1340 L=70:PRINT#15,"B-P":3;22
1350 FORX=1TON:FORY=1TO10:IFLEFT$(ST$(X,Y),1)="-"X"THENLS(X)-Y-1:GOTO1380
1360 NEXTY
1370 LS(X)-10
1380 L=L+10:L1=L+LS(X)-1:PRINT#3,CHR$(L1)
1390 NEXTX
1400 L=70:FORX=1TON:L=L+10
1410 PRINT#15,"B-P":3;L
1420 FORY=1TOLS(X):PRINT#3,CHR$(VAL(ST$(X,Y)))
1430 NEXTY:NEXTX
1440 REM *
1450 REM * RICERCA STRINGHE ED OUTPUT
1460 REM *
1470 FORZ=1TO4:READT1,T2,S
1480 FORX=1TOT2:FORY=OTOS
1490 PRINT#15,"U1":4;0;X;Y:PRINT#15,"U5":PRINT#15,"B-P":3;64
1500 GET#3,AS:IFAS$="GOTO1550
1510 A=ASC(AS):PRINT#15,"B-P":3;144
1520 FORW=OTOA-1:GET#3,AS:H=ASC(AS):GOSUB350
1530 PRINT"(DOWN)"XTAB(12)YTAB(25)ASC(AS)TAB(30)" - $H$
1540 NEXTW
1550 NEXTY:NEXTX:NEXTZ
1560 CLOSE3:CLOSE4:CLOSE15
1570 PRINTSPC(7)"(DOWN)(CYN)ESAMINI UN ALTRO DISCO ? S(UP)":POKE211,32:GOSUB90
1580 IFX$="N"GOTO1610
1590 PRINT:PRINT"(DOWN)(RIGHT)(RIGHT)INSERISCI IL DISCO SUL QUALE LAVORARE"
1600 PRINTSPC(12)"(DOWN)E PREMI 'RETURN'":GOSUB220:RESTORE:GOTO1190
1610 PRINT"(CLR)"

```

### Commento al listato L.M.

Poiché la routine lavora nel buffer #2 del drive (\$0500 - 05ff), è stata assemblata nella memoria del computer a partire da \$5006. Una volta copiato il listato, esse deve essere salvata sul disco che contiene la parte Basic con il nome di «ROUTINE L.M.» mediante un monitor o Assembler, salvando da \$5004 (!) - \$5070 (proprio per poterla far partire da \$0506).

Essa lavora usando i parametri di ricerca che sono stati trasferiti dalla parte Basic in \$0300 - \$0390 del drive e deposita il risultato della ricerca nel singolo settore in \$0340 (numero delle stringhe - sequenze ASCII trovate in quel settore) e da \$0390 in poi (posizione dell'ultimo byte della stringa - sequenza ASCII trovata nel settore). Le locazioni di memoria nel buffer #0, importanti per la comprensione del suo funzionamento, sono le seguenti:

**\$0300** = numero di stringhe - sequenze ASCII ricercate in ciascun settore

**\$0310 - \$0315** = dà la posizione dell'inizio delle stringhe - sequenze ASCII da 1 a 6

**\$0316 - \$031b** = dà la posizione della fine delle stringhe sequenze ASCII da 1 a 6

**\$0340** = numero delle stringhe - sequenze ASCII trovate in quel settore (max. 110)

**\$0350** = inizio area stringhe - sequenze ASCII

**\$0390** in poi = posizione ultimo byte della prima (seconda, terza ecc.) stringa - sequenza ASCII trovata in quel settore.