Flib

di Roberto Boccato
S. Donà di Piave (VE)

Il programma che Vi propongo e che gira nel Commodore 64 mi è stato suggerito dalla lettura dell'articolo di A.K. Dewdney «Esplorando algoritmi genetici in un mare primordiale pieno di flib» apparso nella rubrica (Ri) creazioni al calcolatore del numero di gennaio 86 di Le Scienze.

In questo programma viene simulata l'esistenza di particolari organismi detti flib (da «finite living blobs» o bolle viventi finite), immersi in un ambiente digitale chiamato dall'autore brodo primordiale, con evidente riferimento alle condizioni esistenti negli oceani terrestri qualche miliardo di anni fa in coincidenza con l'apparizione sul pianeta delle prime strutture vitali.

In realtà la primitività dei flib è notevole, in quanto la loro vita trascorre quasi unicamente con lo scopo di rispondere ad una varietà molto limitata di stimoli ambientali che influiscono sull'organismo a seconda del particolare stato in cui esso si trova in un certo istante.

Supponiamo che gli stimoli ambientali si riducano a due e vengano introdotti nel brodo primordiale in una sequenza determinata. Supponiamo anche che i flib possano esistere solo in 4 stati diversi (non importa conoscere i particolari di questi stati). Potremo allora rendere il calcolatore in grado di controllare questa situazione adoperando una serie di 0 e 1 per rappresentare gli stimoli ambientali (ogni cifra corrisponde ad uno stimolo) e le lettere a, b, c, d, per rappresentare gli stati possibili dei flib.

Ogni flib reagisce ad uno stimolo ambientale in un modo che dipende dal suo stato: percepito lo stimolo, il flib emette un segnale (anche qui uno 0 o un 1) in risposta, e cambia il suo stato in un altro tra quelli disponibili.

Naturalmente i flib sono diversi l'uno dall'altro, per lo meno all'inizio.

Potremmo schematizzare l'esistenza di un flib con una tabella:

	0		1	
a	0	b	1	c
b	1	a	0	d
c	1	a	1	b
d	0	c	0	a

Da questa tabella si ricava che se un flib si trova ad esempio nello stato c e in quel momento lo stimolo ambientale è 1 allora il flib emette un 1 e passa nello stato b. Oppure se il flib si trova nello stato b e lo stimolo ambientale è uno 0, esso emette un 1 e passa allo stato a.

Saranno avvantaggiati nel gioco della sopravvivenza quei flib che riusciranno in qualche modo a prevedere lo stimolo ambientale successivo, si saranno cioè adattati all'ambiente, che per semplicità supponiamo possa variare ma solo in modo ciclico con periodo da 4 a 10. Il flib avrà previsto l'evolversi ambientale se il segnale da lui emesso in risposta allo stimolo precedente coinciderà con lo stimolo ambientale che arriverà subito dopo.

Questo programma è disponibile su cassette presso la redazione. Vedere l'elenco dei programmi disponibili e le istruzioni per l'acquisto a pag. 161.

Ad esempio, facendo riferimento al flib il cui comportamento è rappresentato nella tabella precedente, se lo stimolo attuale è uno 0 e il flib si trova nello stato c, esso emetterà un 1 in risposta e cambierà stato. Se lo stimolo ambientale successivo sarà proprio un 1 vorrà dire che il flib lo avrà previsto.

Sottoponiamo ad una serie di 100 stimoli ambientali, che si ripetono in modo ciclico, una popolazione di 10 flib. Per ogni stimolo successivo previsto da un flib gli assegneremo 1 punto. Ovviamente esisteranno nella popolazione flib che riescono a prevedere meglio degli altri la successione di sti-

Nota

I codici di controllo nei listati sono riportati in forma «esplicita», in conseguenza dell'impiego della stampante Star NL10 e relativa interfaccia per Commodore. Ovviamente, nella digitazione del programma è necessario usare i consueti tasti che corrispondono alle indicazioni fra parentesi: ad esempio cursore destro per (RGHT), CTRL-3 per (RED) eccetera.

(CLR)	=	(YEL)	=
(HOME)	=	(RVS)	=
(DOWN)	=	(OFF)	=
(UP)	=	(ORNG)	=
(RGHT)	=	(BRN)	=
(LEFT)	=	(LRED)	=
(BLK)	=	(GRY1)	=
(WHT)	=	(GRY2)	=
(RED)	=	(LGRN)	=
(CYN)	=	(LBLU)	=
(PUR)	=	(GRY3)	=
(GRN)	=	(SWLC)	=
(BLU)	=		

```
10 REM   ***************************:1383
20 REM   *                         *:1143
30 REM   *         F L I B         *:1310
40 REM   *                         *:1163
50 REM   *    DI ROBERTO BOCCATO   *:1850
60 REM   *                         *:1183
70 REM   ***************************:1443
80 REM:223
90 REM:233
100 GE=1:PRINT"(CLR)":PRINT"(WHT)(UP)":POKE53
280,0:POKE53281,0:PRINTCHR$(14)"(UP)":3246
110 PRINT"(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN
)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)":PRINTSPC(3):PRINT"SEQ
UENZA":1988
120 PRINT"(UP)(UP)":PRINTSPC(14):PRINT"GEN.":
1693
130 PRINT"(UP)(UP)":PRINTSPC(20):PRINT"STIMOL
O":1987
140 PRINT"(UP)(UP)":PRINTSPC(30):PRINT"P.TOT.
":PRINT"":2164
150 FORNF=1TO10:PRINT"FLIB N."NF:NEXT:1971
160 FORI=0TO9:FORJ=0TO14STEP2:1780
170 LO=1075+J+40*L:LU=55347+J+40*L:2893
180 U=INT(RND(1)*2)+48:POKELO+40*I,U:POKELU+4
0*I,1:POKELU+40*I+4,1:4708
190 S=INT(RND(1)*4)+1:POKELO+40*I+1,S:POKELU+
40*I+1,1:POKELU+40*I+4,1:5095
200 NEXTJ,I:521
210 FORLS=1093TO1453STEP40:1519
220 SF=INT(RND(1)*4)+1:1573
230 POKELS,SF:737
240 NEXTLS:529
250 FORPL=55371TO55731STEP40:POKEPL,1:NEXTPL:
2470
260 FORLP=1099TO1459STEP40:1323
270 POKELP,48:NEXTLP:818
280 PRINT:PRINT:PRINT"↖NTRODUCI LA SEQUENZA D
I STIMOLI":3060
290 PRINT"AMBIENTALI (MIN.4 - MAX.10):1869
300 PRINT"(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN
)"SPC(1):INPUTSA$:POKE1865,32:1565
310 PE=LEN(SA$):IFPE<4ORPE>10THENPRINT"(UP)(U
P)(UP)(UP)(UP)(UP)(UP)":GOTO300:3908
320 FORCO=1TOPE:880
330 IFMID$(SA$,CO,1)="0"ORMID$(SA$,CO,1)="1"T
HENNEXTCO:GOTO350:3269
340 PRINT"(UP)(UP)(UP)(UP)(UP)(UP)(UP)(UP)":G
OTO300:1808
350 PRINT"(UP)(UP)(UP)(UP)(UP)(UP)(UP)(UP)(UP
)┌RA STO  SOTTOPONENDO  I 10 FLIB AD UNA":426
5
360 PRINT"SERIE DI 100  STIMOLI AMBIENTALI AV
ENTI":2906
370 PRINT"PER BASE LA SEQUENZA CHE MI HAI FOR
NITO":2948
380 FORDE=1TOPE:931
390 POKE678+DE,ASC(MID$(SA$,DE,1)):1822
400 NEXTDE:412
410 REM:298
420 REM:308
430 REM      ***  MODULO  1 ***:1435
440 REM:328
450 REM:338
460 A=1:POKE252,A:968
470 B=1 :PRINTCHR$(19):FORQT=0TO9:PU(QT)=0:NE
XTQT:WW=0:3442
480 PRINT"(HOME)":PRINTSPC(21):PRINT"(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN
)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)      ":1761
490 PRINT"(HOME)":FORV=0TO9:PRINTSPC(34):PRIN
T"      ":PRINTSPC(34)"(UP)"PU(QT):NEXTV:PRINT
"":3872
500 PRINT"(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN
)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)  ":PRINTSPC
(13)GE:PRINTCHR$(19):2076
510 S1=PEEK(679):1006
520 IFA>PETHENA=1:999
530 PRINT"(HOME)":PRINTSPC(21):PRINT"(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN
)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)"B:PRINT"(HOME)":1760
540 SS=PEEK(678+A):S2=PEEK(679+A):2094
550 FORQT=0TO9:781
560 C=PEEK(1093+40*QT):C1=1093+40*QT:2546
570 US=4*(C-1)+2*(SS-48):MS=PEEK(1075+US+40*Q
T):M1=PEEK(1076+US+40*QT):5525
580 POKEC1,M1:507
590 IFA=PEANDMS=S1THENPU(QT)=PU(QT)+1:PRINTSP
C(34)PU(QT):WW=WW+1:NEXTQT:GOTO630:5078
600 IFMS=S2THENPU(QT)=PU(QT)+1:PRINTSPC(34)PU
(QT):WW=WW+1:NEXTQT:GOTO630:4522
610 PRINT:253
620 NEXTQT:405
630 B=B+1:IFB=101THEN700:1554
640 A=A+1:PRINTCHR$(19):GOTO520:1600
650 REM:283
660 REM:293
670 REM      ***  MODULO  2 ***:1421
680 REM:313
690 REM:323
700 PRINT"(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN
)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)
          ";:2374
710 PRINT"
                                        ":2565
720 FORT=1TO500:NEXTT:1235
730 PRINT"(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN
)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)●TO ORDINANDO I FLIB IN BASE AL        ":
3106
740 PRINT"(UP)PUNTEGGIO REALIZZATO DURANTE IL
 TEST.    ";:3388
750 PRINT"─RA QUALCHE ISTANTE ORDINERO' ANCHE
 LA  ";:3281
760 PRINT"TABELLA SUL VIDEO.":1701
770 D=100:N=-1:930
780 FORQT=0TO9:IFPU(QT)=DTHENN=N+1:GOSUB820:2
683
790 NEXTQT:320
800 IFN=9 THEN860:844
810 D=D-1:GOTO780:933
820 FORLO=0TO18:834
830 POKE50000+LO+40*N,PEEK(1075+LO+40*QT):276
2
840 POKE50020+40*N,PU(QT):1448
850 NEXTLO:RETURN:570
860 PRINT"(HOME)":PRINTSPC(30):PRINT"(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN
)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)          ":1695
870 PRINT"(HOME)":PRINTSPC(30):PRINT"(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN
)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)"WW:1655
880 FORI=0TO9:FORJ=0TO18:1520
890 POKE1075+J+40*I,PEEK(50000+J+40*I):2563
900 NEXTJ:NEXTI:600
910 PRINTCHR$(19):684
920 FORI=0TO9:PRINTSPC(34)"      ":PRINT"(UP)"S
PC(34)PEEK(50020+40*I):3292
930 NEXTI:368
940 XS=0:572
950 SM=INT(RND(1)*4)+1:IFSM=1THENXS=1:2694
```

(continua a pagina 150)

(segue da pagina 149)

```
990 IFSM=1THEN1060:1117
1000 GOTO1400:569
1010 REM:388
1020 REM:398
1030 REM      ***  MODULO  3 ***:1240
1040 REM:163
1050 REM:173
1060 PRINT"(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(DOW
N)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(
DOWN)
            ";:2224
1070 PRINT"
     ";:1578
1080 PRINT"
            ":1785
1090 FORT=1TO1000:NEXTT:1139
1100 RC=INT(RND(1)*8)+1:1433
1110 R1=INT(RND(1)*6)+1:R2=INT(RND(1)*6)+3:28
17
1120 PRINT"(UP)(UP)(UP)(UP)ÌRA IL FLIB N. 1 S
I ACCOPPIA CON IL":3223
1130 PRINT"(UP)FLIB N."RC+1". ÙL FLIB IBRIDO
RISULTANTE    ";:3464
1140 PRINT"SOSTITUISCE QUELLO CHE NEL TEST PR
ECE-":2993
1150 PRINT"(UP)DENTE HA OTTENUTO IL MINOR PUN
TEGGIO.":3077
1160 FORIB=1TOR1:930
1170 POKE50999+IB,PEEK(1074+IB):POKE1074+IB,P
EEK(1074+IB)+128:3593
1180 NEXTIB:429
1190 FORIB=R1+1TOR1+R2:1563
1200 POKE50999+IB,PEEK(1074+40*RC+IB):POKE107
4+40*RC+IB,PEEK(1074+40*RC+IB)+128:5396
1210 NEXTIB:459
1220 FORIB=R1+R2+1TO16:1565
1230 POKE50999+IB,PEEK(1074+IB):POKE1074+IB,P
EEK(1074+IB)+128:3653
1240 NEXTIB:489
1250 FORJ=0TO15:925
1260 POKE1435+J,PEEK(51000+J)+128:1974
1270 NEXTJ:454
1280 PRINT"(HOME)":PRINT"(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)"SPC(35):PR
INT"    ":1363
1290 FORT=1TO2000:NEXTT:1085
1300 FORI=0TO9:FORJ=0TO15:1427
1310 LL=1075+J+40*I:1329
1320 IFPEEK(LL)>127THENPOKELL,PEEK(LL)-128:22
09
1330 NEXTJ:NEXTI:520
1340 FORT=1TO1000:NEXTT:1134
1350 REM:218
1360 REM:228
1370 REM      ***  MODULO  4 ***:1326
1380 REM:248
1390 REM:258
1400 PRINT"(HOME)":PRINT"(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN
)                                        ":207
9
1410 PRINT"
                                        ";:2559
1420 PRINT"
                                        ";:2569
1430 PRINT"
                                        ":2520
1440 FORT=1TO500:NEXTT:1190
1450 PRINT"(UP)(UP)(UP)(UP)(UP)(UP)¯' IN ARRI
VO UN RAGGIO COSMICO":3409
1460 PRINT"(UP)CHE INDURRA' UNA MUTAZIONE NEL
 CROMOSOMA":3330
1470 PRINT"(UP)(UP)DI UN FLIB. ÙL GENE MUTATO
 LAMPEGGERA'":3335
1480 PRINT"(UP)PER QUALCHE ISTANTE.":1963
1490 J=INT(RND(1)*16):I=INT(RND(1)*10):V=0:28
48
1500 POKE1075+J+40*I,PEEK(1075+J+40*I)+128:29
48
1510 FORT=1TO200:NEXTT:IFV=10THEN1540:2184
1520 POKE1075+J+40*I,PEEK(1075+J+40*I)-128:V=
V+1:3596
1530 FORT=1TO200:NEXTT:GOTO1500:1670
1540 PP=PEEK(1075+J+40*I):1587
1550 PRINT"(HOME)":PRINT"(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN
)(DOWN)
        ";:2274
1560 PRINT"
                                        ";:2454
1570 PRINT"                                  ":1125
1580 IFPP=176THENPOKE1075+J+40*I,49:FORT=1TO1
000:NEXTT:PRINT"(HOME)(UP)":GE=GE+1:GOTO470:4
775
1590 IFPP=177THENPOKE1075+J+40*I,48:FORT=1TO1
000:NEXTT:PRINT"(HOME)(UP)":GE=GE+1:GOTO470:4
785
1600 AA=INT(RND(1)*4)+1:1400
1610 IFAA=PP-128THEN1600:1379
1620 POKE1075+J+40*I,AA:1379
1630 FORT=1TO1000:NEXTT:GE=GE+1:PRINT"(HOME)(
UP)":GOTO470:2697
1640 GOTO1640:450
```

moli: sono quelli che totalizzeranno i punteggi più alti.

Ogni flib contiene un cromosoma. Niente paura! non è altro che una stringa di caratteri (0,1,a,b,c,d) che si ottiene ritagliando la tabella vista sopra lungo le righe ed unendone i capi. Il cromosoma del flib descritto nella tabella è quindi così strutturato:

0	b	1	c	1	a	0	d	1	a	1	b	0	c	0	a

Come ogni organismo che si rispetti il flib si riproduce. Non ci interessa la modalità di riproduzione, ma solo il prodotto di essa che è un altro flib il cui cromosoma è ibrido tra quello dei genitori. Si verifica cioè quello che in genetica si chiama crossing-over: un pezzo del cromosoma è uguale a quello paterno, il pezzo successivo a quello materno, poi è di nuovo uguale al paterno, ecc.

La simulazione ha lo scopo di portare una popolazione di flib in condizione di massimo adattamento all'ambiente. Allora è necessario che alla riproduzione partecipi sempre il flib che ha ottenuto il miglior punteggio nel precedente ciclo di test con l'ambiente. Il partner può essere uno qualsiasi degli altri flib «inferiori», tranne l'ultimo. Il prodotto del loro accoppiamento deve sempre sostituire il flib con il punteggio inferiore che in questo modo si estingue essendo non sufficiente il suo grado di adattamento.

Se le cose sembrano così troppo semplici, ecco intervenire un raggio cosmico a mutare il cromosoma di qualche flib. Il raggio colpisce a caso il cromosoma, quindi può colpire un gene che specifica uno stato oppure quello che specifica una certa risposta ad uno stimolo ambientale. Nel primo caso il carattere viene sostituito con uno dei tre rimasti scelti a caso, nel secondo caso il carattere 1 è sostituito da uno zero, oppure lo 0 è sostituito da un 1.

Il programma è suddiviso in 5 blocchi di cui il primo è dedicato all'inizializzazione, cioè alla determinazione dei flib partecipanti alla simulazione e alla richiesta della serie fondamentale di stimoli (una serie di 0 e 1 min 4 - max 10) che poi saranno proposti in numero di 100 alla popolazione.

I restanti 4 moduli sono ciclici.

Nel primo avviene il test dei 10 flib

con i 100 stimoli ambientali e viene assegnato il punteggio ad ognuno.

Il secondo modulo identifica il flib con il punteggio più alto e quello con il punteggio più basso.

Il terzo modulo provvede ad incrociare il flib con punteggio più alto con un altro a caso e a sostituire il flib con punteggio più basso con il prodotto di tale incrocio.

Nel quarto modulo è simulato l'arrivo di un raggio cosmico che colpisce il cromosoma di un flib e provoca una mutazione.

Poi si ritorna al modulo 1.

Il passaggio dal secondo al terzo modulo non è automatico, nel senso che non sempre il flib più adatto all'ambiente si riproduce.

Questo è, per sommi capi, il funzionamento del programma.

È scritto totalmente in basic e quindi è piuttosto lento.

La scelta del basic è stata dettata però dalla convinzione che, come dice lo stesso autore dall'articolo, ci sia ancora molta sperimentazione da fare su questo tipo di programmi in cui compaiono algoritmi genetici. Ho quindi preferito non introdurre routine in linguaggio macchina che renderebbero meno leggibile il programma.

L'uso della stampante è consigliato in quanto rende possibile il controllo delle generazioni passate che sono ormai scomparse dal video.

Le 5 colonne visualizzano nell'ordine:

1 - la generazione
2 - il punteggio totale della popolazione nella generazione considerata (è un indice dell'adattamento della popolazione all'ambiente)
3 - il punteggio del flib più adattato all'ambiente
4 - il punteggio del flib meno adattato all'ambiente
5 - l'avvenuta (1) o non avvenuta (0) riproduzione dei flib nella generazione considerata.

Le righe da aggiungere al programma volendo usare la stampante, sono:

```
960 open4,4
970 print#4,ge;ww;peek(50020);
    peek(50380);xs
980 close4,4
```

Consiglio vivamente agli interessati di procurarsi una copia della rivista che riporta l'articolo. In esso sono descritti più ampiamente i particolari della simulazione.

Analisi del listato

100-140	inizializzazione.
150-270	visualizza la popolazione di flib, il loro stato attuale, il loro punteggio attuale.
280-330	richiesta sequenza stimoli ambientali.
380-400	memorizza la sequenza.
460-640	applica una serie di 100 stimoli alla popolazione di flib.
770-930	riordina la posizione dei flib in base al loro punteggio decrescente.
940-950	scelta prossimo modulo (riproduzione o mutazione).
960-980	eventuali istruzioni per la stampante.
1100-1110	scelta punti di cross-over.
1160-1270	evidenzia i flib che si incrociano e il cross-over dei cromosomi rispettivi.
1490-1620	genera il raggio cosmico che fa mutare un gene di un flib a caso.

Elenco delle variabili usate nel programma

A	=	puntatore lettura sequenza
AA	=	mutazione stato
B	=	contatore stimoli ambientali
C	=	stato del flib sotto controllo
CO	=	controllo caratteri sequenza
D	=	valore decrescente per controllo punteggio
DE	=	locazioni deposito della sequenza stimoli
GE	=	numero generazione
I	=	numero cromosoma (riga)
IB	=	gene cromosoma ibrido
J	=	gene cromosoma
LL	=	riconoscimento caratteri in reverse
LO	=	locazione griglia
LP	=	locazione punteggio
LS	=	locazione stato flib
LU	=	locazione colore griglia
N	=	numero riga della matrice memorizzata dalla loc. 50000
NF	=	numero flib
PE	=	periodo sequenza stimoli
PL	=	locazione colore punteggio
PP	=	contenuto locazione mutato dal raggio
PU(QT)	=	punteggio flib
QT	=	ciclo stimolazione flib
R1	=	punto di cross-over
R2	=	punto di cross-over
RC	=	flib che si accoppia con il numero I
S	=	stato flib nel cromosoma
SA$	=	sequenza stimoli ambientali
SF	=	stato del flib
SM	=	numero casuale per scelta modulo successivo
SS	=	stimolo ambientale attuale
S1	=	prima posizione in memoria della sequenza stimoli
S2	=	posizioni successive nella sequenza
US	=	risposta del flib a uno stimolo
WW	=	punteggio totale popolazione attuale
XS	=	riconoscimento per la stampante dell'avvenuta riproduzione

Note per la copiatura dei listati per il 64

Nel numero 44 (settembre 85) è stato pubblicato un programma di Checksum per aiutare i lettori nella copiatura dei listati per il Commodore 64 pubblicati sulla rivista.

Il funzionamento è il seguente:

— copiate il programma Checksum del numero 44 e salvatelo su disco o cassetta;

— per la successiva copiatura di un listato (con Checksum), caricate (dal vostro disco o dal vostro nastro) il programma di Checksum e fatelo partire; a questo punto potete copiare le varie linee del listato, compresi i due punti ed il numero che trovate alla fine di ciascuna riga. Alla pressione del return, se la linea è stata copiata bene si può passare a copiare la successiva, altrimenti il programma di Checksum vi lascerà "inchiodati" sulla linea mal copiata obbligandovi a correggere l'errore prima di proseguire.

A quanto detto nel numero 44 riguardo al programma Checksum in questione, aggiungiamo che la routine di Checksum in LM si avvia con SYS 52480 mentre, in caso di arresto con Run-Stop/Restore, il restart si effettua con SYS 53072.

Attenzione: chi non vuole usare il Checksum, NON DEVE copiare i due punti e il numero alla fine delle righe, pena la segnalazione di "syntax error" da parte del computer.

Locatore

di Vincenzo Schena - Fasano (BR)

Certamente a tutti gli utilizzatori di un CBM 64 sarà capitato di non ricordare la locazione da porre dopo la SYS per far partire un programma in linguaggio macchina: sia esso un gioco che una utility o altro. L'ideale sarebbe, sicuramente, quello di poter caricare questi programmi con un semplice LOAD e avviarli con un RUN.

Lo scopo del programma Locatore è quello, di trasformare un programma scritto in codice macchina posto in una qualsiasi parte della RAM accessibile all'utente, in uno caricabile ed eseguibile come un normale programma Basic. Per consentire questa variazione, il programma, dopo la traslazione nella RAM Basic, viene fornito di una piccola routine in L.M. che serve a riposizionarlo nella sua naturale sede dopo il RUN e ad avviarlo.

Locatore è utilizzabile sia per programmi su nastro che su disco.

Vediamone le modalità d'uso.

Modalità d'uso

Per poter utilizzare Locatore, bisogna caricare in memoria un breve pro-

```
1 REM ************************:1142
2 REM **  L O C A T O R E  **:1298
3 REM **                    **:954
4 REM **         BY         **:1046
5 REM **                    **:956
6 REM **  VINCENZO SCHENA   **:1563
7 REM ************************:1148
8 ::66
10 CK=0:PRINT"(CLR)(WHT)(DOWN)(DOWN)(RGHT)(RGHT)(RVS)STO CARICAND
O LA PRIMA PARTE DI DATA(OFF)":3468
20 FORI=682TO714:READA:POKEI,A:POKE53281,A:CK=CK+A:NEXTI:POKE5328
1,15:3736
30 IFCK<>3476THENPRINT"ERRORE NEL PRIMO GRUPPO DI DATA!":END:3623
40 CK=0:PRINT"TUTTO O.K.":PRINT"(DOWN)(RGHT)(RGHT)(RVS)STO CARICA
NDO LA SECONDA PARTE DI DATA(OFF)":4434
50 POKE53280,1:FORI=40960TO41187:READA:POKEI,A:POKE53280,A:CK=CK+
A:NEXTI:3915
60 IFCK<>25741THENPRINT"ERRORE NEL SECONDO GRUPPO DI DATA!":END:3
832
70 PRINT"TUTTO O.K.!":FORI=0TO1000:NEXTI:POKE53280,15:2735
80 PRINT"(CLR)(BLK)"CHR$(14)"(DOWN)(DOWN)(DOWN)(DOWN)(RGHT)(RGHT)
xUOI LE ISTRUZIONI ? (S/N)":3043
90 GETA$:IFA$="N"THENPRINT"(CLR)":END:1695
100 IFA$<>"S"THEN90:1119
110 PRINT"(CLR)(DOWN)     ——— STRUZIONI PER L'USO ———":3616
120 PRINT"(DOWN)     ER USARE IL PROGRAMMA BASTA":2547
130 PRINT"    INSERIRE IL DISCHETTO O":PRINT"    LA CASSETTA CONT
ENENTE IL":4239
135 PRINT"    PROGRAMMA DA TRASFORMARE,":PRINT"    E DIGITARE:":3
392
140 PRINT"(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RVS)● ●LOCAZIONE,I
NIZ"CHR$(34)"NOME PROG"CHR$(34):3695
145 PRINT"(DOWN)(RGHT)(RGHT)DOVE:":801
150 PRINT"(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(
RGHT)(RGHT)(RGHT)r":PRINT"(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RG
HT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)| 686 => PER IL NASTRO":3041
155 PRINT"(RGHT)(RGHT)LOCAZIONE=+":1350
160 PRINT"(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(
RGHT)(RGHT)(RGHT)| 682 => PER IL DISCO":PRINT"(RGHT)(RGHT)(RGHT)(
RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)└":2943
165 PRINT"(DOWN)(RGHT)(RGHT)INIT = SYS D'INIZIO DEL PROGRAMMA
        IN L.M.":3553
170 PRINT"(DOWN)(RGHT)(RGHT)NOME = NOME DEL PROGRAMMA IN L.M.":NE
W:2832
280 REM ***INIZIO DATA *************:1684
1000 DATA169,8,208,2,169,1,133,2,32,253,174:2047
1005 DATA32,138,173,32,247,183,72,169,54,133,1:2207
1010 DATA104,32,0,160,169,55,133,1,76,134,227:2147
1050 REM:173
1060 REM **** SECONDE DATA ********:1610
1070 REM:193
2000 DATA160,255,200,185,156,160,153,0,8,192,69:2228
2005 DATA208,245,165,20,141,70,8,165,21,141,71:2172
2010 DATA8,32,121,0,201,34,208,14,160,0,200:1999
2015 DATA177,122,201,34,208,249,136,192,17,48,2:2244
2020 DATA160,0,230,122,152,166,122,164,123,32,189:2329
2025 DATA255,166,2,224,1,208,17,32,110,160,173:2186
2030 DATA61,3,141,43,8,173,62,3,141,47,8:1902
2035 DATA24,96,32,147,160,32,192,255,162,5,32:2162
2040 DATA198,255,32,207,255,141,43,8,32,207,255:2273
2045 DATA141,47,8,32,204,255,169,5,32,195,255:2180
2050 DATA32,147,160,169,0,162,72,160,8,32,213:1913
2055 DATA255,134,45,132,46,160,255,200,177,187,153:2180
2060 DATA14,8,196,183,208,246,169,41,153,14,8:1945
2065 DATA32,231,255,96,169,5,166,2,160,0,76:1842
2070 DATA186,255,0,31,8,193,7,158,50,48,56:1806
2075 DATA50,58,162,58,40,0,0,0,0,0,0:1466
2080 DATA0,0,0,0,0,0,0,0,0,0,0:1139
2085 DATA0,0,0,169,72,133,251,169,8,133,252:1848
2090 DATA169,0,133,253,169,192,133,254,160,0,177:2114
2095 DATA251,145,253,200,208,249,230,252,230,254,165:2307
2100 DATA252,197,46,208,239,76,0,0,0:1528
```

gramma Basic il quale lo loca nella stessa tramite frasi DATA e dà le necessarie istruzioni per un suo corretto uso. Dopo questa facile procedura, Locatore è pronto ad essere usato nel seguente modo:

1) assicuratevi di aver inserito nel DRIVE il dischetto o di aver posizionato all'inizio il programma su nastro che intendete trasformare;

2) se utilizzate il DRIVE digitate quanto segue:

```
SYS682,ST«NOME»
```

dove «ST» deve essere la locazione d'inizializzazione del vostro programma in linguaggio macchina (il valore che avreste usato con l'SYS per intenderci). Mentre NOME è, ovviamente, il nome del programma da trasformare. Notate bene che quest'ultimo deve essere esattamente quello del programma, altrimenti Locatore non funzionerà né col nastro e tantomeno col dischetto;

3) se usate il registratore sostituite a 682 il valore 686.

Una volta terminata l'esecuzione di Locatore, cioè quando riappare il cursore sul video, provate a dare LIST: a questo punto apparirà la seguente linea:

```
1985 SYS2082:NEW:(nome)
```

dove nome è lo stesso del precedente. Quindi, a tal punto, salvate il programma così trasformato con un semplice SAVE e potrete ricaricarlo con un normale LOAD.

Descrizione tecnica di Locatore

Locatore è un programma interamente scritto in linguaggio macchina composto da due parti: una allocata a partire da 682 ($02AA) fino a 715 ($02CB), l'altra da 40960 ($A000) fino a 41186 ($A0E4).

La prima parte, costituita da poche istruzioni, è situata in una zona tra le prime quattro pagine di memoria non usata né dall'interprete Basic tantomeno dal Sistema Operativo (ricordiamo che una pagina di memoria è costituita da 256 bytes). Questa imposta il numero di periferica (registratore o drive); legge la locazione di avvio del programma da trasformare tramite delle routine del Basic; disabilita e, in seguito, riabilita la R.O.M. del Basic. La seconda parte del programma, che è quella principale, è posta sotto la R.O.M. Basic. La scelta di questa particolare zona della R.A.M. consente di tenere il programma in memoria senza disturbare eventuali routines di espansione (come l'ADP BASIC) poste nelle usuali zone di memoria usate a tale scopo. È interessante soffermarsi su

come sia possibile usare questa particolare R.A.M.

Bisogna sapere che il microprocessore 6510 (quello del CBM 64) nonostante sia compatibile con il 6502 (quello del VIC 20), presenta una fondamentale differenza: ha in più rispetto al 6502, una così detta porta di Input/Output. Essa non è altro che un registro che serve a selezionare, cioè a mettere in contatto con il microprocessore, dei circuiti di memoria anziché altri contenuti nel CBM 64. Questo in particolare si ottiene settando (cioè mettendo ad 1) o disattivando (cioè mettendo a 0) gli opportuni bits di questo registro. Infatti, nonostante il CBM 64 possieda 64K di memoria R.A.M. indirizzabile, ha in più delle R.O.M. contenenti l'Interprete Basic, il Sistema Operativo, gli Input/Output e il Generatore di caratteri. Cioè ha un quantitativo di memoria maggiore di 64K. Il problema viene risolto come segue: all'accensione del computer, viene depositato nella locazione 1 della memoria (cioè quella corrispondente al sopra citato registro) il valore 55 ($37) il quale fa in modo che il microprocessore «veda» le R.O.M. suddette e solo una parte di memoria R.A.M. (quella che usiamo per il Basic più alcuni altri Kbytes). Quindi in particolare se sostituiamo al 55 il valore 54 ($36), otteniamo la disattivazione dell'Interprete Basic. Di contro si aumenta la memoria R.A.M. disponibile di 8 Kbytes; questo è quanto viene fatto nella prima parte del programma in L.M.

Analizziamo, ora, il funzionamento della seconda parte del programma. Essa comincia con una procedura che ricopia nella memoria Basic a partire da 2048 ($0800) la linea che sarà visualizzata quando si impartirà il LIST, più un piccolo caricatore che collocherà, dopo il RUN, nella opportuna zona R.A.M., il programma così trasformato. Segue una seconda procedura che legge il nome del programma ricavandone gli opportuni parametri; quindi una terza che seleziona la periferica prescelta. Infine è eseguito il caricamento del programma da manipolare accodandolo al caricatore prima citato e leggendo la locazione da cui esso sarà riposizionato in memoria dopo il RUN. Il tutto si conclude con una breve routine che ricopia il nome nella linea Basic d'inizializzazione tra parentesi e quindi si ritorna nella prima parte di Locatore.

Concludo citando le procedure, implementate in R.O.M., che vengono usate dal programma:

$AEFD: serve a saltare la virgola nel SYS che darete;

$AD8A: trasforma un numero ASCII in un valore interpretabile dal calcolatore;

$B7F7: trasforma un valore in virgola mobile in intero ponendo il byte basso nella locazione 20 ($14) e il byte alto nella locazione 21 ($15);

$FFBD: prepara i parametri del nome quando si dà un SAVE, LOAD ecc...;

$FFC0: è la OPEN del Kernal;

$FFC6: predispone un canale opportunamente aperto, come ingresso;

$FFCF: mette un byte inviato da una periferica, nell'accumulatore;

$FFCC: chiude tutti i canali;

$FFC3: chiude un file logico;

$FFD5: LOAD del Kernal;

$FFBA: predispone i parametri di un file;

$FFE7: chiude tutti i files aperti;

$E386: rimanda il controllo al Basic.

```
10 CO=0:SC=53280:1026
20 POKE SC,CO:543
30 POKE SC+1,CO:772
40 PRINT"(CLR)(GRN)";TAB(4)"(RVS) PROGRAMMA DI FRANCESCO DA VILLA
 (OFF)":3165
50 PRINT"(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(RGHT)QUESTO PROGRAM
MA CONTIENA UNA ROUTINE ":3092
60 PRINT TAB(7);"(DOWN)CHE PERMETTE DI OTTENERE :":2411
70 PRINT TAB(4)"(DOWN)(DOWN)(RVS)[ F1 ](OFF) -> LIST + (RETURN)":
2299
80 PRINT TAB(4)"(DOWN)(DOWN)(RVS)[ F7 ](OFF) -> PROGRAMMABILE":22
93
85 PRINT"(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(R
GHT)(RGHT)(RGHT)(RGHT)(RGHT)DEFAULT = RUN + (RETURN)":2284
90 N=679:512
100 READ A:IF A>=0THENPOKEN,A:N=N+1:GOTO100:2485
110 POKE204,0:PRINT"(DOWN)(DOWN)(DOWN)(DOWN)INPUT COMMAND?  ";:19
79
120 GETA$:IF A$=""THEN120:1272
125 IF A$<>CHR$(13)THENB$=B$+A$:PRINTA$;:GOTO120:2666
127 IF B$=""THEN200:959
130 N=737:X=LEN(B$):FORP=1TOX:1995
135 POKEN+P,ASC(MID$(B$,P,1)):NEXT:1727
140 POKE N+P,0:743
150 DATA 120,169,180,141,20,3,169,2,141,21,3,88,96,165,215,201,13
3,240,7,201:3599
160 DATA 136,240,7,76,49,234,162,0,240,2,162,5:2144
170 DATA 189,221,2,240,6,32,210,255,232,208,245,169,13,141,119,2,
169,1,133,198:3735
180 DATA 208,225,76,73,83,84,0,82,85,78,,0,-1:2124
200 SYS 679:556
```

Definizione di due tasti funzione

di Francesco Da Villa - Venezia

Il programma che vi propongo consente di assegnare un comando fisso ai tasti funzione F1 e F7 del Commodore 64.

Al tasto F1 è assegnato il comando LIST, al tasto F7 il comando RUN. Tuttavia quest'ultimo è assegnato come default in quanto può essere ridefinito dall'utente.

Il programma Basic che alloca la routine in linguaggio macchina a partire dalla locazione 679, richiede ad un certo punto se si desidera mutare il comando RUN con qualsiasi altra funzione o gruppo di funzioni.

Questo può essere fatto semplicemente scrivendo tale nuovo comando seguito dalla pressione del tasto RETURN.

Per esempio provate a scrivere la seguente stringa:

```
FOR N=1 TO 100:PRINT N:NEXT
```

seguita da RETURN.

Da questo momento in poi premendo F7 si otterrà la stampa dei primi 100 numeri naturali.

A parte questa curiosità, particolarmente utile risulta il seguente comando:

```
LOAD«S»,8
```

che, assieme a LIST associato a F1, consente di manipolare con meno fatica la DIRECTORY dei dischi.

Coloro che invece desiderano mantenere il default RUN per F7, è sufficiente che premano solo il tasto RETURN quando il programma Basic pone la domanda di cambio.

La routine si attiva semplicemente con un SYS 679 e si disattiva, senza venire distrutta, con la pressione di RUN/STOP e RESTORE.