

software COMMODORE 64

Questo mese qualche utility. La prima, Goto & Gosub Calcolati, consente di indirizzare dei salti alle linee di programma desiderate in forma parametrica (Goto n, Gosub n, dove n è il numero di riga).

La seconda ci permetterà invece di far scorrere sullo schermo in modo fine, orizzontalmente, una o più stringhe che saranno riprodotte in caratteri giganti.

Un'ulteriore utility ci darà invece la possibilità di avere, sull'ultima riga dello schermo, una scritta di 255 caratteri che, sfruttando la gestione dell'interrupt concessa dal sistema, scorrerà in orizzontale da destra verso sinistra permettendo delle presentazioni molto interessanti. La scritta mobile viene composta con il programma presentato e archiviata su disco o nastro sotto forma di una routine in LM. Essa potrà in qualunque momento essere richiamata dall'interno di un programma ed avviata con un SYS.

Per finire, alcune note riguardanti la stampante Riteman C+ usata insieme all'Easy Script che pubblichiamo... così come le abbiamo ricevute.

Goto & Gosub calcolati di Stefano Minardi - Firenze

Il programma si compone di 2 routine LM allocate in \$02A7 e \$C000. La prima costituisce un ampliamento dell'interprete Basic e consente di utilizzare una variabile o un'espressione aritmetica in luogo del numero di riga con le istruzioni GOTO e GOSUB. Le righe 200-270 contengono una dimostrazione della nuova sintassi. Naturalmente l'indirizzamento mediante il

```

C000 A0 00      LDY #000
C002 84 FB      STY $FB
C004 A3 A0      LDA #$A0
C006 85 FC      STA $FC
C008 01 FB      LDA ($FB),Y
C00A 91 FB      STA ($FB),Y
C00C E8 FB      INC $FB
C00E D0 F9      BNE $C008
C010 E8 FC      INC $FC
C012 A5 FC      LDA $FC
C014 C9 C0      CMP #000
C016 30 F0      BMI $C008
C018 A9 36      LDA #36
C01A 85 01      STA $01
C01C A9 A6      LDA #$A6
C01E 8D 1E A0   STA $A01E
C021 A9 02      LDA #02
C023 8D 1F A0   STA $A01F
C025 A9 AD      LDA #$AD
C028 8D 3B A9   STA $A93B
C02B A9 02      LDA #02
C02D 8D 3C A9   STA $A93C
C030 80         RTS
02A7 20 04 02   JSR #02B4
02A9 40 A3 A9   JMP #A9A3
02AD 20 04 02   JSR #02B4
02B0 20 A3 A9   JSR #A9A3
02B3 80         RTS
02D4 20 0A AD   JSR #AD0A
02B7 20 F7 B7   JSR #B7F7
02BA 80         RTS
    
```

Disassemblato della routine LM del programma Goto e Gosub Calcolati.

```

10 RESTORE PK=679:707
20 READA:IFAC0THEN50:977
30 POKEPK,A:PK=PK+1:GOTO20:1503
50 PK=49152:644
60 READA:IFAC0THEN100:1061
70 POKEPK,A:PK=PK+1:GOTO60:1547
100 SYS49152:REM*** TRASFERISCE ROM ***:2133
200 A=200:GOTOA+50:1120
210 PRINT"210":RETURN:778
220 PRINT"220":RETURN:789
230 PRINT"230":RETURN:800
240 PRINT"PROVA GOSUB CALCOLATO":RETURN:2159
250 PRINT"PROVA GOTO CALCOLATO":1898
260 FORT=240T0210STEP-10:1294
270 GOSUB:NEXT:END:614
1000 DATA 32,180,2,76,163,168,32,180,2,32,163,168,96,32,138,2826
1010 DATA 173,32,247,183,96,-1,1402
1020 DATA 160,0,132,251,169,160,133,252,177,251,145,251,230,251,208,3222
1030 DATA 248,230,252,165,252,201,192,48,240,169,54,133,1,169,166,2901
1040 DATA 141,30,160,169,2,141,31,160,169,173,141,155,168,169,2,2799
1050 DATA 141,156,168,96,-1,1039
    
```

Routine per abilitare il Goto e Gosub Calcolati. Nel listato è contenuto un esempio di applicazione.

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.

semplice numero di linea è ancora possibile. La seconda — che è rilocabile — provvede a trasferire l'interprete Basic nella RAM sottostante, a disabilitare l'interprete ROM e a modificare i puntatori delle istruzioni GOTO e GOSUB in modo che le due istruzioni vengano eseguite a partire dalla nuova routine posta in \$02A7. La routine in \$C000 può essere cancellata subito do-

po la SYS che la attiva, per cui le uniche locazioni occupate realmente sono quelle in \$02A7-02BA, oltre ovviamente a tutti gli 8K di RAM sotto l'interprete. Comunque, impiegando il Basic, non si avrà nessuna perdita di memoria. La nuova routine non ha effetto con l'istruzione IF...THEN se il GOTO o il GOSUB non vengono espressamente inseriti.

Striscia scorrevole

di Stefano Minardi - Firenze

Le linee 1000-1220 contengono una routine LM che consente di "scrollare" una scritta attraverso lo schermo,

C000 20 30 C1	JSR \$C130	C06D 10 F8	BPL \$C067	C0E9 9D 08 06	STA \$0608,X
C003 4C B7 C0	JMP \$C0B7	C06F A9 37	LDA ##37	C0EC BD 00 A6	LDA \$A600,X
C006 EA	NOP	C071 85 01	STA \$01	C0EF 9D 30 06	STA \$0630,X
C007 C9 40	CMP ##40	C073 58	CLI	C0F2 BD 00 A7	LDA \$A700,X
C009 30 23	BMI \$C02E	C074 86 FD	STX \$FD	C0F5 9D 58 06	STA \$0658,X
C00B C9 60	CMP ##60	C076 20 AC C0	JSR \$C0AC	C0F8 CA	DEX
C00D 10 05	BPL \$C014	C079 A2 00	LDX ##00	C0F9 10 CD	BPL \$C0C8
C00F 18	CLC	C07B BD F0 02	LDA \$02F0,X	C0FB 20 E4 FF	JSR \$FFE4
C010 E9 3F	SBC ##3F	C07E 18	CLC	C0FE D0 2B	BNE \$C12B
C012 10 1A	BPL \$C02E	C07F 0A	ASL	C100 EE C9 C0	INC \$C0C9
C014 C9 80	CMP ##80	C080 90 07	BCC \$C089	C103 EE CF C0	INC \$C0CF
C016 10 05	BPL \$C01D	C082 48	PHA	C106 EE D5 C0	INC \$C0D5
C018 18	CLC	C083 AD FD 02	LDA \$02FD	C109 EE DB C0	INC \$C0DB
C019 E9 1F	SBC ##1F	C086 91 FB	STA (\$FB),Y	C10C EE E1 C0	INC \$C0E1
C01B 10 11	BPL \$C02E	C088 68	PLA	C10F EE E7 C0	INC \$C0E7
C01D C9 C0	CMP ##C0	C089 C8	INY	C112 EE ED C0	INC \$C0ED
C01F 10 05	BPL \$C026	C08A C0 08	CPY ##08	C115 EE F3 C0	INC \$C0F3
C021 18	CLC	C08C 30 F0	BMI \$C07E	C118 AD F8 02	LDA \$02F8
C022 E9 3F	SBC ##3F	C08E EE B1 C0	INC \$C0B1	C11B D0 49	BNE \$C166
C024 10 08	BPL \$C02E	C091 20 AC C0	JSR \$C0AC	C11D A9 00	LDA #\$00
C026 C9 FF	CMP ##FF	C094 E8	INX	C11F 85 A2	STA \$A2
C028 D0 04	BNE \$C02E	C095 E0 08	CPX ##08	C121 A5 A2	LDA \$A2
C02A A9 5E	LDA ##5E	C097 30 E2	BMI \$C07B	C123 CD FC 02	CMP \$02FC
C02C 10 07	BPL \$C035	C099 A6 FD	LDX \$FD	C126 30 F9	BMI \$C121
C02E C9 80	CMP ##80	C09B A9 A0	LDA ##A0	C128 4C C6 C0	JMP \$C0C6
C030 30 03	BMI \$C035	C09D 8D B1 C0	STA \$C0B1	C12B A9 37	LDA #\$37
C032 18	CLC	C0A0 AD AD C0	LDA \$C0AD	C12D 85 01	STA \$01
C033 E9 7F	SBC ##7F	C0A3 18	CLC	C12F 6D	RTS
C035 8D FF 02	STA \$02FF	C0A4 69 08	ADC ##08	C130 AD FB 02	LDA \$02FB
C038 A9 00	LDA ##00	C0A6 8D AD C0	STA \$C0AD	C133 8D 20 D0	STA \$D020
C03A 85 FB	STA \$FB	C0A9 4C B9 C0	JMP \$C0B9	C136 8D 21 D0	STA \$D021
C03C AD FE 02	LDA \$02FE	C0AC A9 00	LDA ##00	C139 A9 93	LDA #\$93
C03F D0 06	BNE \$C047	C0AE 85 FB	STA \$FB	C13B 20 D2 FF	JSR \$FFD2
C041 A9 D0	LDA ##D0	C0B0 89 A0	LDA ##A0	C13E A2 00	LDX #\$00
C043 85 FC	STA \$FC	C0B2 85 FC	STA \$FC	C140 AD FA 02	LDA \$02FA
C045 D0 04	BNE \$C04B	C0B4 A0 00	LDY ##00	C143 9D 00 D8	STA \$D800,X
C047 A9 D8	LDA ##D8	C0B6 60	RTS	C146 9D 00 D9	STA \$D900,X
C049 85 FC	STA \$FC	C0B7 A2 FF	LDX ##FF	C149 9D 00 DA	STA \$DA00,X
C04B AC FF 02	LDY \$02FF	C0B9 E8	INX	C14C 9D 00 DB	STA \$DB00,X
C04E F0 10	BEQ \$C060	C0BA BD A7 02	LDA \$02A7,X	C14F E8	INX
C050 A5 FB	LDA \$FB	C0BD F0 03	BEQ \$C0C2	C150 D0 F1	BNE \$C143
C052 18	CLC	C0BF 4C 07 C0	JMP \$C007	C152 20 AC C0	JSR \$C0AC
C053 69 08	ADC ##08	C0C2 A9 36	LDA ##36	C155 A9 20	LDA #\$20
C055 85 FB	STA \$FB	C0C4 85 01	STA \$01	C157 91 FB	STA (\$FB),Y
C057 A5 FC	LDA \$FC	C0C6 A2 27	LDX ##27	C159 E6 FB	INC \$FB
C059 69 00	ADC ##00	C0C8 BD 00 A0	LDA \$A000,X	C15B D0 FA	BNE \$C157
C05B 85 FC	STA \$FC	C0CB 9D 40 05	STA \$0540,X	C15D E6 FC	INC \$FC
C05D 88	DEY	C0CE BD 00 A1	LDA \$A100,X	C15F A5 FC	LDA \$FC
C05E D0 F0	BNE \$C050	C0D1 9D 68 05	STA \$0568,X	C161 C9 C0	CMP ##C0
C060 78	SEI	C0D4 BD 00 A2	LDA \$A200,X	C163 30 F0	BMI \$C155
C061 A9 33	LDA ##33	C0D7 9D 90 05	STA \$0590,X	C165 60	RTS
C063 85 01	STA \$01	C0DA BD 00 A3	LDA \$A300,X	C166 AD C9 C0	LDA \$C0C9
C065 A0 07	LDY ##07	C0DD 9D B8 05	STA \$05B8,X	C169 D0 B2	BNE \$C11D
C067 B1 FB	LDA (\$FB),Y	C0E0 BD 00 A4	LDA \$A400,X	C16B CE F9 02	DEC \$02F9
C069 99 F0 02	STA \$02F0,Y	C0E3 9D E0 05	STA \$05E0,X	C16E D0 AD	BNE \$C11D
C06C 88	DEY	C0E6 BD 00 A5	LDA \$A500,X	C170 F0 B9	BEQ \$C12B
				C172 00	BRK

Disassemblato della routine LM del programma «Striscia Scorrevole».

da destra verso sinistra, utilizzando dei caratteri di 8x8 byte. Il programma Basic consente di inserire una stringa e di selezionare le caratteristiche della striscia:

NS (linea 35): numero di stringhe che si vogliono far comparire;

LO (linea 50): velocità di scorrimento della striscia (0-255); il valore è quello del ritardo;

CO (linea 55): colore della scritta (0-15);

BA (linea 60): colore dello sfondo (0-15);

GR (linea 65): codice di schermo (0-255) del carattere con cui comporre le lettere della striscia;

UL (linea 70): selezione set caratteri: maiuscole/grafici (0), minuscole/maiuscole (1);

RV (linea 75): codice di schermo (0-255) del carattere con cui costruire lo sfondo di ciascuna lettera della striscia (i bit posti a zero nella composizione dei caratteri video);

PS (linea 95): se zero, il ritorno della routine si ha con la pressione di un tasto; se maggiore di zero (max. 255) si ha il ritorno dopo un numero di passaggi della scritta pari al valore della variabile PS.

Le linee 100 possono contenere un numero di stringhe pari al valore assegnato alla variabile NS. Ovviamente le stringhe devono essere inserite in una matrice alfanumerica come quella nella linea 100 e con un numero d'ordine

```

95 PS=0:484
100 A$(1)="MICROCOMPUTER+++++++" 2928
200 FORT=1TONS/A#A#(T) 1568
201 L=LEN(A#):IFL>255THENR=LEFT$(A#,26) 2362
205 IFL<31THENR#=""*A#L=L+1 GOTO205 2469
210 POKE760,0:IFPS>0THENPOKE760,1:POKE761,PS 2337
220 POKE766,UL:739
230 POKE765,GR:740
240 POKE764,LO:751
250 POKE763,BA:736
260 POKE762,CO:505
270 POKE4949,RV:643
270 FORY=1TOL/POKE678+Y,ASC(MID$(A#,Y,1)):2266
280 NEXTY/POKE678+Y,0 969
285 POKE53272,21:583
290 SYS49152:454
300 POKE49325,0:NEXTT:823
310 END:183
1000 DATA32,48,193,76,183,192,234,201,64,48,35,201,96,16,5,24,2947
1010 DATA233,63,16,26,201,129,16,5,24,233,31,16,17,201,192,16,2925
1020 DATA5,24,233,63,16,8,201,255,208,4,169,94,16,7,201,128,2055
1030 DATA48,3,24,233,127,141,255,2,169,0,133,251,173,254,2,208,2749
1040 DATA6,169,208,133,252,208,4,169,216,133,252,172,255,2,240,16,2919
1050 DATA165,251,24,105,8,133,251,165,252,105,0,133,252,136,208,240,3009
1060 DATA120,169,51,133,1,160,7,177,251,153,240,2,136,16,248,169,2085
1070 DATA55,133,1,88,134,253,32,172,192,162,0,189,240,2,24,10,2738
1080 DATA144,7,72,173,253,2,145,251,104,200,192,8,48,240,238,177,2910
1090 DATA192,32,172,192,232,224,8,48,226,166,253,169,160,141,177,192,3136
1100 DATA173,173,192,24,105,8,141,173,192,76,185,192,169,0,133,251,3040
1110 DATA169,160,133,252,160,0,96,162,255,232,189,167,2,240,3,76,2946
1120 DATA7,192,169,54,133,1,162,39,189,0,160,157,64,5,189,0,2715
1130 DATA161,157,104,5,189,0,162,157,144,5,189,0,163,157,184,5,2869
1140 DATA189,0,164,157,224,5,189,0,165,157,8,6,189,0,166,157,2795
1150 DATA48,6,189,0,167,157,88,6,202,15,205,32,228,255,208,43,2846
1160 DATA238,201,192,238,207,192,238,213,192,238,219,192,238,225,192,238,3412
1170 DATA231,192,238,237,192,238,243,192,173,248,2,208,73,169,0,133,3164
1180 DATA162,165,162,205,252,2,48,249,76,198,192,163,55,133,1,96,3038
1190 DATA173,251,2,141,32,208,141,33,208,169,147,32,210,255,162,0,3050
1200 DATA173,250,2,157,0,216,157,0,217,157,0,218,157,0,219,232,2916
1210 DATA208,241,32,172,192,169,32,145,251,230,251,208,250,230,252,165,3328
1220 DATA252,201,192,48,240,96,173,201,192,208,178,206,249,2,208,173,240,195,-1 3793

```

Striscia scorrevole. Nel listato è compreso un esempio.

Note per stampante Riteman C+

Dopo aver letto la prova da voi pubblicata e dopo aver confrontato i vari prodotti in commercio, ho acquistato per il mio C64 la stampante Riteman C+, di cui per ora mi devo ritenere veramente soddisfatto.

Vi voglio ora inviare un piccolo schema che può essere utile a coloro che vogliono sfruttare appieno questa stampante in unione con il diffusissimo word processing Easy Script.

```

Printer type 0
Dip Switch giu/giu (mode Standard 1)

Espanso :ABCabc123 F1/[ e F1/]
Reverse : ABCabc123 F1/( e F1/)

```

```

Printer type 1
Dip Switch giu/su (mode Plus)

Espanso :ABCabc123 F1/[ e F1/]
Esempio di testo in 132 colonne

```

(Per ottenere le 132 cl. e' necessario passare da Printer Type 0, stampare alcune righe e passare poi in Pr.Type 1).

```

Printer type 3
Dip Switch giu/su (mode Plus)

```

Nel seguito ci si riferirà ai dip-switch 2 e 3 della stampante e, per quanto riguarda «Printer Type», alla domanda che Easy Script pone in fase di inizializzazione e che può essere in ogni momento modificata con i tasti RUN STOP/RESTORE.

Si possono inoltre ottenere caratteri particolari ad esempio la parentesi graffa:

{con F1/2} con F1/3.

Le configurazioni che non compaiono devono ritenersi non possibili o non completamente controllabili.

```

Sottolin.:ABCabc123 F1/[ e F1/]
Neretto 1:ABCabc123 F1/; e F1/;
Neretto 5:ABCabc123 come sopra 5 volte

```

```

Printer type 1
Dip Switch su/su (NLQ)

Espanso .:ABCabc123 F1/[ e F1/]

Esempio di testo in NLQ.

```

```

Printer type 4
Dip Switch su/su (NLQ)

Sottolin.:ABCabc123 F1/[ e F1/]
Neretto 1:ABCabc123 F1/; e F2/;
Neretto 5:ABCabc123 come sopra 5 volte

```

progressivo. Se si inseriscono varie stringhe, il programma le eseguirà in successione, (ponendo PS pari al numero delle stringhe. N.d.r.) ritornando dalla routine LM alla pressione di un tasto o dopo il numero specificato di passaggi per ogni stringa.

Ciascuna stringa può contenere al massimo 26 caratteri.

Tutte le funzioni di formazione dei caratteri 8 x 8 e di stampa sono svolte dalla routine LM.

Generatore di scritte mobili

di Paolo Agostini - Padova

Uno dei possibili modi per dare un «look» professionale ad un programma è quello di far scorrere orizzontalmente sullo schermo una scritta, ricorrendo a quello che viene chiamato «smooth scrolling» dello schermo. Purtroppo l'italiano — lingua aulica di retori e poeti — manca di quei termini d'uso comune che ci sarebbero indispensabili per tradurre questa definizione. In inglese «smooth» — oltre al senso di «liscio, non ruvido, levigato» — ha anche il senso di «scorrevole, senza scossoni, privo di sobbalzi». Il verbo «to scroll» ha invece il senso di «svolgere» qualcosa che è stato arrotolato. Per capire il senso di questa frase bisogna riandare ai tempi in cui si faceva del cinema con mezzi di fortuna. Il cow-boy prendeva posto su un cavallo finto che veniva fatto dondolare avanti e indietro e — per dare l'illusione del movimento — veniva fatto scorrere alle spalle dell'attore il fondale, che altro non era se non un ROTOLO continuo di carta su cui era dipinta la scenografia, posto su due rulli che gli consentivano di girare. Lo schermo del vostro monitor o del vostro televisore viene considerato come uno di quei rotoli, che può essere svolto per consentirvi di «vedere» dentro la memoria del vostro computer.

Il modo più comune per il computer di effettuare uno scroll è quello di andare a capo a piè di pagina: come tutti sanno, l'effetto di questo comando è di far apparire una nuova riga in fondo allo schermo e di far scomparire la riga più in alto. Il Commodore 64 è fornito di un chip video che consente di effettuare lo scroll sia in direzione verticale che in direzione orizzontale. Ma perché «smooth», cioè «senza scossoni»? Provate a digitare il seguente programmino in Basic:

```
100 PRINT CHR$(147)
110 INPUT "TESTO";M$
120 FOR I=40 TO 40+LEN(M$)
130 PRINT CHR$(19)
```

```
140 FOR J=1 TO 10
150 PRINT CHR$(17)::NEXT J
160 PRINT TAB(40);MID$(M$,I-39,39); " "
170 FOR K=1 TO 100:NEXT K:NEXT I
180 GOTO 120
```

Come si vede il movimento è a sobbalzi in quanto i caratteri vengono po-

sizionati uno alla volta e la riga si sposta di un intero carattere. Lo scroll più comune è quello della «finestra» vice-verso l'alto: quando si scrive nel-

```
100 REM ***** 1493
110 REM * GENERATORE SCRITTE MOBILI * 2231
120 REM * PAOLO AGOSTINI 1986 * 1900
130 REM ***** 1523
140 REM 283
150 PRINT CHR$(8);CHR$(5);POKE 53272,23;POKE 53286,14;2352
160 PRINT CHR$(147);CHR$(17);" *ATTENDERE PREGO*"POKE 53281,14;3237
170 FOR I=49152 TO 49468:READ A:CK=CK+A:2387
180 POKE I,A:NEXT IF CK<>38338 THEN PRINT CHR$(147);"ERRORE DATI"(END:3675
190 SET=49152;IN=49155;RE=49158;SA=49161;TE=49468;PLOT=58640;SYS (IN):PRINT 4984
200 POKE 214,5;POKE 211,0;SYS (PLOT);POKE TE-1,32;2519
201 REM "E NON SI VOGLIONO "MARCARE" GLI ASTERISCHI SCORREVOLI:4248
202 REM "SI DEVE CANCELLARE LA RIGA 203 SEGUENTE: 3012
203 FOR I=TE TO TE+256 STEP 10:POKE I,43:NEXT:2455
210 PRINT "INTRODURRE TESTO (LUNGHEZZA MASSIMA):PRINT "256 LETTERE)":4259
220 POKE 214,8;POKE 211,0;SYS (PLOT);T=0;2151
230 PRINT ">";CHR$(157);1100
240 GET A#:IF A#="" THEN 240:1491
250 A=ASC (A#):IF A=34 THEN 240:1861
260 IF A=13 THEN A#=" "POKE TE+T,32;GOTO 350:2301
270 IF A=20 AND T>1 THEN T=T-1:PRINT A#:GOTO 320:2830
280 IF T>255 THEN 240:1172
290 IF A<32 THEN 240:932
300 IF (A<127 AND A<160) THEN 240:1708
310 PRINT A#;POKE TE+T,PEEK (1344+T);T=T+1;2480
320 T=STR$(255-T);T=RIGHT$(T$,LEN (T$)-1);2501
330 T=RIGHT$("000"+T$,3):FOR I=1 TO 3:POKE 1263+I,ASC(MID$(T$,I,1));NEXT:3894
340 GOTO 230:403
350 PRINT CHR$(147);SYS (SET);PRINT "MENU":2119
360 PRINT PRINT CHR$(18);" 1 ";CHR$(146);" MODIFICA IL TESTO":3091
370 PRINT PRINT CHR$(18);" 2 ";CHR$(146);" ALVA PROGRAMMA SU DISCO":3605
380 PRINT PRINT CHR$(18);" 3 ";CHR$(146);" ALVA PROGRAMMA SU NASTRO":3717
390 PRINT PRINT CHR$(18);" 4 ";CHR$(146);" FINE LAVORO":2729
400 PRINT PRINT:PRINT "SCELTE SCELGI ?":1890
410 POKE 214,20;POKE 211,0;SYS (PLOT);FOR I=1 TO 39:PRINT CHR$(192);NEXT:3541
420 PRINT PRINT "VISUALIZZAZIONE DEL TESTO IMMESSO":3271
430 POKE 198,0;WAIT 198,1;GET A#:1455
440 POKE 214,12;POKE 211,15;SYS (PLOT);PRINT A#:2189
450 IF A#="1" THEN SYS (RE):RUN:1611
460 IF A#="2" THEN POKE 49371,8;GOSUB 560;SYS (SA);GOSUB 500;RUN:2990
470 IF A#="3" THEN POKE 49371,1;GOSUB 560;SYS (SA);RUN:2614
480 IF A#="4" THEN SYS 64738:1484
490 GOTO 430:555
500 OPEN 15,8,15:794
510 INPUT# 15,E1,E1#,E2,E3;CLOSE 15:1560
520 PRINT "STATUS DISCO"E1,"E1#,"E2","E3:2182
530 PRINT CHR$(17);"PREMERE UN TASTO":1994
540 POKE 198,0;WAIT 198,1;POKE 198,0:1453
550 RETURN:182
560 PRINT "INTRODURRE UN NUMERO (0-99) PER":2493
570 PRINT "CONTRASSEGNARE IL PROGRAMMA."":2305
580 A#=""INPUT A#:A=VAL (A#):1453
590 IF A<0 OR A>99 THEN PRINT CHR$(145);CHR$(145);GOTO580:2802
600 IF LEN (A#)<2 THEN A#="."A#:1794
610 POKE 49456,ASC (LEFT$(A#,1)):1413
620 POKE 49457,ASC (RIGHT$(A#,1)):1425
630 RETURN:262
640 DATA 76,34,192,76,12,192,76,188,192,76,1956
650 DATA 211,192,162,0,169,42,157,59,193,232:2041
660 DATA 208,250,24,32,68,229,169,250,160,1902
670 DATA 192,32,30,171,96,120,169,70,160,192:2056
680 DATA 141,20,3,140,21,3,173,17,208,41:1841
690 DATA 127,141,17,208,169,186,141,18,208:1985
700 DATA 169,129,141,26,208,169,0,141,56,193:2095
710 DATA 88,96,173,25,208,141,25,208,48,7:1964
720 DATA 173,13,220,88,76,49,234,173,58,193:2076
730 DATA 201,6,240,16,169,6,141,58,193,169,1:2116
740 DATA 141,18,208,32,126,192,76,188,254:1989
750 DATA 169,0,141,58,193,169,231,141,18,208:2142
760 DATA 169,200,141,22,208,76,188,254,174:2056
770 DATA 57,193,202,142,57,193,142,22,208:1754
780 DATA 224,255,240,1,96,162,7,142,57,193:1816
790 DATA 142,22,208,162,0,189,193,7,157,192:1877
800 DATA 7,232,224,39,208,245,162,0,169,1:1779
810 DATA 157,192,219,232,224,39,208,248,174:1908
820 DATA 56,193,232,142,56,193,189,59,193:1829
830 DATA 141,231,7,96,120,169,49,141,20,3,169:2009
840 DATA 234,141,21,3,169,240,141,26,208:1761
850 DATA 169,200,141,22,208,88,96,32,188,192:1995
860 DATA 32,231,255,24,162,8,32,186,255,162:1941
870 DATA 48,160,193,169,8,32,189,255,162:1824
880 DATA 0,160,192,134,251,132,252,169,251:1904
890 DATA 162,59,160,194,32,216,255,96,147:1889
900 DATA 103,101,110,101,114,97,116,111,114:1947
910 DATA 101,32,100,105,32,115,99,114,105:1968
920 DATA 116,116,101,32,109,111,98,105,108:1938
930 DATA 105,13,40,99,41,32,112,97,111,108:1950
940 DATA 111,32,97,103,111,115,116,105,110:1944
950 DATA 105,32,49,57,56,54,0,46,46,83,67:1944
960 DATA 82,79,76,76,0,0,0,0,0,1386
```

l'ultima riga dello schermo, lo schermo si sposta automaticamente verso l'alto onde consentire di continuare la scrittura. Anche questo è un sistema che va a sbalzi, dato che un'intera riga «balza» sullo schermo. Il video chip del Commodore 64 è però in grado di effettuare anche un movimento più graduale, spostando l'intera «finestra» video di un solo pixel alla volta (un pixel è uno di quei puntini che compongono un carattere sullo schermo e che possono essere accesi o spenti. Ogni carattere è composto di 8 pixel orizzontali x 7 pixel verticali). Spostandoli uno alla volta si otterrà un movimento graduale e — se la velocità di scorrimento sarà sufficiente — il nostro occhio avrà l'impressione di un movimento senza soluzione di continuità.

Il fatto però che un carattere appaia all'improvviso al bordo dello schermo guasta l'illusione dello scorrimento. Per questa ragione si appropfiterà della possibilità fornita dal video chip di ridurre le dimensioni dello schermo da 40 a 38 colonne mediante l'espansione del bordo. Provate a digitare il seguente comando:

POKE 53270,PEEK (53270) AND 247

Dopo aver premuto RETURN lo schermo si «raggrinzirà» e la prima e l'ultima colonna dello schermo scompariranno. Tale scomparsa sarà soltanto apparente, in quanto abbiamo costretto il bordo a fornirci una comoda «quinta» teatrale mediante la diaframmazione. Ora è possibile effettuare lo scrolling mediante una tecnica relativamente semplice: per prima cosa si porta il registro che determina la posizione dello schermo al valore minimo (o massimo, a seconda della direzione di scorrimento); si mette un carattere nella colonna nascosta dal bordo; si sposta di un pixel alla volta questo carattere verso l'esterno e, quando tutto il carattere è fuori, si sposta di un carattere tutta la riga. A questo punto si ricomincia, riposizionando un carattere nella parte nascosta e ricominciando daccapo. Per riportare lo schermo nello stato normale si dà il comando:

POKE 53270,PEEK(53270) OR 8

Questo sistema ha purtroppo come contropartita il fatto che TUTTO lo schermo si muove di un pixel alla volta cosicché tutto ciò che è stampato sullo schermo (all'infuori della scritta scorrevole) è soggetto ad una rapida vibrazione che ne rende impossibile la lettura. La soluzione a ciò è quindi l'utilizzo contemporaneo di una routine che — sfruttando l'interrupt generato dalla riga di raster (il «pennello» elet-

tronico che «scrive» l'immagine sullo schermo) provvede a far sì che la routine di «smooth scrolling» entri in funzione soltanto quando il «pennello» arriva a disegnare le ultime due righe

dello schermo. Così facendo è possibile avere uno schermo fisso privo di vibrazioni per 22 righe, mentre nelle due righe inferiori scorre la scritta. Una routine del genere in Basic è improprio-

Smooth Scrolling				
	1		ORG	#C000
	2	*		
	3	*		
C000:	4C 22 00	4	JMP	SET
C003:	4C 9C 00	5	JMP	INSTALL
C006:	4C 8C 00	6	JMP	RESET
C009:	4C D3 00	7	JMP	SAVE
	8	*		
	9	*		
C00C:	A2 00	10	INSTALL	LDX #000 ;Inizializzazione
C00E:	A9 2A	11	LDA	#42
C010:	90 38 C1	12	SET1	STA TFST0,X
C013:	F8	13	INX	
C014:	D0 FA	14	BNE	SET1
C016:	18	15	CLC	
C017:	20 44 E5	16	JSR	\$E544
C01A:	A9 FA	17	LDA	#MSG
C01C:	A0 00	18	LDY	#MSG
C01E:	20 1E AB	19	JSR	\$AB1E
C021:	60	20	RTS	
	21			
C022:	78	22	SET	SET ;abilita routine di interrupt
C023:	A9 46	23	LDA	#FOUR10
C025:	A0 00	24	LDY	#FOUR10
C027:	80 14 03	25	STA	\$0314
C02A:	8C 15 03	26	STY	\$0315
C02D:	A0 11 D0	27	LDA	\$D011
C030:	29 7F	28	AND	#17F
C032:	8D 11 D0	29	STA	\$D011
C035:	A9 BA	30	LDA	#EBA
C037:	8D 12 D0	31	STA	\$D012
C03A:	A9 81	32	LDA	#81
C03C:	8D 1A D0	33	STA	\$D01A
C03F:	A9 00	34	LDA	#00
C041:	8D 38 C1	35	STA	UN0
C044:	58	36	CLI	
C045:	60	37	RTS	
	38			
	39	*		
C046:	A0 19 D0	40	OUR10	LDA \$D019 ;routine principale di interrupt
C049:	8D 19 D0	41	STA	\$D019
C04C:	30 07	42	BMI	DOMOVE
C04E:	A0 00 DC	43	LDA	\$DC00
C051:	58	44	CLI	
C052:	4C 31 EA	45	JMP	\$EA31
	46	*		
	47	*		
C055:	A0 3A C1	48	DOMOVE	LDA COUNTER ;routines di scrolling
C058:	C9 06	49	CMR	#06
C05A:	F0 10	50	BEQ	AVANTI
C05C:	A9 06	51	LDA	#06
C05E:	8D 3A C1	52	STA	COUNTER
C061:	A9 01	53	LDA	#01
C063:	8D 12 D0	54	STA	\$D012
C066:	20 7E C0	55	JSR	MOVE
C069:	4C BC FE	56	JMP	\$FEBC
	57	*		
	58	*		
C06C:	A9 00	59	AVANTI	LDA #00
C06E:	8D 3A C1	60	STA	COUNTER
C071:	A9 E7	61	LDA	#E7
C073:	8D 12 D0	62	STA	\$D012
C076:	A9 08	63	LDA	#08
C078:	8D 16 D0	64	STA	\$D016
C07B:	4C BC FE	65	JMP	\$FEBC
	66	*		
	67	*		
C07E:	AE 39 C1	68	MOVE	LDX DUE
C081:	CA	69	DEX	
C082:	8E 39 C1	70	STX	THE
C085:	8E 16 00	71	STX	\$D016
C088:	E0 FF	72	CPX	#FF
C08A:	F0 01	73	BEQ	MOVE1
C08C:	60	74	RTS	
	75	*		
C08D:	A2 07	76	MOVE1	LDX #07
C08F:	8E 39 C1	77	STX	DUE
C092:	8E 16 00	78	STX	\$D016
C095:	A2 00	79	LDX	#00
C097:	BD C1 07	80	MOVE2	LDA \$07C1,X
C09A:	9D C0 07	81	STA	\$07C0,X
C09D:	E8	82	INX	
C0A0:	E0 27	83	CPX	#27
C0A3:	D0 F5	84	BNE	MOVE2
C0A2:	A2 00	85	LDX	#00
C0A4:	A9 01	86	LDA	#01
C0A6:	9D C0 DB	87	MOVE3	STA \$DBC0,X

Disassemblato del programma "Generatore di scritte mobili".

nibile a causa dell'alta velocità richiesta nella lettura del registro di interrupt, pertanto non rimane altra soluzione che la programmazione in linguaggio macchina, che non è però alla

portata di tutti. Questo programma è concepito in modo da creare su disco o su nastro una routine in linguaggio macchina che si incaricherà di far scorrere sulla parte inferiore dello

schermo una scritta a vostro piacimento (purché della lunghezza massima di 255 caratteri) che potrà contenere — per esempio — il vostro nome e indirizzo oppure le istruzioni di funzionamento del vostro programma. Dopo aver attivato il programma generatore Basic si dovrà introdurre la stringa da visualizzare sullo schermo. Dopo aver premuto RETURN il programma visualizzerà la stringa introdotta e tramite un menu fornirà le opzioni seguenti: modifica della stringa, salvataggio su nastro o su disco o fine lavoro. Se decidete di salvare su nastro o disco la stringa scorrevole, il programma vi chiederà di contrassegnare questa stringa con un numero da 0 a 99 (allo scopo di evitare problemi con il disk drive) e poi salverà la stringa scorrevole su nastro o disco col nome «xx SCROLL», dove xx è il numero che avrete fornito.

A questo punto non vi resta altro da fare che scrivere il vostro programma, facendo attenzione che le ultime due righe di schermo vengano lasciate libere e — possibilmente — che il colore del bordo e dello schermo siano uguali. Da programma sarà possibile caricare la routine in l.m. da disco o da nastro utilizzando come prima riga del vostro programma la riga seguente (anche se con un numero diverso a vostro piacimento):

```
10 IFX=0THENX=1:LOAD"xxSCROLL",8,1:REM CARICAMENTO DA DISCO
```

oppure

```
10 IFX=0THENX=1:LOAD"xxSCROLL",1,1:REM CARICAMENTO DA NASTRO
```

Lo scroll partirà col comando SYS49152, sarà gestito da interrupt (per cui il computer eseguirà normalmente il programma in Basic da voi scritto). Per fermare lo scroll basterà effettuare una chiamata SYS49158 e poi cancellare lo schermo (o la riga 24 a piacere). Se il colore bianco della scritta scorrevole non vi aggrada, sarà possibile cambiarlo da programma con un POKE 49317, numero (dove il numero del colore prescelto sarà quello stesso che viene normalmente utilizzato per cambiare il colore al bordo e allo schermo).

Vi ricordo che sia il programma in Basic che le righe di DATA sono molto critici e anche un piccolo errore può pregiudicarne il funzionamento, pertanto raccomandando la massima attenzione nella copiatura.

```

C0A9: E8      88      INX
C0AA: E9 27    99      CPX    ##27
C0AC: D0 F8    90      BNE    MOVE3
C0AE: AE 38 C1 91      LDX    UNO
C0B1: E8      92      INX
C0B2: 8E 38 C1 93      STX    UNO
C0B5: B0 38 C1 94      LDA    TEST0,X
C0B8: 8D F7 A7 95      STA    $07E7
C0BB: 60      96      RTS
          97
          98
          *
C0BC: 78      99      RESET SEI      ;disabilita routine interrupt
C0BD: A9 31   100     LDA    ##31
C0BF: 8D 14 03 101     STA    $A314
C0C2: A9 EA   102     LDA    ##EA
C0C4: 8D 15 03 103     STA    $A315
C0C7: A9 F0   104     LDA    ##F0
C0C9: 8D 1A D0 105     STA    $D01A
C0CC: A9 C8   106     LDA    ##C8
C0CE: 8D 16 D0 107     STA    $D016
C0D1: 58      108     CLI
C0D2: 60      109     RTS
          110
          *
          111
C0D3: 20 BC C0 112     SAVE  JSR    RESET ;salva RAM su disco [nastro]
C0D6: 20 E7 FF 113     JSR    $FFE7
C0D9: 18      114     CLC
C0DA: A2 08   115     LDX    ##08
C0DC: 20 BA FF 116     JSR    $FFBA
          117
C0DF: A2 30   117     LDX    #<FILNAM
C0E1: A0 C1   118     LDY    #>FILNAM
C0E3: A9 08   119     LDA    #8
C0E5: 20 BD FF 120     JSR    $FFBD
C0E8: A2 00   121     LDX    ##00
C0EA: A0 C0   122     LDY    #<C0
C0EC: 86 FB   123     STX    #FB
C0EE: 84 FC   124     STY    #FC
C0F0: A9 FB   125     LDA    ##FB
C0F2: A2 38   126     LDX    #<TEST0+256
C0F4: A0 C2   127     LDY    #>TEST0+256
C0F6: 20 D8 FF 128     JSR    $FFD8
C0F9: 60      129     RTS
          130
          *
          131
C0FA: 93      132     MSG  DFB    147
C0FB: 67 65 6E 133     TXT    'GENERATORE DI SCRITTE MOBILI'
C0FE: 65 72 61 74 6F 72 65 20
C106: 64 69 20 73 63 72 69 74
C10E: 74 65 20 6D 6F 62 69 6C
C116: 69
          134
C117: 00      134     DFB    13
C118: 28 63 29 135     TXT    '(C) Paolo Agostini 1986'
C11B: 20 70 41 4F 4C 4F 20 61
C123: 47 4F 53 54 49 4E 49 20
C12B: 31 39 38 36
C12F: 00      136     DFB    0
          137
          *
C130: 2E 2E 53 138     FILNAM TXT    '..scroll'
C133: 43 52 4F 4C 4C
C138: 00      139     UNO   DFB    0
C139: 00      140     DUE   DFB    0
C13A: 00      141     COUNTER DFB    0
C13B: 00      142     TESTO DFB    0
          143
          *
          144 * Qui va il testo che verra' visualizzato

```

--End assembly, 316 bytes, Errors: 0

Symbol table - alphabetical order:

AVANTI	=#C06C	COUNTER	=#C13A	DOMOVE	=#C055	DUE	=#C139
FILNAM	=#C130	INSTALL	=#C09C	MOVE	=#C07E	MOVE1	=#C08D
MOVE2	=#C097	MOVE3	=#C0A6	MSG	=#C0FA	OURIRQ	=#C046
RESET	=#C0BC	SAVE	=#C0D3	SET	=#C022	SET1	=#C010
TESTO	=#C13B	UNO	=#C138				

Symbol table - numerical order:

INSTALL	=#C09C	SET1	=#C010	SET	=#C022	OURIRQ	=#C046
DOMOVE	=#C055	AVANTI	=#C06C	MOVE	=#C07E	MOVE1	=#C08D
MOVE2	=#C097	MOVE3	=#C0A6	RESET	=#C0BC	SAVE	=#C0D3
MSG	=#C0FA	FILNAM	=#C130	UNO	=#C138	DUE	=#C139
COUNTER	=#C13A	TESTO	=#C13B				