

# 77 PROGRAMMI PER SPECTRUM

GRAFICA - BUSINESS GRAFICA - UTILITY - ANIMAZIONI - MUSICA - GIOCHI



GRUPPO  
EDITORIALE  
JACKSON

di Gaetano Marano



# **77** PROGRAMMI PER **SPECTRUM**

di  
**Gaetano Marano**



GRUPPO  
EDITORIALE  
JACKSON  
Via Rosellini, 12  
20124 Milano

© Copyright 1983 Gruppo Editoriale Jackson

Il Gruppo Editoriale Jackson ringrazia per il prezioso lavoro svolto nella stesura del volume le signore Francesca Di Fiore, Marta Menegardo e l'ing. Roberto Pancaldi.

Tutti i diritti sono riservati. Nessuna parte di questo libro può essere riprodotta, posta in sistemi di archiviazione, trasmessa in qualsiasi forma o mezzo, elettronico, meccanico, fotocopiatura ecc., senza l'autorizzazione scritta.

I contenuti di questo libro sono scrupolosamente controllati. Tuttavia, non si assume alcuna responsabilità per eventuali errori od omissioni. Le caratteristiche tecniche dei prodotti descritti, possono essere cambiate in ogni momento senza alcun preavviso. Non si assume alcuna responsabilità per eventuali danni risultanti dall'utilizzo di informazioni contenute nel testo.

Prima edizione: 1983

Stampato in Italia da:  
S.p.A. Alberto Matarelli - Milano - Stabilimento Grafico

Fotocomposizione:  
CorpoNove s.n.c. - Bergamo - tel. 035/22.33.63 - 22.33.65

# PREFAZIONE

Nel realizzare i programmi di questo libro si è cercato di sfruttare al massimo le nuove interessanti prestazioni dello Spectrum ed in particolare le nuove istruzioni BASIC, il Suono, il Colore e soprattutto la Grafica ad alta risoluzione.

Anche la scelta dei programmi è molto varia, essendoci infatti programmi di Grafica (30), Utility (8), Musica (5), Animazione (6), Giochi (7), Effetti Sonori (12), Elettronica e Circuiti Logici (4) ed altro (5).

Fiore all'occhiello di questo libro è senz'altro la 'BUSINESS GRAFICA' (9 programmi) realizzata, con una certa efficacia visiva e pratica superando alcune limitazioni di colore e di risoluzione che relegavano tali programmi solo a computer molto più costosi.

In tutti i programmi si è sfruttata anche la maggiore velocità dello Spectrum rispetto ai precedenti computer Sinclair ed in particolare nelle Animazioni che sono ora molto più veloci e realistiche. Altra caratteristica molto utilizzata è la possibilità sullo Spectrum di definire dei Caratteri Speciali, inseriti ovunque possibile per rendere i programmi molto curati anche esteticamente o per rendere più realistiche alcune situazioni di programma, come nei giochi dove si creano, per esempio, un sommergibile (nella Caccia al sommergibile), o degli omini (nel Tiro alla Fune o nel Campo Minato), ecc... Spero quindi che troverete molte cose utili da utilizzare sul vostro meraviglioso Spectrum.

Prima di terminare permettetemi di ringraziare, per l'apporto di collaborazione da loro ricevuto, i Sigg.ri Arenai e Harms della Easy Byte di Roma e la Sinclair Research (per il tramite dell'importatore, la GBC Italiana) per i due Spectrum utilizzati nello scrivere il libro, ed i Sigg.ri Dentale e Ferrari della TELAV di Roma per il Monitor a Colori (un Barco CM33) utilizzato per ottenere delle immagini ad alta qualità di cui le foto che ho realizzato spero diano buona testimonianza.

**Gaetano Marano**



# INDICE

<b>PREFAZIONE</b> .....	III
<b>NOTE IMPORTANTI SUI PROGRAMMI</b> .....	VIII
<b>GRAFICA</b> .....	1
1 Disegnatore Caratteri Speciali .....	2
2 Disegnatore a Colori .....	4
3 Plotter .....	7
4 Curve Matematiche .....	9
5 Disegno Poligoni .....	11
6 Plotaggio Ellissi .....	14
7 Disegno Quadrati .....	16
8 Spirali .....	17
9 Cono Tridimensionale .....	18
10 Cilindro Tridimensionale .....	19
11 Sfera Terrestre .....	20
Programmi di 'Business Grafica' .....	22
12 Diagramma a Torta .....	22
13 Cilindri 3 D .....	25
14 Istogrammi Orizzontali .....	27
15 Istogrammi Verticali .....	29
16 Istogrammi Verticali a Colori Affiancati .....	32
17 Istogrammi Verticali a Colori Sovrapposti .....	35
18 Istogrammi Verticali Tridimensionali .....	38
19 Diagramma .....	42
20 Diagramma a Superficie .....	45
21 NEW YORK .....	49
22 Big Bang .....	51
23 Disegno di Linee .....	51
24 Alveare .....	52
25 Piramide .....	52
26 Cielo Stellato .....	53
27 Montagne .....	54
28 Bandiera Regno Unito .....	55
29 Bandiera Stati uniti .....	57
30 Bandiera Giapponese .....	58

<b>PROGRAMMI VARI</b> .....	59
31 Fisica dei Reattori Nucleari .....	59
32 Lettura Veloce .....	62
33 Cronometro .....	65
34 Scritte Rotanti .....	67
35 Contapezzi .....	70
36 Simulatore di Porte Logiche .....	72
37 Progettazione di Multivibratore Monostabile con 555 .....	76
38 Progettazione di Oscillatore con 555 .....	81
39 Legge di OHM .....	86
<b>UTILITY</b> .....	89
40 Routine 'PAINT' in BASIC .....	89
Note sulle routines 'SCROLL' .....	91
41 Scroll a Sinistra .....	92
42 Scroll a Destra .....	93
43 Scroll in Alto .....	94
44 Scroll in Basso .....	95
45 Renumber .....	96
46 Lettore Memoria RAM/ROM .....	98
47 Conversione Decimale/Esadecimale ed inverso .....	100
<b>ANIMAZIONI</b> .....	103
48 Viaggio Spaziale .....	103
49 Lancio SPACE SHUTTLE .....	105
50 Animazione Esplosione .....	107
51 Caleidoscopio .....	108
52 Rombospirale .....	110
53 Programma Universale Animazione con PRINT AT .....	112
<b>MUSICA ED EFFETTI SONORI</b> .....	115
54 Miniorgano .....	115
55 Generatore di Musica Casuale — 130 Note .....	116
56 Metronomo .....	117
57 Scale Musicali .....	117
58 Programma Universale per Musica .....	118
59 Red Alert .....	120
60 Orologio .....	121
61 Mitra .....	121
62 Sirena Americana .....	121
63 Sirena a 2 Toni .....	122
64 Sirena Multitono .....	122
65 Squillo del telefono .....	122
66 Segnale telefonico di linea libera .....	123

67	Segnale telefonico di linea occupata .....	123
68	Generatore di suoni casuali .....	123
69	Din — Don .....	123
70	Grilli .....	124
<b>GIOCHI .....</b>		<b>125</b>
71	Caccia al Sommergibile .....	125
72	Tiro alla Fune .....	127
73	Roulette .....	130
74	Campo Minato .....	132
75	Flipper Giapponese .....	136
76	Tavola di Numeri e Lettere .....	138
77	1 — 40 DADI .....	140

## NOTE IMPORTANTI SUI PROGRAMMI

Dopo aver inserito i programmi, e PRIMA di premere RUN, è sempre bene registrare su cassetta il programma e controllare la avvenuta registrazione con 'VERIFY', ma ciò è particolarmente IMPORTANTE in tutti i programmi che contengono delle routines in linguaggio macchina ed in genere delle POKE. Queste perché se si aggiunge o si toglie per errore una istruzione della routine o se si sbaglia l'indirizzo della POKE, può capitare che il computer si blocchi o addirittura che il programma si cancelli.

Registrando il programma si può evitare quindi di doverlo riscrivere e sarà più facile trovare l'istruzione che si è inserita in modo errato.

Tutti i programmi sono perfettamente funzionanti in quanto sono stati stampati direttamente dal computer, dopo aver girato con successo, e riprodotti esattamente in questo libro per cui occorre soltanto inserirli con esattezza per vederli funzionare. A tale proposito è bene tenere presente di inserire con particolare cura gli spazi sia come numero che come posizione esattamente come nei programmi, poiché in molti casi sono determinanti per un funzionamento corretto. Per contare gli spazi brevi basta riferirsi alle linee superiori o inferiori, mentre per gli spazi più lunghi, questi sono stati indicati in alcuni programmi, con dei REM subito dopo la fine.

In alcuni programmi, che usano routines in linguaggio macchina per leggere la tastiera, è indicato di non inserire nel programma l'istruzione BORDER altrimenti il programma si blocca. Se comunque si desidera un dato colore per il bordo lo si può inserire con un comando diretto prima di scrivere il programma.

Infine i programmi sono stati scritti in modo da funzionare senza modifiche sia su Spectrum da 16K che da 48K.

# GRAFICA

## DISEGNATORE CARATTERI SPECIALI

Lo Spectrum ha la possibilità di definire 21 caratteri speciali memorizzabili tramite la funzione USR o tramite il comando POKE, e utilizzabili in un programma tramite l'istruzione CHR\$ o inseribili direttamente da tastiera premendo i tasti da A a U con il Computer in modo grafico.

Il programma di figura 1 è molto utile perchè permette di disegnare e modificare tali caratteri speciali in grandi dimensioni e quindi di memorizzarli sotto il carattere alfabetico prescelto.

Il programma permette inoltre (premendo il tasto '1' alla fine o anche durante il disegno) di visualizzare i codici delle POKE da effettuare nel caso si vogliano caricare molti caratteri speciali nel modo utilizzato in molti programmi di questo libro; vale a dire inserendo tutti i codici in istruzioni DATA e caricandoli con delle POKE direttamente nella RAM a partire dall'indirizzo 32600 (nello Spectrum 16K) o dall'indirizzo 65367 (nel 48K).

Le funzioni degli otto tasti utilizzati dal programma sono le seguenti:

tasto	Funzione
1	<b>visualizza i codici POKEs</b>
2	<b>carica il carattere speciale in memoria</b>
3	<b>modo 'disegno'</b>
4	<b>modo 'modifiche e spostamenti'</b>
5	<b>a sinistra</b>
6	<b>in basso</b>
7	<b>in alto</b>
8	<b>a destra</b>

La figura 2 mostra alcuni dei caratteri speciali realizzabili con il programma appena descritto, assieme ai relativi codici delle POKE. In particolare vi è un piccolo E.T. (2a), un PAC-MAN (2b), un'auto da corsa (2c), un omino (2d) ed il simbolo di microFarad (2e).

NOTA: Premendo il tasto '1' o il tasto '2' il cursore quadrato lampeggiante viene incluso nel disegno o nel calcolo dei codici POKE solo se si è in modo 'disegno'.

```

10 REM DISSEGNATURE
20 REM CARATTERI
30 REM SPECIALI
40 REM -----
100 BORDER 0: PAPER 0: INK 6: C
LS
110 LET U=PEEK 23675+256*PEEK 2
3675
111 PRINT AT 16,20: INK 7;"test
0 0=cOPY"
120 FOR a=4 TO 11
130 PRINT AT a,11, PAPER 1: INK
0:
140 NEXT a
144 PLOT 87,79: DRAW 0,65: DRAW
65,0: DRAW 0,-65: DRAW -65,0
150 LET l=11: LET c=11: LET i=0
160 LET k=CODE INKEY$-48
170 IF k>0 AND k<9 THEN GO SUB
k*100+1000
177 PRINT AT l,c: INK 0: PAPER
6: "+" : PAUSE 3
180 IF i=1 THEN PRINT AT l,c: I
NK 6: PAPER 1: CHR$ 143
188 IF i=1 THEN PRINT AT l,c: I
NK 0: PAPER 1: CHR$ 188
190 IF INKEY$="c" THEN COPY
200 GO TO 160
1100 FOR h=4 TO 11
1110 LET p=0
1120 FOR w=11 TO 18
1130 IF ATTR (h,w)=14 THEN LET p
=p+2+(18-w)
1140 NEXT w
1150 PRINT AT h,20: INK 4:p)"
1160 NEXT h
1170 RETURN
1200 PRINT AT 0,5: INK 7:"Premier
e il carattere scatto"
1205 LET t=CODE INKEY$-65
1210 IF t>01 THEN LET t=t-32
1220 IF t>=0 AND t<21 THEN GO TO
1234
1230 GO TO 1205
1240 PRINT AT 0,5: "
: REM 28 spazi
1244 FOR h=4 TO 11
1250 LET p=0
1260 FOR w=11 TO 18
1270 IF ATTR (h,w)=14 THEN LET p
=p+2+(18-w)
1280 NEXT w
1290 POKE U+t*8+(h-4),p
1300 NEXT h
1310 RETURN
1333 LET i=1: RETURN
1444 LET i=0: RETURN

```

```

1500 IF C=11 THEN RETURN
1555 LET C=C-1: RETURN
1600 IF L=11 THEN RETURN
1650 LET L=L+1: RETURN
1700 IF L=4 THEN RETURN
1777 LET L=L-1: RETURN
1800 IF C=18 THEN RETURN
2000 LET C=C+1: RETURN

```

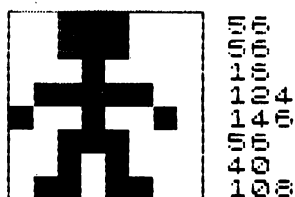
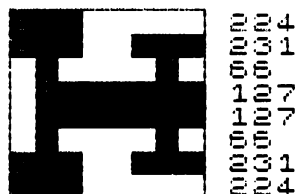


Figura 2

## DISEGNATORE A COLORI

Con il programma di figura 3 si possono realizzare sullo schermo tramite i tasti 5,6,7,8 dei disegni ad otto colori formati con sedici caratteri grafici dello Spectrum (vedere a pag. 186 del manuale).

All'inizio occorre inserire il colore scelto per il bordo ed il colore del fondo.

Il simbolo (che all'inizio è lo spazio inverso) ed il colore (che all'inizio è bianco o nero) possono essere continuamente scelti e modificati durante il disegno premendo (senza SHIFT) i tasti indicati nella Tabella 1.

Come già detto per disegnare con il carattere prescelto occorre premere i tasti 5,6,7,8 INSIEME al tasto CAPS SHIFT.

Infine per spostarsi o per modificare il disegno basta premere il tasto 'A' che seleziona lo spazio vuoto per poi cambiarlo di nuovo quando si vuol ritornare a disegnare.

La figura 4 mostra un esempio di disegno realizzato con il Disegnatore a colori, tale disegno (come qualsiasi altro) può essere registrato su cassetta premendo BREAK e quindi eseguendo in modo diretto il comando 'SAVE "nome" SCREEN\$' per continuare si può premere subito dopo CONTINUE ed ENTER.

NOTA: Se si vuole disegnare meno velocemente si può aggiungere la linea  
233 PAUSE n  
dove n può essere compreso tra 1 e 50.

```
10 REM DISEGNATORE A COLORI
20 REM -----
100 INPUT "Inserire colore bord
o (0--7) e premere ENTER"
;b
110 IF b>7 OR b<0 THEN GO TO 10
120 INPUT "Inserire colore fond
o (0--7) e premere ENTER"
;p
130 IF p>7 OR p<0 THEN GO TO 10
140 IF p>3 THEN LET i=0
150 IF p<4 THEN LET i=7
160 LET m=i: BORDER b: PAPER p:
INK i: CLS
170 LET l=0: LET c=0: LET x=140
180 LET k=CODE INKEY$-7
190 IF k>4 THEN LET k=k-36
200 IF k>12 THEN LET k=k-9
210 IF k>28 THEN LET k=k-32
220 IF k>0 AND k<29 THEN GO SUB
k*100+1000
```

Figura 3

```

230 PRINT AT L,C; INK m;"+";AT
L,C; INK i;CHR$ x
240 GO TO 180
1100 IF c=0 THEN RETURN
1111 LET c=c-1: RETURN
1200 IF c=31 THEN RETURN
1222 LET c=c+1: RETURN
1300 IF l=21 THEN RETURN
1333 LET l=l+1: RETURN
1400 IF l=0 THEN RETURN
1444 LET l=l-1: RETURN
2222 LET i=k-5: RETURN
4444 LET x=k+115: RETURN

```

**Figura 3 (continua)**



**Figura 4**

tasto	colore
0	BLACK
1	BLUE
2	RED
3	MAGENTA
4	GREEN
5	CYAN
6	YELLOW
7	WHITE

tasto	carattere
A	
B	■
C	■
D	■
E	■
F	■
G	■
H	■
I	■
J	■
K	■
L	■
M	■
N	■
O	■
P	■

Tabella 1

## PLOTTER

L'alta risoluzione grafica disponibile sullo Spectrum (256 × 176 pixels) permette di realizzare tramite il programma 'Plotter' (Figura 5) dei disegni molto definiti, tali da essere utilizzati in applicazioni di tipo scientifico e tecnico.

La Figura 6 mostra appunto un disegno tecnico realizzato con 'Plotter'.

Plotter chiede all'inizio di inserire i colori prescelti per il bordo, il fondo e la penna scrivente; dopodichè appare nell'angolo in basso a sinistra un puntino che può essere spostato nelle otto direzioni tramite i tasti 3,4,5,6,7,8,9,0.

I tasti 1 e 2 servono invece per selezionare i modi 'disegno' e 'modifiche e spostamenti'.

Il tasto 'C' permette infine di stampare con la ZX Printer il disegno fatto fino a quel momento.

Fare attenzione che il punto, all'inizio, si vede appena, specialmente se si è selezionato un colore 'penna' molto vicino al colore 'fondo' o al colore 'bordo'.

La funzione dei tasti usati da 'Plotter' è la seguente:

Tasto	Funzione
1	modo 'disegno'
2	modo 'modifiche e spostamenti'
3	in alto a sinistra
4	in basso a sinistra
5	a sinistra
6	in basso
7	in alto
8	a destra
9	in basso a destra
0	in alto a destra
C	'COPY'

```
10 REM ----- PLOTTER -----
20 REM -----
100 INPUT "Inserire colore bord
o (0--7) e premere ENTER"
;b
110 IF b>7 OR b<0 THEN GO TO 10
120 INPUT "Inserire colore fond
o (0--7) e premere ENTER"
;p
130 IF p>7 OR p<0 THEN GO TO 12
140 INPUT "Inserire colore penn
a (0--7) e premere ENTER"
;i
```

Figura 5

```

150 IF i=p OR i>7 OR i<0 THEN G
O TO 140
160 LET x=0: LET y=0: LET d=1
170 LET k=CODE INKEY$-47
177 IF INKEY$="C" THEN COPY
180 IF k>=0 AND k<11 THEN GO SU
B k*100+1000
1900 PLOT x,y
2000 IF d=0 AND POINT (x,y)=1 TH
EN PLOT OVER 1;x,y
2220 GO TO 170
1100 IF x=255 OR y=175 THEN RETU
RN
1111 LET x=x+1: LET y=y+1: RETUR
N
1222 LET d=1: RETURN
1333 LET d=0: RETURN
1400 IF x=0 OR y=175 THEN RETURN
1444 LET x=x-1: LET y=y+1: RETUR
N
1500 IF x=0 OR y=0 THEN RETURN
1555 LET x=x-1: LET y=y-1: RETUR
N
1600 IF x=0 THEN RETURN
1666 LET x=x-1: RETURN
1700 IF y=0 THEN RETURN
1777 LET y=y-1: RETURN
1800 IF y=175 THEN RETURN
1888 LET y=y+1: RETURN
19000 IF x=255 THEN RETURN
20000 LET x=x+1: RETURN
20000 IF x=255 OR y=0 THEN RETURN
2222 LET x=x+1: LET y=y-1: RETUR
N

```

Figura 5 (continua)

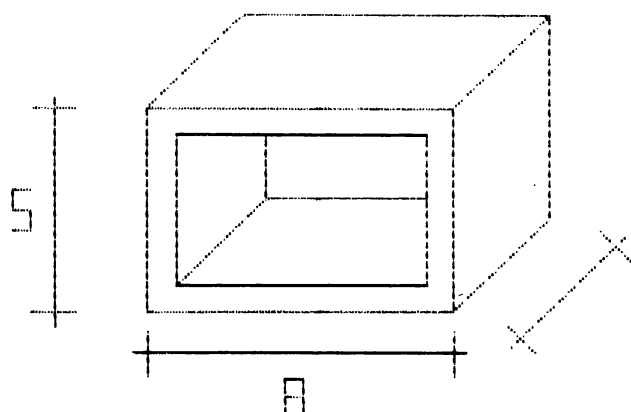


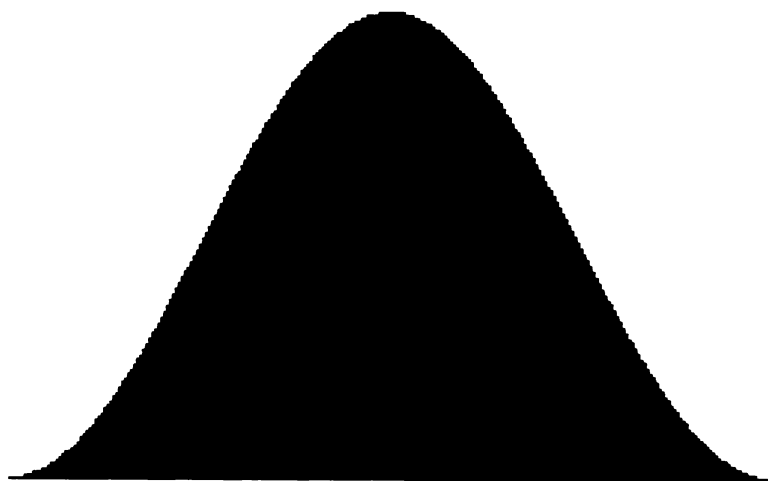
Figura 6

## CURVE MATEMATICHE

Sempre sfruttando l'alta risoluzione grafica, 'Curve Matematiche' (Figura 7) disegna sullo schermo le curve di alcune funzioni matematiche. La Figura 8 mostra una delle quattro curve create dal programma; oltre a queste se ne possono comunque aggiungere molte altre.

```
10 REM CURVE MATEMATICHE
20 REM -----
100 BORDER 0: PAPER 0: CLS
110 INK 6
120 FOR X=0 TO 255
130 PLOT X,X*X/380
140 NEXT X
150 FOR X=0 TO 255
160 PLOT X,X*X/380
170 DRAW 0,-(X*X/380)
180 NEXT X
190 PAUSE 200: INK 5: CLS
200 FOR X=0 TO 255
210 LET Y=RND*X/1.5
220 PLOT X,Y
230 DRAW 0,-Y
240 NEXT X
250 PAUSE 200: INK 3: CLS
260 FOR X=0 TO 255
270 PLOT X,LN (X/5+1)*40
280 NEXT X
290 FOR X=0 TO 255
300 LET Y=LN (X/5+1)*40
310 PLOT X,Y
320 DRAW 0,-Y
330 NEXT X
340 PAUSE 200: INK 4: CLS
350 FOR X=0 TO 255
360 PLOT X,80+COS (PI/180*(180+
X*1.41))*80
370 NEXT X
380 FOR X=0 TO 255
390 LET Y=80+COS (PI/180*(180+X
*1.41))*80
400 PLOT X,Y
410 DRAW 0,-Y
420 NEXT X
```

Figura 7



**Figura 8**

## DISEGNO POLIGONI

Il disegnatore di poligoni (Figura 9) è una routine grafica che può essere utilizzata appunto per creare uno o più poligoni a partire da quattro valori numerici, vale a dire, le coordinate X e Y del centro, il raggio R e il numero dei lati NL di ciascun poligono (Figura 10).

Un esempio di poligono a nove lati è in Figura 11.

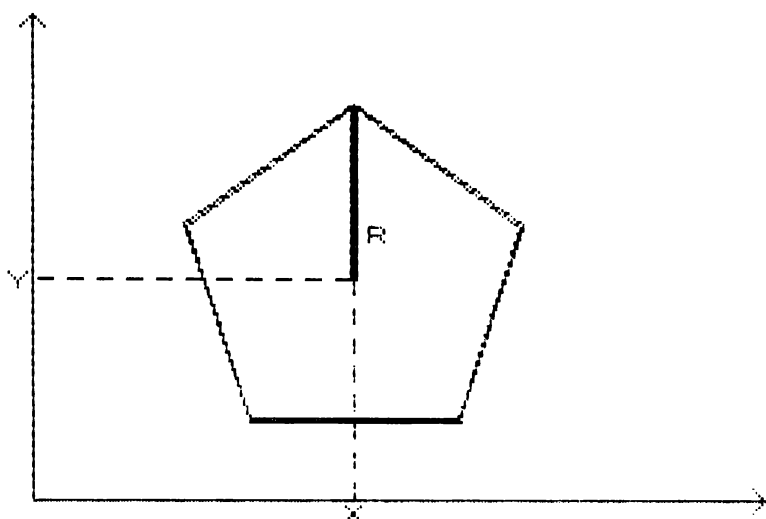
Sostituendo al programma di Figura 9 le linee da 99 a 999 con le nuove linee di Figura 12 si può creare sul video un'immagine dimostrativa di sei poligoni concentrici di vari colori e con un numero di lati da tre a otto (Figura 13).

```

99  BORDER 1: PAPER 0: INK 7: C
LS
100 REM  X,Y =coordinate centro
110 INPUT "Inserire X " ;X
120 INPUT "Inserire Y " ;Y
130 REM  R = raggio
140 INPUT "Inserire R " ;r
150 REM  NL = numero lati
160 INPUT "Inserire NL " ;nl
170 GO SUB 4000
180 STOP
999 REM  -----

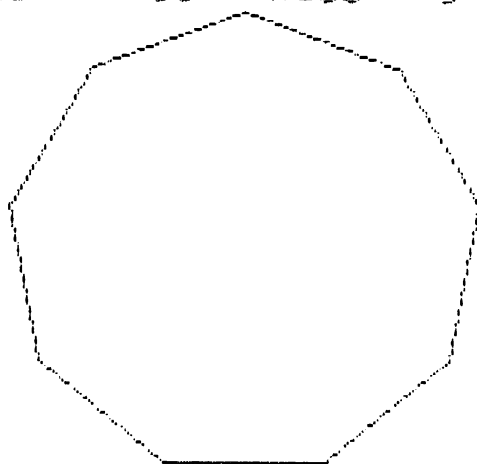
4000 REM  Disegno POLIGONI
4010 LET cx=x: LET cy=y+r
4030 FOR p=90 TO 451 STEP (360/nl)
4040 LET j=PI/180*p
4050 LET hx=r*COS j+x
4060 LET hy=r*SIN j+y
4070 PLOT cx,cy
4080 DRAW hx-cx,hy-cy
4090 LET cx=hx: LET cy=hy
4100 NEXT p: RETURN
```

Figura 9



**Figura 10**

$X=128$      $Y=65$      $R=60$      $\theta$  lati



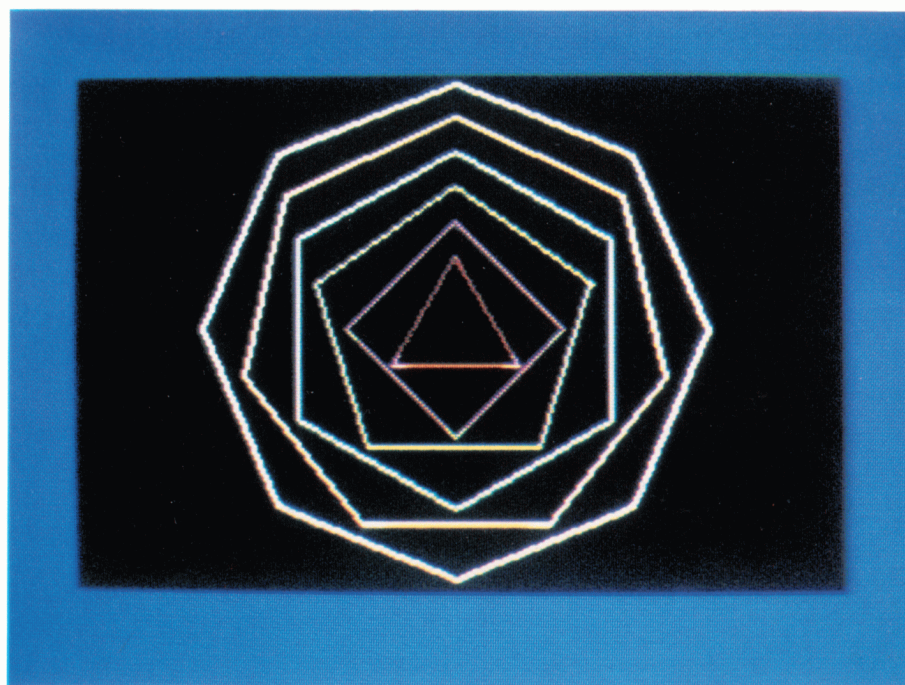
**Figura 11**

```

10 REM programma dimostrativo
99 BORDER 1: PAPER 0: CLS
100 LET x=128: LET y=88
110 FOR i=2 TO 7
120 INK i
130 LET nl=i+1
140 LET r=i*12
150 GO SUB 4000: NEXT i
160 STOP
999 REM -----

```

**Figura 12**



**Figura\_13**

## PLOTTAGGIO ELLISSI

Un'altra interessante routine grafica (utilizzata anche in altri programmi di questo libro) è il 'Plottaggio Ellissi' (Figura 14).

Con essa si possono disegnare una o più ellissi, verticali o orizzontali, a partire dalle coordinate X e Y del centro e dai raggi maggiore 'RMA' e minore 'RMI' di ciascuna (Figura 15).

Due ellissi con i relativi dati sono in Figura 16.

```

99 BORDER 1: PAPER 0: INK 7: C
LS
100 REM X,Y = coordinate centro
110 INPUT "Inserire X ";X
120 INPUT "Inserire Y ";Y
130 REM RMA = raggio maggiore
140 INPUT "Inserire RMA ";rma
150 REM RMI = raggio minore
160 INPUT "Inserire RMI ";rmi
170 GOSUB 3000
180 STOP
999 REM -----
3000 REM Plottaggio ELLISSI
3010 IF rma>rmi THEN LET st=rma
3020 IF rmi>rma THEN LET st=rmi
3030 FOR p=0 TO 360 STEP 0.4+INT
(30/st)
3040 LET rd=p+PI/180
3050 PLOT rma*(COS rd)+X,rmi*(SI
N rd)+Y
3060 NEXT p: RETURN
    
```

Figura 14

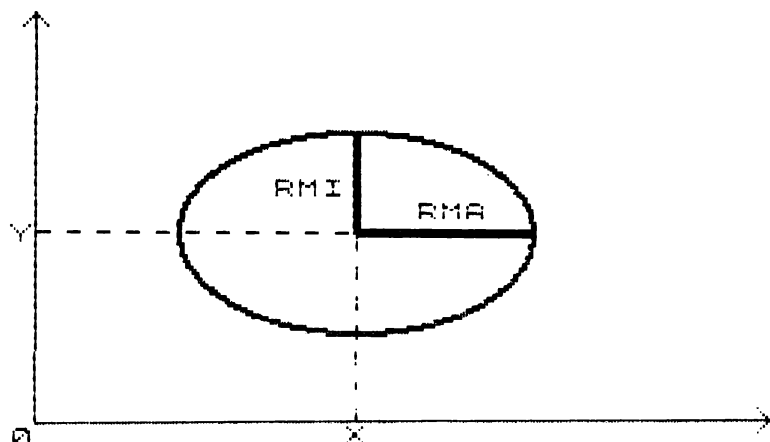


Figura 15



## DISEGNO QUADRATI

Ultima routine grafica di uso generale è il 'Disegno Quadrati' (Figura 17) per disegnare dei quadrati a partire dalle coordinate X e Y dello spigolo in basso a sinistra e dalla lunghezza del lato L (Figura 18).

```

99 BORDER 1: PAPER 0: INK 4: C
LS
100 REM X,Y = coordinate angolo
110 INPUT "Inserire X ";X
120 INPUT "Inserire Y ";Y
130 REM L = lato
140 INPUT "Inserire L ";L
150 GO SUB 2000
160 STOP
999 REM -----
2000 REM Disegno QUADRATI
20010 PLOT X,Y
20020 DRAW 0,L-1
20030 DRAW L-1,0
20040 DRAW 0,-(L-1)
20050 DRAW -(L-1),0
20060 RETURN
```

Figura 17

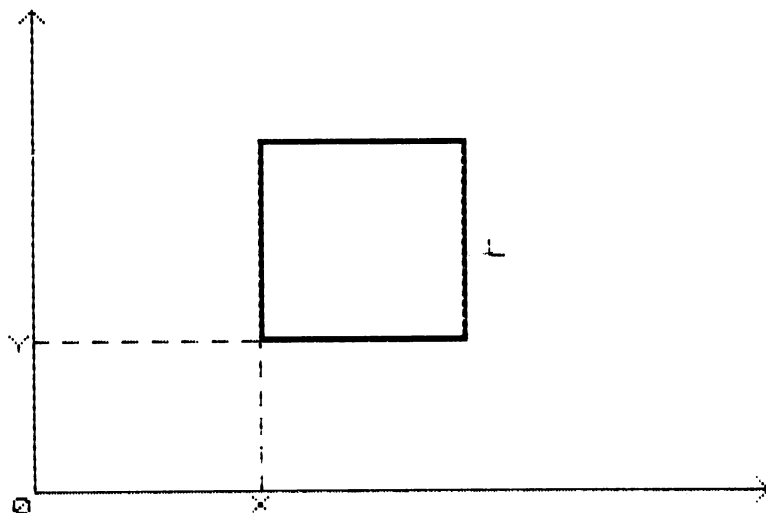


Figura 18

## SPIRALI

Il programma di Figura 19 disegna un numero prestabilito (ns) di spirali a partire dal centro di coordinate x, y e con una determinata larghezza (l). Un esempio di otto spirali è in Figura 20.

```
10 REM SPIRALI
20 REM -----
99 BORDER 1: PAPER 0: INK 6: C
L3
100 REM x,y =coordinate centro
110 LET x=128: LET y=68
120 REM ns = numero spirali
130 LET ns=8
140 REM l = larghezza spirale
150 LET l=10
200 LET st=l/180
210 FOR p=0 TO 360*ns STEP 2
220 LET j=PI/180*p
230 PLOT st*COS j+x,st*SIN j+y
240 LET st=st+l/180
250 NEXT p
```

Figura 19

x=128      y=68      l=10      ns=8

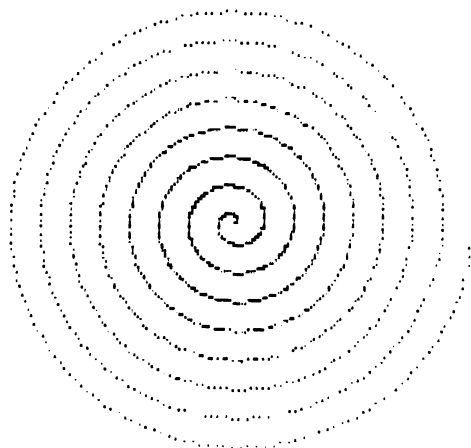


Figura 20

## CONO TRIDIMENSIONALE

I dati X,Y,RMA,RMI,H all'interno del programma di Figura 21 che disegna un cono tridimensionale (Figura 22), possono essere variati a piacere per disegnare coni con basi e/o altezze diverse.

```
10 REM CONO TRIDIMENSIONALE
20 REM -----
99 BORDER 1: PAPER 0: INK 4: C
LS
100 REM X,Y = coordinate base
110 LET x=120: LET y=40
120 REM RMA = raggio maggiore
130 REM RMI = raggio minore
140 LET rma=60: LET rmi=25
150 REM H = altezza
160 LET h=120
170 GO SUB 1000
180 STOP
1000 IF rma>rmi THEN LET st=rma
1010 IF rmi>rma THEN LET st=rmi
```

Figura 21 (continua)

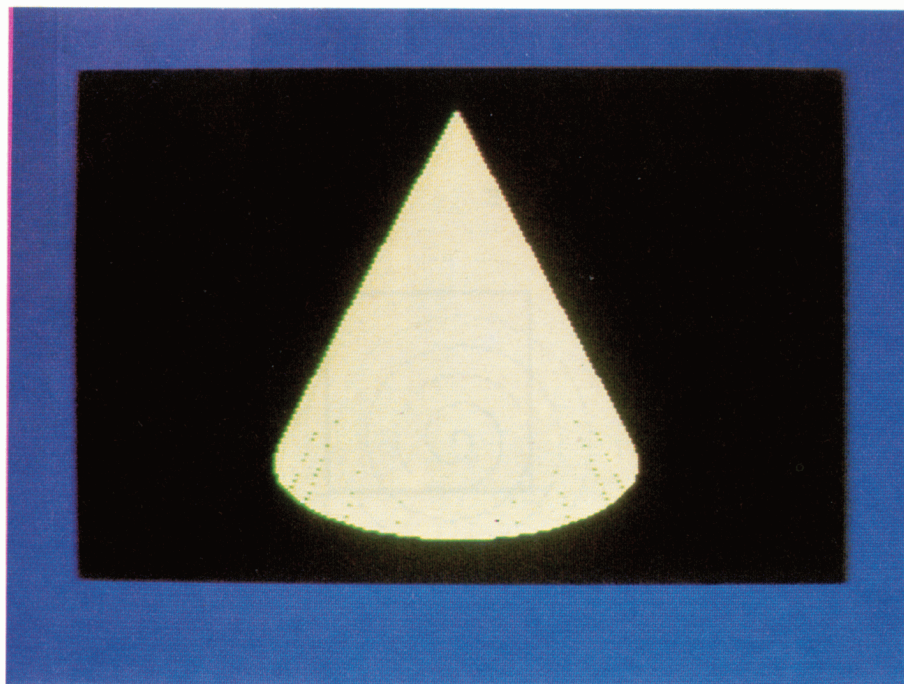


Figura 22

```

1020 FOR p=0 TO 360 STEP 0.2+INT
(30/st)
1030 LET rd=p*PI/180
1040 LET a=rma*(COS rd)+x
1050 LET b=rmi*(SIN rd)+y
1060 PLOT a,b
1070 IF p<=30 OR p>=150 THEN GO
TO 1090
1080 NEXT p: RETURN
1090 DRAW -(a-x),h-1-b+y
1100 GO TO 1080

```

**Figura 21**

## **CILINDRO TRIDIMENSIONALE**

Anche nel programma di Figura 23 che disegna un cilindro tridimensionale (Figura 24), modificando i dati X,Y,RMA,RMI,H, si possono disegnare dei cilindri di diverse grandezze. Una variante di questo programma è usata in uno dei programmi di Business Grafica per visualizzare dei dati.

```

10 REM CILINDRO
20 REM TRIDIMENSIONALE
30 REM -----
99 BORDER 1: PAPER 0: INK 6: C
LS
100 REM X,Y = coordinate base ↑
110 LET x=128: LET y=128
120 REM RMA = raggio maggiore
130 REM RMI = raggio minore
140 LET rma=80: LET rmi=40
150 REM H = altezza
160 LET h=80
170 GO SUB 1000
180 STOP
1000 IF rma>rmi THEN LET st=rma
1010 IF rmi>rma THEN LET st=rmi
1020 FOR p=0 TO 360 STEP 0.4+INT
(30/st)
1030 LET rd=p*PI/180
1040 LET a=rma*(COS rd)+x
1050 LET b=rmi*(SIN rd)+y
1060 PLOT a,b
1070 IF p>=180 THEN GO TO 1090
1080 NEXT p: RETURN
1090 DRAW 0, -(h-1)
1100 GO TO 1060

```

**Figura 23**

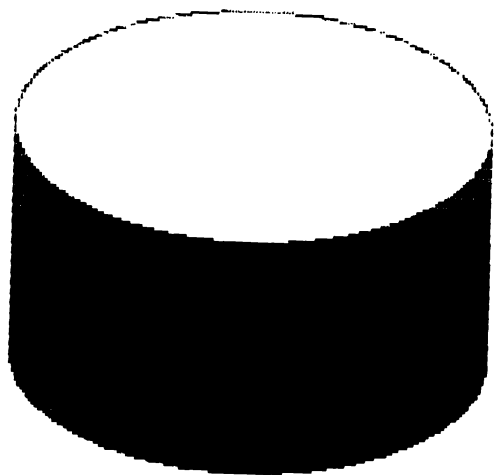


Figura 24

## SFERA TERRESTRE

Il programma di Figura 25 disegna, in modo molto bello a vedersi, una sfera terrestre (Figura 26) intrecciando varie ellissi per ottenere le principali latitudini e longitudini della terra.

```

10 REM SFERA TERRESTRE
20 REM -----
99 BORDER 0: PAPER 0: INK 5: C
LS
100 LET x=128: LET y=88
110 LET rma=85: LET rmi=85
120 GO SUB 1000
122 LET rmi=23: GO SUB 1000
124 LET rma=74: LET rmi=85
126 GO SUB 1000
128 LET rma=43: GO SUB 1000
130 LET y=120

```

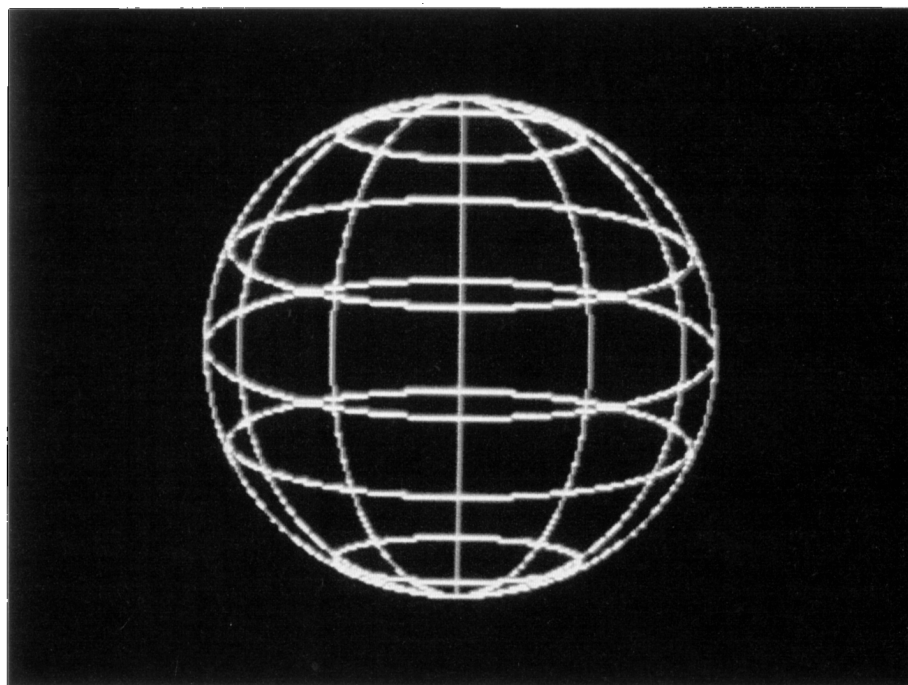
Figura 25 (continua)

```

140 LET rma=77: LET rmi=18
150 GO SUB 1000
152 LET y=56
154 GO SUB 1000
160 LET y=160
170 LET rma=41: LET rmi=8
180 GO SUB 1000
190 LET y=16: GO SUB 1000
200 PLOT 128,3: DRAW 0.170
9999 STOP
1000 FOR p=0 TO 360 STEP 0.4
1010 LET rd=p*PI/180
1020 PLOT rma*COS rd+x,rmi*SIN r
d+y
1030 NEXT p: RETURN

```

**Figura 25**



**Figura 26**

## PROGRAMMI DI BUSINESS GRAFICA

I prossimi nove programmi introducono alla cosiddetta 'Business Grafica' vale a dire la rappresentazione in forme grafiche diverse di dati economici, statistici o anche scientifici, tecnici, matematici, ecc.

Nonostante lo Spectrum ponga alcune limitazioni di carattere grafico il risultato è comunque visivamente molto efficace.

Tutti i programmi sono divisi in due parti fondamentali: la parte 'Dati' che contiene appunto tutti i dati numerici ed alfanumerici relativi al problema trattato, con i valori massimi consentiti e la parte 'Programma' che controlla la correttezza dei dati e li trasforma nella figura grafica scelta. È importante tenere conto delle note indicate per i vari programmi.

TUTTI I DATI, I NOMI, LE CIFRE, ECC.. USATI NEGLI ESEMPI SONO PARZIALMENTE O COMPLETAMENTE INVENTATI E SERVONO A MOSTRARE IL FUNZIONAMENTO DEI VARI PROGRAMMI.

Naturalmente è possibile utilizzare i grafici per visualizzare qualsiasi tipo di dati reali, come anche è possibile cambiare i colori impiegati negli esempi.

### DIAGRAMMA A TORTA

È certo il più classico modo di rappresentare i dati nella Business Grafica. Il programma è in Figura 27.

I dati vengono rappresentati in forma di fette (da 2 a 5) di una torta (Figura 28) che costituisce il 100%.

Le percentuali delle varie fette sono contenute nelle variabili  $f(1) \dots f(5)$  il cui totale deve essere sempre 100, la natura di tali percentuali è invece contenuta nelle variabili stringa  $P\$(1) \dots p\$(5)$ .

A causa delle limitazioni di cui si è detto nell'introduzione, in questo programma le fette adiacenti devono essere minimo di 2-3%.

```
10 REM          Business Grafica
20 REM          DIAGRAMMA A TORTA
30 REM          -----
```

Figura 27 (continua)

```

33 BORDER 0: PAPER 0: INK 7: C
LS
44 LET x=80: LET y=80
55 CIRCLE INK 4:x,y,80
1000 REM Dati
110 REM a$=max 64 caratteri
120 LET a$="COMPUTER PRODUCTS s
.r.l. Milano Percentuali di spes
a anno 1982 "
130 DIM f(5)
140 REM f(1)--f(5)=larghezza
145 REM segmenti torta in %
150 REM 5--5 segmenti
160 REM totale 100% esatto !!
165 LET f(1)=15.7
170 LET f(2)=20.5
180 LET f(3)=9.5
190 LET f(4)=30
200 LET f(5)=10.3
205 DIM p$(5,9)
210 REM p$1-5=max 9 caratteri
220 LET p$(1)="Pag"
230 LET p$(2)="Imposte"
240 LET p$(3)="Varie"
250 LET p$(4)="Materiali"
260 LET p$(5)="Stipendi"
3000 REM Programma
310 IF LEN a$>64 THEN LET a$=a$
(1 TO 64)
320 PRINT AT 0,0: INK 7: INVERS
E 330 a$
340 LET ns=0
350 FOR a=1 TO 5
360 IF f(a)>0 THEN LET ns=ns+1
370 NEXT a
380 LET j=0
390 FOR a=1 TO 5
400 LET j=j+f(a)
410 NEXT a
420 IF INT j<>100 THEN GO TO 20
0 0
430 DIM r(ns)
440 FOR a=1 TO ns
450 LET r(a)=PI/50*f(a)
460 NEXT a
470 LET k=0
480 FOR a=1 TO ns
490 LET k=k+r(a)
500 PLOT INK 4:x,y: DRAW INK 4:
600 CIRCLE k,80*6IN k
510 LET w=k-(r(a)/2)
520 LET z=x+80*COS w
530 LET t=y+80*6IN w
540 PLOT INK 4:z,t
550 LET zz=10*COS w
560 LET tt=10*6IN w

```

Figura 27 (continua)

```

670 DRAW INK 4;ZZ,tt
680 LET t=21.5-INT (t+tt+1)/8
690 LET c=0.5+INT (z+zz+1)/8-1
700 PRINT AT t,c: INK a+2;"■"
710 NEXT a
720 FOR a=1 TO ns
730 LET d$=STR$ t(a)
740 IF LEN d$>4 THEN LET d$=d$(
1 TO 4)
750 PRINT AT a*3+1,21: INK a+2;
"■"+d$+"%"
760 PRINT AT a*3+2,23: INK a+2;
P$(a)
770 NEXT a
1000 STOP
2000 PRINT AT 5,5," VALORI ERRE
II"

```

Figura 27

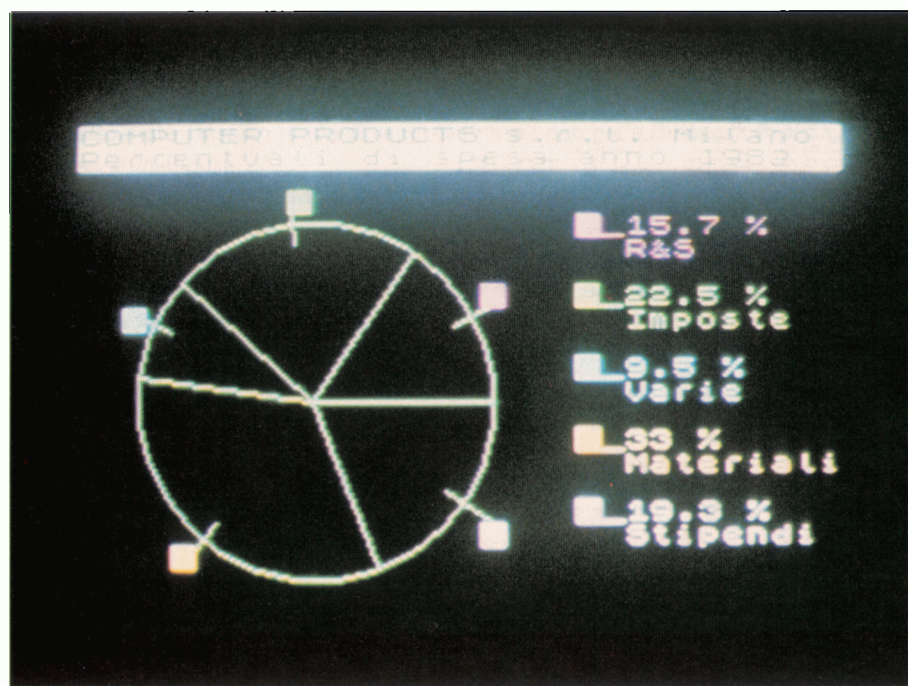


Figura 28

## CILINDRI 3 D

Molto bella è in questo programma (Figura 29) la rappresentazione di quattro dati in forma di cilindri tridimensionali di diversa altezza e colore (Figura 30).

L'altezza dei cilindri è inserita nelle variabili h(1)...h(4).

```

10 REM          Business Grafica
20 REM          CILINDRI 3 D
30 REM          -----
33 BORDER 0: PAPER 0: INK 7: C
LS
44 LET rmi=11: LET rma=23
77 FOR a=16 TO 19
86 FOR b=0 TO 31
99 PRINT AT a,b: PAPER 1;" ":
NEXT b: NEXT a
100 REM          Dati
110 REM          a$ max 64 caratteri
120 LET a$="Proiezioni vendite
Computers      in Italia (pezzi x1
0000)
122 DIM h(4)
130 REM          h(1) -- h(4) max 100
140 LET h(1)=24
150 LET h(2)=42
160 LET h(3)=64
170 LET h(4)=88
177 REM          u$ 1-4 max 4 caratteri
180 DIM u$(4,4)
190 LET u$(1)="1000"
200 LET u$(2)="1004"
210 LET u$(3)="1005"
220 LET u$(4)="1006"
230 REM          n$ 1-4 max 5 numeri
240 DIM n$(4,5)
250 LET n$(1)="24"
260 LET n$(2)="42"
270 LET n$(3)="64"
280 LET n$(4)="88"
290 LET i$="56744662"
300 REM          Programma
310 PRINT AT 0,0: INK 7: INVERS
E 1): a$
320 FOR c=0 TO 3
330 PRINT AT 20,c*7+3: INK 3: u$
(c+1)
340 NEXT c
350 FOR c=0 TO 3
365 IF h(c+1)<0 OR h(c+1)>100 T
HEN GO TO 2000
370 LET x=40+c*56
370 LET y=136-(100-h(c+1))
380 LET h=h(c+1)

```

Figura 29 (continua)

```

580 LET i=VAL i$(c+1)
590 GO SUB 1000
600 NEXT c
610 FOR c=0 TO 3
620 IF h(c+1)>93 THEN PRINT AT
2,2+c*7; INK VAL i$(c+5);n$(c+1)
630 IF h(c+1)<94 THEN PRINT AT
21-(h(c+1)/8+7),2+c*7; INK VAL i
$(c+5);n$(c+1)
640 NEXT c
9990 STOP
1000 FOR p=0 TO 360 STEP 2
1030 LET rd=p*PI/180
1040 LET a=rma*(COS rd)+x
1050 LET b=rmi*(SIN rd)+y
1060 PLOT INK i;s.b
1070 IF p>=180 THEN GO TO 1090
1080 NEXT p: RETURN
1090 DRAW INK i;0,-(h-1)
1100 GO TO 1030
2000 PRINT AT 5,5;" VALORI ERRE
TI"

```

Figura 29

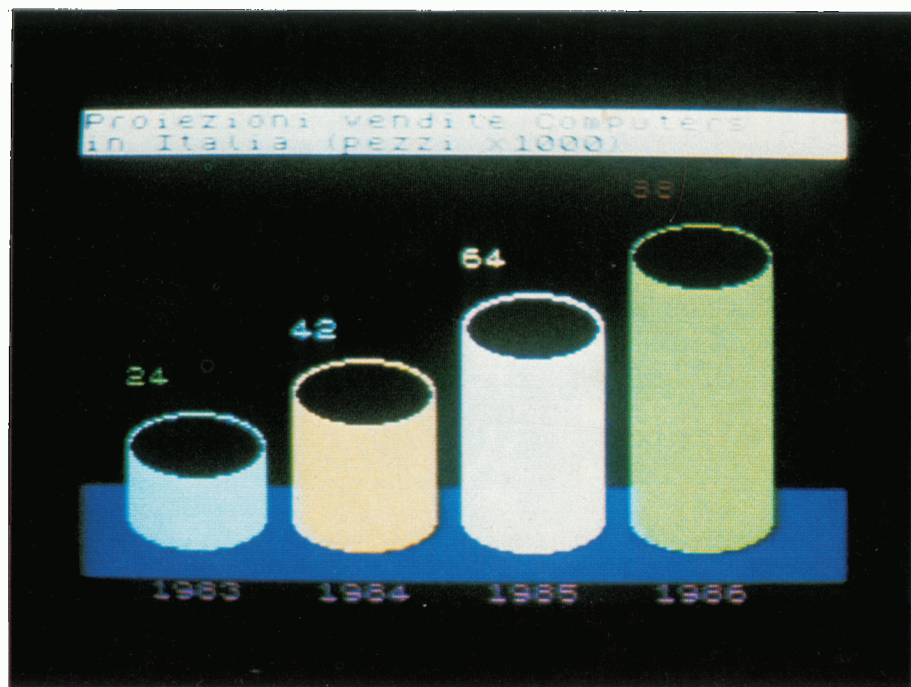


Figura 30

## ISTOGRAMMI ORIZZONTALI

Per usare questo programma (Figura 31) occorre stabilire il valore massimo 'r' della scala (linea 230) e quindi inserire nelle variabili d(1)...d(8) dei valori reali compresi tra 0 ed r.

Da ciò si ottiene un grafico come quello di Figura 32.

```

10 REM      Business Grafica
20 REM      ISTOGRAMMI ORIZZONTALI
30 REM      -----
99 BORDER 0: PAPER 0: CL3
100 INK 6: PLOT 0,24
110 DRAW 255,0
120 FOR a=79 TO 240 STEP 8
130 PLOT a,16: DRAW 0,135
140 NEXT a
150 LET U=PEEK 23675+256*PEEK 2
3676
155 FOR a=0 TO 5
160 FOR p=0 TO 7
165 POKE U+p+a*8,255-2*(7-a)
170 NEXT p: NEXT a
180 FOR a=9 TO 30 STEP 5
182 PRINT AT 20,a: INVERSE 1;CH
R$ 150
184 NEXT a
190 INK 7
2000 REM      Dati
2010 REM      a$ max 64 caratteri
2020 LET a$="Reddito pro capite
10001 (Milioni lire Itali
2120
2230 LET r=20
2240 REM      r=valore massimo scala
2250 REM      valori per r:
2260 REM      10 20 40 100 200 400
2270 DIM d(8)
2280 REM      valori d(1)--d(8) tra
2290 REM      0 e il valore max "r"
2300 LET d(1)=10.5
2310 LET d(2)=9.7
2320 LET d(3)=8.1
2330 LET d(4)=4.8
2340 LET d(5)=5.4
2350 LET d(6)=3.2
2360 LET d(7)=10.3
2370 LET d(8)=1.2
2380 DIM c$(8,9)
2390 REM      c$1-9 max 9 caratteri
4000 LET c$(1)="U.S.A."
410 LET c$(2)="Giappone"
420 LET c$(3)="Germania"
430 LET c$(4)="Francia"

```

Figura 31 (continua)

```

440 LET C$(5) = "Inghilt. _"
450 LET C$(6) = "Italia _"
460 LET C$(7) = "Canada _"
470 LET C$(8) = "Grecia _"
480 REM
490 REM
500 REM
510 IF LEN A$ > 64 THEN LET A$ = A$
(1 TO 64)
520 PRINT AT 0,0; INVERSE 1; A$
530 BRIGHT 1; INK 2; PRINT AT 2
1,9;"0"; AT 21,14;r/4; AT 21,19;r/
2; AT 21,24;r/4*3; AT 21,29;r
540 BRIGHT 0; INK 7
550 FOR a=1 TO 8
560 IF d(a) < 0 OR d(a) > r THEN GO
TO 2000
570 NEXT a
580 FOR l=17 TO 3 STEP -2
590 LET k=(l-1)/2
600 PRINT AT l,0; INK 4; C$(k)
610 FOR c=10 TO 8+INT (d(k)/r/
20))
620 PRINT AT l,c; INK 5; BRIGHT
1;"■"

```

Figura 31 (continua)

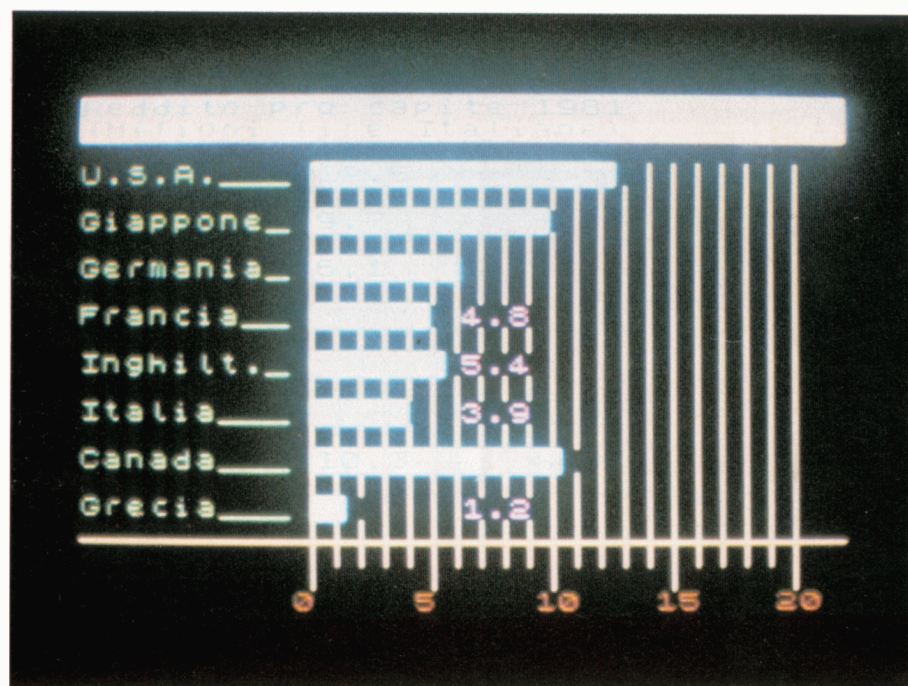


Figura 32

```

622 NEXT c
630 LET n=((d(k)/(r/20))-INT (d
(k)/(r/20))) *8
640 IF n>=1 THEN PRINT AT L,c)
INK 5; BRIGHT 1;CHR$ (143+n)
644 LET X$=STR$ d(k); IF LEN X$
>4 THEN LET X$=X$(1 TO 4)
650 IF c>15 THEN PRINT AT L,10;
BRIGHT 1; PAPER 5; INK 0;X$
660 IF c<16 THEN PRINT AT L,16;
BRIGHT 1; PAPER 0; INK 3;X$
700 NEXT L
1000 STOP
2000 PRINT AT 5,5;" VALORI ERRE
TI"

```

Figura 31

## ISTOGRAMMI VERTICALI

La sistemazione verticale degli istogrammi permette di visualizzare con questo programma (Figura 33) dodici dati diversi (nel modo indicato in Figura 34), dati i cui valori vanno inseriti (dopo aver stabilito 'r') nelle variabili d(1)....d(12).

```

10 REM Business Grafica
20 REM ISTOGRAMMI VERTICALI
30 REM -----
99 BORDER 0: PAPER 0: CLS
100 INK 2: PLOT 47,24
110 DRAW 0,127
120 FOR a=23 TO 153 STEP 8
130 PLOT 40,a: DRAW 199,0
140 NEXT a
150 LET U=PEEK 23675+256*PEEK 2
3675
155 FOR a=0 TO 6
160 FOR p=0 TO 7
165 POKE U+p+a*8,0
166 IF a<p THEN POKE U+p+a*8,25
5
170 NEXT p: NEXT a
180 FOR a=19 TO 4 STEP -5
182 PRINT AT a,4, INVERSE 1;CHR
$ 144
184 NEXT a
190 INK 7
200 REM Dati

```

Figura 33 (continua)

```

0010 REM      a$=max 64 caratteri
0020 LET      a$="Consumo medio elett
ricita' in Italia (in GWh)

0030 LET      r=30
0040 REM      r=valore massimo scala
0050 REM      e valori per r :
0060 REM      e 15 30 60 150 300 600
0070 REM      d(10)
0080 REM      valori d(1) -- d(12) tra
0090 REM      0 e 1: valore max "r"
0100 LET      d(1)=0
0110 LET      d(2)=0.5
0120 LET      d(3)=0.5
0130 LET      d(4)=0.5
0140 LET      d(5)=0.5
0150 LET      d(6)=0.5
0160 LET      d(7)=0.5
0170 LET      d(8)=0.5
0180 LET      d(9)=0.5
0190 LET      d(10)=0.5
0200 LET      d(11)=0.5
0210 LET      d(12)=0.5
0220 REM      c$(1) = 30
0230 REM      c$(1-12) = max 3 caratteri
0240 LET      c$(1) = "30"
0250 LET      c$(2) = "60"
0260 LET      c$(3) = "150"
0270 LET      c$(4) = "300"
0280 LET      c$(5) = "600"
0290 LET      c$(6) = "1200"
0300 LET      c$(7) = "2400"
0310 LET      c$(8) = "4800"
0320 LET      c$(9) = "9600"
0330 LET      c$(10) = "19200"
0340 LET      c$(11) = "38400"
0350 LET      c$(12) = "76800"
0360 REM
0370 REM      Programma
0380 IF LEN a$ > 64 THEN LET a$ = a$
(1) TO 64)
0390 PRINT AT 0,0: INVERSE 1: a$
0400 BRIGHT 1: INK 4: PRINT AT 1
0410 "0"
0420 PRINT AT 4,4-LEN STR$ c(r)
0430 PRINT AT 9,4-LEN STR$ (r/3*
2) (r/3*2)
0440 PRINT AT 14,4-LEN STR$ (r/3
1) (r/3)
0450 BRIGHT 0: INK 7
0460 FOR a=1 TO 12
0470 IF d(a) < 0 OR d(a) > r THEN GO
TO 2000
0480 NEXT a
0490 FOR c=7 TO 29 STEP 2
0500 LET k=(c-5)/2

```

Figura 33 (continua)

```

600 PRINT AT 19,c; INK 5;c$(k,1
)
602 PRINT AT 20,c; INK 5;c$(k,2
)
604 PRINT AT 21,c; INK 5;c$(k,3
)
610 FOR l=18 TO 19-INT (d(k)/(r
/15)) STEP -1
620 PRINT AT l,c; INK 6;"■"
622 NEXT l
630 LET n=((d(k)/(r/15))-INT (d
(k)/(r/15)))*8
640 IF n>=1 THEN PRINT AT l,c;
INK 6;CHR$ (143+(8-n))
650 NEXT c
1000 STOP
2000 PRINT AT 5,5;" VALORI ERRO
RI"

```

Figura 33

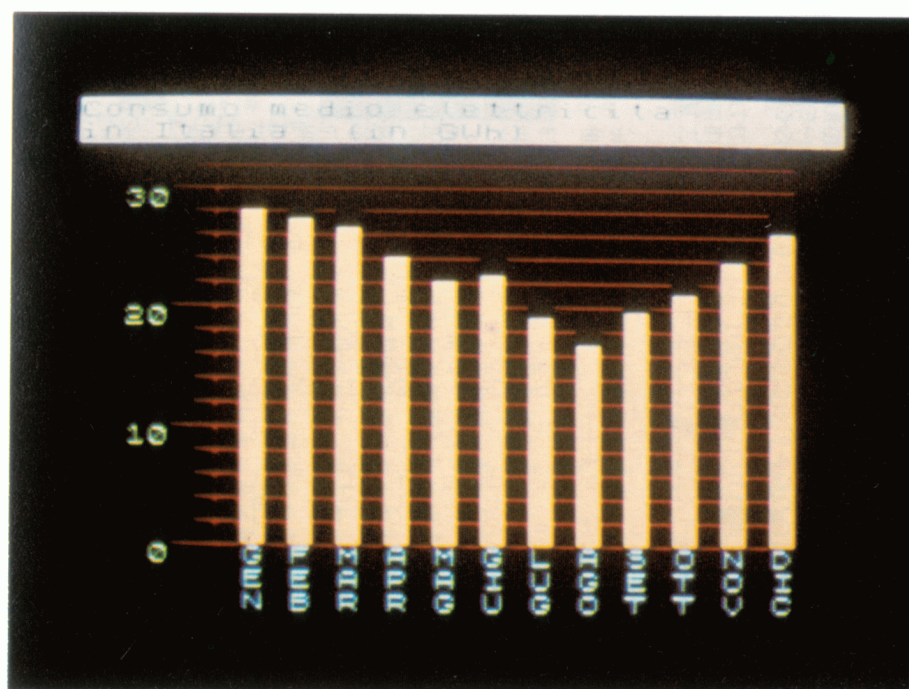


Figura 34

## ISTOGRAMMI VERTICALI A COLORI AFFIANCATI

Questo è il primo di tre programmi che visualizzano e paragonano tra loro dati provenienti da fonti diverse ma della stessa natura.

Anche in questo programma (Figura 35) va stabilito 'r' tra i valori indicati e quindi vanno inseriti i dati degli istogrammi blu, verdi e azzurri rispettivamente nelle variabili d(1)...(4), e(1)...(4), f(1)...(4). Tutti i dati inseriti in tali variabili devono avere valori arrotondati a causa di alcune limitazioni grafiche. Nell'esempio di Figura 36 i valori potranno essere per esempio: 80, 90, 110, 140, ecc...

Le stringhe j\$, k\$, w\$ contengono gli argomenti a cui si riferiscono rispettivamente gli istogrammi blu, verdi e azzurri.

```

10 REM Business Grafica
20 REM ISTOGRAMMI VERTICALI
30 REM A COLORI AFFIANCATI
40 REM -----
99 BORDER 0: PAPER 0: CLS
100 INK 2: PLOT 47,16
110 DRAW 0,119
120 FOR a=15 TO 138 STEP 8
130 PLOT 40,a: DRAW 215,0
140 NEXT a
150 FOR a=15 TO 138 STEP 40
160 PLOT 32,a: DRAW 8,0
170 NEXT a
180 INK 7
200 REM
210 REM
220 REM
230 REM
240 REM
250 REM
260 REM
270 REM
280 REM
290 REM
300 REM
310 REM
320 REM
330 REM
340 REM
350 REM
360 REM
370 REM
380 REM
390 REM
400 REM
410 REM
420 REM
430 REM
440 REM
450 REM
460 REM
470 REM
480 REM
490 REM
500 REM
510 REM
520 REM
530 REM
540 REM
550 REM
560 REM
570 REM
580 REM
590 REM
600 REM
610 REM
620 REM
630 REM
640 REM
650 REM
660 REM
670 REM
680 REM
690 REM
700 REM
710 REM
720 REM
730 REM
740 REM
750 REM
760 REM
770 REM
780 REM
790 REM
800 REM
810 REM
820 REM
830 REM
840 REM
850 REM
860 REM
870 REM
880 REM
890 REM
900 REM
910 REM
920 REM
930 REM
940 REM
950 REM
960 REM
970 REM
980 REM
990 REM

```

Figura 35 (continua)

```

372 LET f(2)=30
374 LET f(3)=40
376 LET f(4)=50
380 DIM c$(4,5)
390 REM c$1-4 max 5 caratteri
400 LET c$(1)="1979"
410 LET c$(2)="1980"
420 LET c$(3)="1981"
430 LET c$(4)="1982"
499 REM
500 REM
502 IF LEN j$>8 THEN LET j#=j$(
1 TO 8)
504 IF LEN k$>8 THEN LET k#=k$(
1 TO 8)
506 IF LEN w$>8 THEN LET w#=w$(
1 TO 8)
510 IF LEN a$>64 THEN LET a#=a$
(1 TO 64)
520 PRINT AT 0,0; INVERSE 1; a$
522 INK 6; PRINT AT 20,3;"0"
530 PRINT AT 5,4-LEN STR$ r/r
533 PRINT AT 10,4-LEN STR$ (r/3
*2); (r/3*2)
535 PRINT AT 15,4-LEN STR$ (r/3
); (r/3)
540 INK 7
542 PRINT AT 3,0; INK 1;" "; P
APER 7; j$; PAPER 0; INK 4; AT 3,1
1;" "; k$; INK 5; AT 3,2;" "; w$
550 FOR a=1 TO 4
560 IF d(a)<0 OR d(a)>r THEN GO
TO 2000
562 IF e(a)<0 OR e(a)>r THEN GO
TO 2000
564 IF f(a)<0 OR f(a)>r THEN GO
TO 2000
570 NEXT a
580 FOR c=7 TO 25 STEP 6
590 LET k=(c-1)/6
600 PRINT AT 21,c; c$(k)
610 FOR l=19 TO 20-INT (d(k)/r
/15); STEP -1
620 PRINT AT l,c; INK 1;" "
630 NEXT l
640 FOR l=19 TO 20-INT (e(k)/r
/15); STEP -1
650 PRINT AT l,c+1; INK 4;" "
660 NEXT l
670 FOR l=19 TO 20-INT (f(k)/r
/15); STEP -1
680 PRINT AT l,c+2; INK 5;" "
690 NEXT l: NEXT c
1000 STOP
2000 PRINT AT 5,5;" VALORI ERRO
RI "

```

Figura 35



**Figura 36**

## ISTOGRAMMI VERTICALI A COLORI SOVRAPPOSTI

In questo programma (Figura 37) i dati vengono confrontati tra loro sovrapponendo in blocchi i vari valori.

Come nel programma precedente i dati inseriti nelle variabili devono avere valori arrotondati ed in più la somma dei dati di un singolo istogramma non deve superare il valore di 'r' stabilito precedentemente nella linea 230. Per esempio con  $r=30$  la somma di  $d(3)$ ,  $e(3)$ ,  $f(3)$  non deve superare appunto il valore di 30.

Ai blocchi blu, rossi, gialli corrispondono rispettivamente i dati delle variabili  $d(1)...(5)$ ,  $e(1)...(5)$ ,  $f(1)...(5)$  e gli argomenti contenuti nelle stringhe  $j\$$ ,  $k\$$ ,  $w\$$ .

Con i dati esempio inseriti nel programma si ottiene l'immagine di Figura 38.

```

10 REM Business Grafica
20 REM ISTOGRAMMI VERTICALI
30 REM A COLORI SOVRAPPOSTI
40 REM -----
50 BORDER 0: PAPER 0: CLS
100 INK 3: PLOT 47,16
110 DRAW 0,119
120 FOR a=15 TO 135 STEP 5
130 PLOT 40,a: DRAW 207,0
140 NEXT a
150 FOR a=15 TO 135 STEP 40
160 PLOT 32,a: DRAW 5,0
170 NEXT a
180 INK 7
2000 REM Dati
2010 REM a$=max 24 caratteri
2020 LET a$="Produzione mondiale
2030 LET a$="in Milioni di pezzi
2040 REM i)
2050 LET c$="U.S.A."
2060 LET k$="Europa"
2070 LET w$="Giappone"
2080 REM c$ k$ w$ max 8 caratt.
2090 LET r=30
2100 REM r=valore massimo scala
2110 REM r=valori per n:
2120 REM 0 10 20 30 40 50 60 70
2130 DIM d(5)
2140 DIM e(5)
2150 DIM f(5)
2160 REM Val. d+e+f(1)-(5) tra
2170 LET a=1: valore max "r"
2180 LET d(1)=5
2190 LET d(2)=14
2200 LET d(3)=10
2210 LET d(4)=11
2220 LET d(5)=12
2230 LET e(1)=5
2240 LET e(2)=7
2250 LET e(3)=7
2260 LET e(4)=9

```

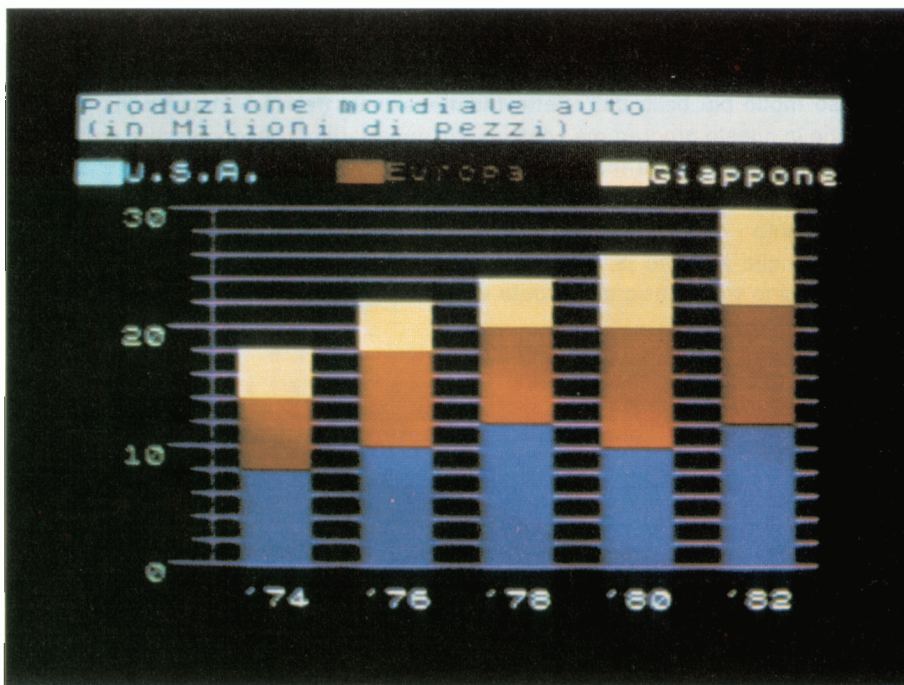
Figura 37 (continua)

```

370 LET f(1)=3
372 LET f(2)=4
374 LET f(3)=5
376 LET f(4)=7
378 LET f(5)=9
380 DIM c$(15,5)
382 REM c$1-5=5X 5 caratteri
400 LET c$(1)="12345"
410 LET c$(2)="78901"
420 LET c$(3)="23456"
430 LET c$(4)="34567"
440 LET c$(5)="45678"
450 REM
5000 REM
5002 IF LEN j$>8 THEN LET j$=j$(
1 TO 8)
504 IF LEN k$>8 THEN LET k$=k$(
1 TO 8)
506 IF LEN m$>8 THEN LET m$=m$(
1 TO 8)
510 IF LEN a$>64 THEN LET a$=a$
(1 TO 64)
520 PRINT AT 0,0: INVERSE 1:a$
522 BRIGHT 1: INK 4: PRINT AT 2
0,0:"0"
530 PRINT AT 5,4-LEN STR$ r/r
532 PRINT AT 10,4-LEN STR$ (r/3
*20):(r/3*2)
535 PRINT AT 15,4-LEN STR$ (r/3
):r/3)
540 BRIGHT 0: INK 7
542 BRIGHT 1: PRINT AT 0,0: INK
0:"j$"; INK 0:AT 0,11:"k$"; INK
0:AT 0,22:"m$"; INK 0:AT 0,33:"a$"; INK 0:AT 0,44:""
544 BRIGHT 0
550 FOR a=1 TO 5
560 IF (d(a)+e(a)+f(a))<0 OR (d
(a)+e(a)+f(a))>7 THEN GO TO 2000
570 NEXT a
580 FOR c=7 TO 27 STEP 5
590 LET k=(c-2)/5
600 PRINT AT 01,c:c$(k)
610 FOR l=19 TO 20-INT ((d(k)+e
(k)+f(k))/r/15)) STEP -1
620 PRINT AT l,c: INK 0;" "
630 NEXT l
640 FOR l=19 TO 20-INT ((d(k)+e
(k))/r/15)) STEP -1
650 PRINT AT l,c: INK 2;" "
660 NEXT l
670 FOR l=19 TO 20-INT (d(k)/r
/15)) STEP -1
680 PRINT AT l,c: INK 1;" "
690 NEXT l: NEXT c
1 2000 STOP
2000 PRINT AT 5,5:" VALORI ERR
11"

```

Figura 37



**Figura 38**

## ISTOGRAMMI VERTICALI TRIDIMENSIONALI

Questo programma (Figura 39) è sostanzialmente simile a quello di Figura 35-36 ma reso molto più bello dall'effetto tridimensionale dei vari istogrammi.

Anche in questo programma i dati inseriti nelle variabili devono essere arrotondati e non superare il valore massimo 'r'.

Questa volta gli istogrammi sono verdi, azzurri e gialli e ad essi corrispondono rispettivamente i dati delle variabili d(1)...(4), e (1)...(4), f(1)...(4) e gli argomenti contenuti nelle stringhe j\$, k\$, w\$.

L'immagine tridimensionale creata con i dati esempio è in figura 40.

```

10 REM      Business Grafica
20 REM      ISTOGRAMMI VERTICALI
30 REM      TRIDIMENSIONALI
40 REM      -----
50 BORDER 0: PAPER 0: CLS
1000 INK 2: PLOT 39,24
110 DRAW 0,119
112 LET U=PEEK 23675+256*PEEK 2
3570
114 DATA 255,127,63,31,15,7,3,1
120 ,192,224,240,248,252,254,255
115 DATA 255,255,255,255,223,207,19
9,195,193,192,90,48,24,12,6,3,1,
192,192,192,192,192,192,192,192
116 FOR a=0 TO U+39
118 READ d: POKE a,d: NEXT a
120 FOR a=5 TO 7
122 PRINT AT 14+a,a: INK 0: PAP
ER 2: CHR$ 145;"
";CHR$ 144
124 NEXT a
126 PLOT 32,23: DRAW 7,0
128 FOR a=31 TO 146 STEP 8
130 PLOT 32,a: DRAW 192,0
140 NEXT a
150 FOR a=23 TO 146 STEP 40
160 PLOT 24,a: DRAW 8,0
170 NEXT a
180 INK 7
200 REM
210 REM      Dati
220 LET a$=max 32 caratteri
222 LET a$="Vendite programmi S
OFTCORP in M$"
224 LET j$="Profes."
226 LET k$="Scient."
228 LET w$="Giochi"
230 REM j$ k$ w$ max 8 caratt.
232 LET r=15
240 REM r=valore massimo scala

```

Figura 39 (continua)

```

0000 REM      Valori per r :
0000 REM      0 15 30 60 150 300 600
0000 DIM d(4)
0000 DIM e(4)
0000 DIM f(4)
0000 REM      Val. d e f(1) -- (4) tra
0000 REM      0 e il valore max "r"
0000 LET d(1)=4
0000 LET d(2)=5
0000 LET d(3)=7
0000 LET d(4)=12
0000 LET e(1)=2
0000 LET e(2)=4
0000 LET e(3)=6
0000 LET e(4)=10
0000 LET f(1)=1
0000 LET f(2)=2
0000 LET f(3)=3
0000 LET f(4)=9
0000 DIM c$(4,3)
0000 REM      c#1-4 max 3 caratteri
0000 LET c$(1)="79"
0000 LET c$(2)="00"
0000 LET c$(3)="01"
0000 LET c$(4)="02"
0000 REM
0000 REM      Programma
0000 IF LEN j$>8 THEN LET j$=j$(
1 TO 8)
0004 IF LEN k$>8 THEN LET k$=k$(
1 TO 8)
0006 IF LEN w$>8 THEN LET w$=w$(
1 TO 8)
0010 IF LEN a$>32 THEN LET a$=a$(
1 TO 32)
0020 PRINT AT 0,0; INVERSE 1; a$
0024 INK 7; PRINT AT 19,2; "0"
0030 PRINT AT 4,3-LEN STR$ r;r
0033 PRINT AT 9,3-LEN STR$ (r/3*
2); (r/3*2)
0035 PRINT AT 14,3-LEN STR$ (r/3
); (r/3)
0042 PRINT AT 2,0; INK 4; "███"; j$
; INK 5; AT 2,11; "███"; k$; INK 6; A
T 2,22; "███"; w$
0050 FOR a=1 TO 4
0060 IF d(a)<0 OR d(a)>r THEN GO
TO 2000
0062 IF e(a)<0 OR e(a)>r THEN GO
TO 2000
0064 IF f(a)<0 OR f(a)>r THEN GO
TO 2000
0070 NEXT a
0080 FOR c=7 TO 22 STEP 5
0090 LET k=(c-2)/5

```

Figura 39 (continua)

```

600 IF d(k)/(r/15) >= 1 THEN PRIN
T AT 19,c) INK 4) PAPER 2,CHR$ 1
47;"██████"/
610 FOR l=18 TO 19-INT (d(k)/(r
/15)) STEP -1
620 PRINT AT l,c) INK 4)CHR$ 14
8;"██████"
630 NEXT l
632 IF d(k)/(r/15) >= 1 THEN PRIN
T AT l+1,c) INK 4)CHR$ 148;"██████";
CHR$ 145
634 NEXT c
636 FOR c=7 TO 22 STEP 5
638 LET k=(c-2)/5
640 IF e(k)/(r/15) >= 1 THEN PRIN
T AT 20,c+1) INK 5, PAPER 2;CHR$
147;"██████"
644 FOR l=19 TO 20-INT (e(k)/(r
/15)) STEP -1
650 PRINT AT l,c+1) INK 5)CHR$
148;"██████"
660 NEXT l

```

Figura 39 (continua)

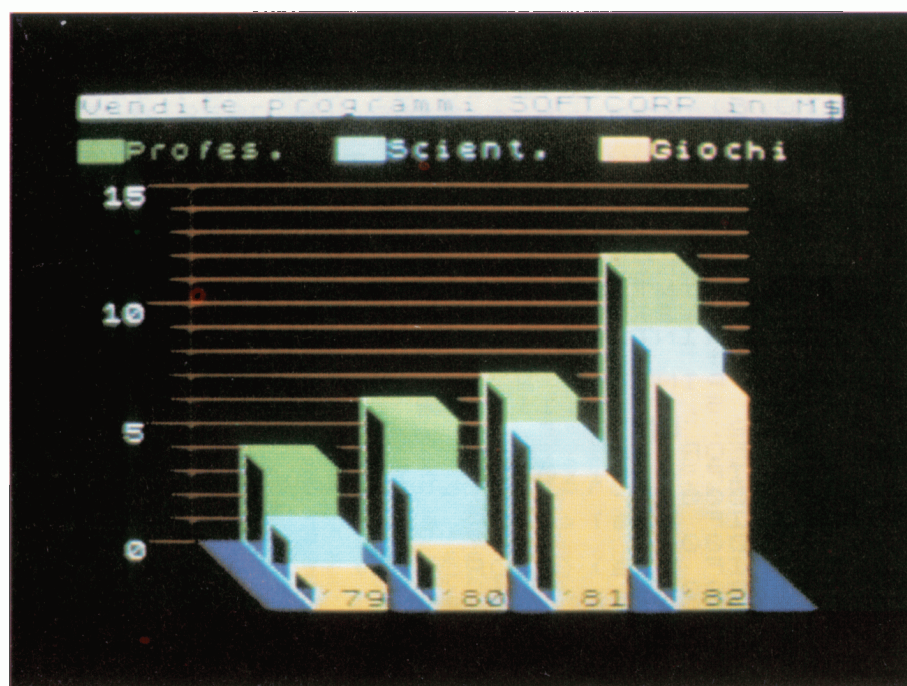


Figura 40

```

662 IF  $e(k)/(r/15) \geq 1$  THEN PRIN
T AT l+1,c+1; INK 5;CHR$ 146;"■"
";CHR$ 145
664 NEXT c
666 FOR c=7 TO 22 STEP 5
668 LET k=(c-2)/5
670 IF  $f(k)/(r/15) \geq 1$  THEN PRIN
T AT 21,c+2; INK 6; PAPER 2;CHR$
147;"■"
677 FOR l=20 TO 21-INT ( $f(k)/(r/15)$ ) STEP -1
680 PRINT AT l,c+2; INK 6;CHR$
148;"■"
690 NEXT l
692 IF  $f(k)/(r/15) \geq 1$  THEN PRIN
T AT l+1,c+2; INK 6;CHR$ 146;"■"
";CHR$ 145
695 PRINT AT 21,c+3; INK 0; PAP
ER 0;c$(k)
699 NEXT c
1000 STOP
2000 PRINT AT 5,5;"■ VALORI ERRO
RI ■"
```

Figura 39

## DIAGRAMMA

Altro classico sistema di rappresentazione di dati in modo grafico è il 'Diagramma' prodotto dal programma di Figura 41.

Nel grafico, la linea segue da sinistra a destra i sei valori contenuti nelle variabili d(1)...(6), valori che possono essere anche decimali ma non superiori ad 'r' che anche in questo caso va prescelto tra i valori indicati nelle linee 260 e 266 ed inserito nella linea 230. Come visibile dall'esempio di Figura 42 l'unico difetto (del resto non determinante) è che la linea gialla cambia il colore del reticolo quando vi passa sopra e questo a causa delle limitazioni di colore in una stessa zona del video.

```

10 REM          Business Grafica
20 REM          DIAGRAMMA
30 REM          -----
40 BORDER 0: PAPER 0: CLS
100 INK 2
102 FOR a=23 TO 143 STEP 8
104 PLOT 40,a: DRAW 7,0
120 NEXT a
130 FOR a=47 TO 248 STEP 8
132 PLOT a,16: DRAW 0,7
140 NEXT a
150 FOR a=23 TO 143 STEP 40
152 PLOT 32,a: DRAW 215,0
160 NEXT a
170 FOR a=47 TO 248 STEP 40
172 PLOT a,8: DRAW 0,135
180 NEXT a
190 INK 7
200 REM          Dati
210 REM          a$ max 64 caratteri
220 LET a$="Variazione % ""Prim
e Rate""
230 LET a$="Banche Italiane decennio
1972-1982"
240 LET r=30
250 REM          r=valore massimo scala
260 REM          valori per r:
262 REM          5 15 30 60 150 300
264 REM          600 1500 3000 6000
270 DIM d(6)
280 REM          valore d(1) -- (6) tra
290 REM          0 e il valore max "r"
300 LET d(1)=5.2
310 LET d(2)=11.4
320 LET d(3)=15.0
330 LET d(4)=16.0
340 LET d(5)=22.5
350 LET d(6)=19.7
360 NEXT a
370 DIM c$(6,3)
380 REM          c$1-6 max 3 caratteri
390 LET c$(1)=" 72"

```

Figura 41 (continua)

```

370 LET c$(2) = " 74"
380 LET c$(3) = " 75"
390 LET c$(4) = " 76"
400 LET c$(5) = " 80"
410 LET c$(6) = " 82"
499 REM
500 REM
510 IF LEN a$ > 64 THEN LET a$ = a$
(1 TO 64)
520 PRINT AT 0,0; INVERSE 1; a$
530 INK 7: PRINT AT 19,3; "0"
540 PRINT AT 4,4-LEN STR$ (r,r
550 PRINT AT 9,4-LEN STR$ (r/3*
2); (r/3*2)
560 PRINT AT 14,4-LEN STR$ (r/3
); (r/3)
570 INK 4
580 FOR a=1 TO 6
590 IF d(a) < 0 OR d(a) > r THEN GO
TO 2000
600 NEXT a
610 FOR a=4 TO 29 STEP 5

```

Figura 41 (continua)

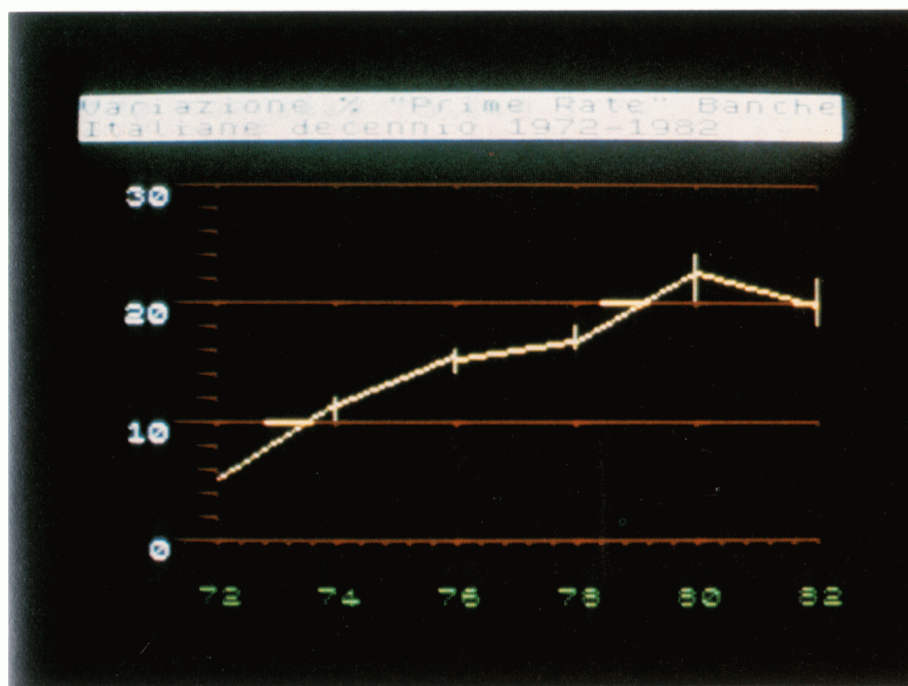


Figura 42

```

0020 PRINT AT 21,a) c$((a+1)/5)
0030 NEXT a
0040 INK 6
0050 FOR x=48 TO 240 STEP 40
0060 LET k=(x-6)/40
0070 LET y=23+d(k)/(r/120)
0080 LET l=23+d(k+1)/(r/120)
0090 PLOT x,y: DRAW 39,l-y
0100 NEXT x
0110 INK 7
10000 STOP
20000 PRINT AT 5,5: " VALORI ERRA
TI "

```

Figura 41

## DIAGRAMMA A SUPERFICIE

Ultimo esempio di Business Grafica è il 'Diagramma a Superficie' utilizzato per confrontare l'andamento di due curve riferite a due argomenti legati tra loro.

Nel programma (Figura 43) i dodici valori che determinano l'andamento della curva blu e della curva gialla vanno inseriti rispettivamente nelle variabili d(1)...(6) e i(1)...(6).

Questo programma può essere utilizzato solo se la distanza tra le due curve è, in tutti i punti, superiore a 16 pixels.

Con i dati esempio si ottiene l'immagine di Figura 44.

```

10 REM Business Grafica
20 REM DIAGRAMMA A SUPERFICIE
30 REM -----
99 BORDER 0: PAPER 0: CLS
100 INK 4
102 FOR a=23 TO 143 STEP 6
104 PLOT 40,a: DRAW 7,0
120 NEXT a
130 FOR a=47 TO 248 STEP 6
132 PLOT a,16: DRAW 0,7
140 NEXT a
150 FOR a=23 TO 143 STEP 40
152 PLOT 32,a: DRAW 15,0: IF a=
23 THEN DRAW 200,0
160 NEXT a
170 FOR a=47 TO 248 STEP 40
172 PLOT a,8: DRAW 0,15: IF a=4
7 THEN DRAW 0,120
180 NEXT a
182 FOR l=4 TO 18
184 PRINT AT l,6: PAPER 6: "
": REM 25
spazi
186 NEXT l
190 INK 7
200 REM Dati
210 REM a$ max 64 caratteri
220 LET a$="Quotazioni medie $
USA (blu) e Inglese (giallo)
230 REM "line"
240 LET r=3000
250 REM r=valore massimo scala
260 REM valor: per r:
270 REM 6 15 30 60 150 300
280 REM 600 1500 3000 6000
290 DIM d(6)
300 REM valore d(1) -- (6) tra
310 DIM i(6)
320 REM valore i(1) -- (6) tra
330 REM 0 e il valore max "r"

```

Figura 43 (continua)

```

0000 LET d(1) = 0000
0010 LET d(2) = 0004
0020 LET d(3) = 7005
0030 LET d(4) = 0040
0040 LET d(5) = 1000
0050 LET d(6) = 1400
0060 LET i(1) = 1000
0070 LET i(2) = 1010
0080 LET i(3) = 1000
0090 LET i(4) = 1070
0100 LET i(5) = 1005
0110 LET i(6) = 2040
0120 NEXT a
0130 DIM c$(6,3)
0140 REM c$1-6 max 3 caratteri
0150 LET c$(1) = "77"
0160 LET c$(2) = "78"
0170 LET c$(3) = "79"
0180 LET c$(4) = "80"
0190 LET c$(5) = "81"
0200 LET c$(6) = "82"
0210 REM
0220 REM
0230 > REM
0240 IF LEN a$ > 64 THEN LET a$ = a$
(1 TO 64)
0250 PRINT AT 0,0: INVERSE 1: a$
0260 INK 7: PRINT AT 19,3: "0"
0270 PRINT AT 4,4-LEN STR$ r: r
0280 PRINT AT 9,4-LEN STR$ (r/3*
2) : (r/3*2)
0290 PRINT AT 14,4-LEN STR$ (r/3
1) : (r/3)
0300 INK 3
0310 FOR a=1 TO 6
0320 IF d(a) < 0 OR d(a) > r THEN GO
TO 2000
0330 IF i(a) < 0 OR i(a) > r THEN GO
TO 2000
0340 NEXT a
0350 FOR a=4 TO 29 STEP 5
0360 PRINT AT 21,a,c$((a+1)/5)
0370 NEXT a
0380 INK 1: PAPER 6
0390 FOR x=48 TO 240 STEP 40
0400 LET k=(x-8)/40
0410 LET y=23+d(k)/(r/120)
0420 LET l=23+d(k+1)/(r/120)
0430 PLOT x,y: DRAW G9,l-y
0440 NEXT x
0450 FOR x=48 TO 248
0460 FOR y=24 TO 143
0470 IF POINT (x,y)=1 THEN GO TO
0480
0490 PLOT x,y
0500 NEXT y
0510 NEXT x
0520 INK 0: PAPER 6

```

Figura 43 (continua)

```

740 FOR x=48 TO 240 STEP 40
750 LET k=(x-8)/40
760 LET y=23+i(k)/(r/120)
770 LET l=23+i(k+1)/(r/120)
780 PLOT x,y: DRAW 39,l-y
790 NEXT x
800 FOR x=48 TO 248
810 FOR y=143 TO 24 STEP -1
820 IF POINT (x,y)=1 THEN GO TO
850
830 PLOT x,y
840 NEXT y
850 NEXT x
999 INK 7: PAPER 0
1000 STOP
2000 PRINT AT 5,5;" VALORI ERRO
TI"

```

Figura 43

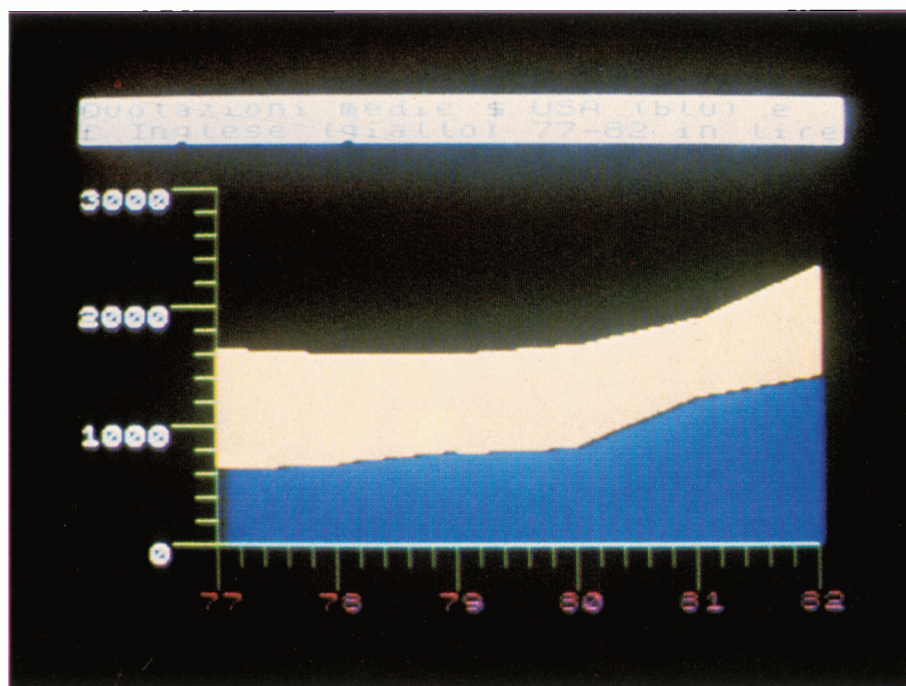


Figura 44



## NEW YORK

'New York' (Figura 45) disegna sullo schermo il panorama di una città con due file di grattacieli. Ogni volta che si fa girare, il programma crea un panorama diverso e casuale; un esempio è in Figura 46.

```
10 REM NEW YORK
20 REM -----
99 BORDER 0: PAPER 0: CLS
100 FOR a=1 TO 20
102 INK 6+INT (RND*2)
104 PLOT RND*255,75+RND*100
106 NEXT a
108 LET k=4
110 DIM l (15)
120 DIM h (15)
130 DIM i (15)
140 FOR a=1 TO 15
150 LET l (a) = (2+INT (RND*2)) * 8
160 LET h (a) = (k+INT (RND*(2+k))
) * 8+INT (RND*k) -1
```

Figura 45 (continua)

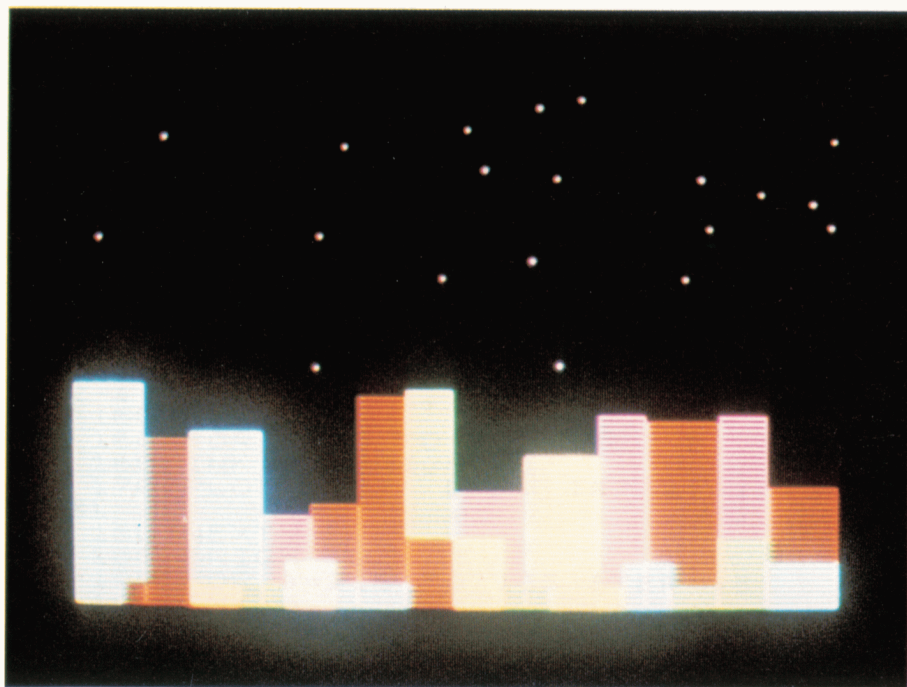


Figura 46

```

170 LET i(a)=2+INT (RND*6)
180 NEXT a
190 LET x=0
200 FOR a=1 TO 15
210 LET x=x+l(a)
220 IF x<255 THEN GO SUB 1000
230 IF x>255 THEN GO TO 250
240 NEXT a
250 LET l(a)=255-(x-l(a))
260 LET x=255
270 GO SUB 1000
280 INK 7
290 IF k=1 THEN GO TO 320
300 LET k=1
310 GO TO 110
320 STOP
1000 INK i(a)
1010 FOR y=0 TO h(a) STEP 2
1020 PLOT x-l(a),y: DRAW l(a)-1,
0
1030 NEXT y
1040 PLOT x-l(a),0: DRAW 0,h(a)-
1
1050 PLOT x-1,0: DRAW 0,h(a)-1
1111 RETURN

```

Figura 45

## BIG BANG

'Big Bang' (o 'Grande Scoppio') è chiamata la teoria, oggi ritenuta più probabile, sulla nascita dell'universo. Il programma di Figura 47 ne dà un'immagine simulata.

```
10 REM BIG BANG
20 REM -----
100 BORDER 0: PAPER 0: INK 2: C
L3
110 FOR a=0 TO 360 STEP 2
120 LET k=5+10*RND
130 LET rd=PI/180*a
140 PLOT 128+k*COS rd,88+k*SIN
rd
150 LET h=30+40.*RND
160 DRAW h*COS rd,h*SIN rd
170 NEXT a
180 INK 7
```

Figura 47

## DISEGNO LINEE

Un disegno molto bello in quattro colori è creato dal programma di Figura 48.

```
10 REM DISEGNO DI LINEE
20 REM -----
88 BORDER 0: PAPER 0: INK 7
99 CLS
100 FOR a=40 TO 120 STEP 8
110 PLOT a,88: DRAW 127-a,a-33
120 NEXT a
130 FOR a=215 TO 135 STEP -8
133 INK 4
140 PLOT a,88: DRAW -(a-128),22
2-a
150 NEXT a
160 FOR a=40 TO 120 STEP 8
166 INK 2: BRIGHT 1
170 PLOT a,87: DRAW 127-a,-(a-3
3)
180 NEXT a
188 BRIGHT 0
190 FOR a=215 TO 135 STEP -8
199 INK 6
200 PLOT a,87: DRAW -(a-128),-(
222-a)
210 NEXT a
222 INK 7
```

Figura 48

### ALVEARE (Figura 49)

```
10 REM      ALVEARE
20 REM      -----
99 BORDER 0: PAPER 0: INK 6: C
LS
100 LET r=10: LET n1=6
110 FOR y=154 TO 0 STEP -30
120 FOR x=9 TO 240 STEP 18
140 GO SUB 4010
150 NEXT x: NEXT y
160 FOR h=154 TO 30 STEP -30
170 FOR a=0 TO 255
180 IF POINT (a,h)=1 THEN PLOT
a,h: DRAW 0,-9
190 NEXT a
200 NEXT h
888 STOP
4010 LET cx=x: LET cy=y+r
4020 PLOT cx,cy
4030 FOR p=90 TO 451 STEP (360/n
1)
4040 LET J=PI/180*p
4050 LET hx=r*COS J+cx
4060 LET hy=r*SIN J+cy
4070 PLOT cx,cy
4080 DRAW hx-cx,hy-cy
4090 LET cx=hx: LET cy=hy
4100 NEXT p: RETURN
```

Figura 49

### PIRAMIDE (Figura 50)

```
10 REM      PIRAMIDE
20 REM      -----
99 BORDER 0: PAPER 0: INK 6: C
LS
100 LET x=122
110 LET k=1
120 FOR y=154 TO 0 STEP -5
130 FOR n=1 TO k
140 PLOT x+(n-1)*8,y: DRAW 7,0:
DRAW 0,-4: DRAW -7,0: DRAW 0,4
150 NEXT n
160 LET x=x-4
170 LET k=k+1
180 NEXT y
```

Figura 50

## CIELO STELLATO

'Cielo Stellato' (Figura 51) disegna anche i pianeti Marte, Giove e Saturno.

```

10 REM CIELO STELLATO
20 REM -----
100 BORDER 0: PAPER 0: CLS
110 OPTPAPER 0: 10: 0: 10: 0: 10: 0
120 OPTPAPER 0: 10: 40: 10: 0: 10: 0
130 OPTPAPER 0: 10: 80: 10: 40: 10: 0
140 OPTPAPER 0: 10: 120: 10: 80: 10: 0
150 OPTPAPER 0: 10: 160: 10: 120: 10: 0
160 LET C=PEEK 23679+256*PEEK 2
170
180 FOR a=0 TO C+47
190 READ d: POKE a,d: NEXT a
200 FOR l=0 TO 21
210 FOR c=0 TO 31
220 LET x=INT (RND*30)
230 LET i=2+INT (RND*6)
240 PRINT AT l,c:CHR$ 128
250 IF x=6 OR x=7 THEN PRINT AT
  l,c: INK 1:CHR$ 144
260 IF x=8 THEN PRINT AT l,c: I
  NK 1:CHR$ 145
270 IF x=9 THEN PRINT AT l,c: I
  NK 1:CHR$ 146
280 IF l=8 AND c=25 THEN PRINT
  AT l,c: INK 8:CHR$ 147
290 IF l=9 AND c=18 THEN PRINT
  AT l,c: INK 5:CHR$ 148
300 IF l=16 AND c=12 THEN PRINT
  AT l,c: INK 0:CHR$ 149
310 NEXT c: NEXT l

```

Figura 51

## MONTAGNE

Ogni volta che si fa girare, il programma 'Montagne' (Figura 52) crea un panorama diverso e casuale.

```
10 REM MONTAGNE
20 REM -----
100 BORDER 0: PAPER 1: INK 4: C
LS
110 DIM h(20)
120 DIM l(20)
130 FOR a=1 TO 20
140 LET l(a)=12+INT (RND*30)
150 LET h(a)=10+INT (RND*50)
160 NEXT a
170 LET x=0: LET y=h(1)
180 PLOT x,y
190 FOR a=2 TO 20
200 LET x=x+l(a)
210 IF x<255 THEN DRAW l(a),h(a)
) -y
220 IF x>255 THEN GO TO 250
230 LET y=h(a)
240 NEXT a
250 DRAW 255-(x-l(a)),h(a)-y
260 FOR a=0 TO 255
270 FOR n=0 TO 175
280 IF POINT (a,n)=1 THEN GO TO
310
320 PLOT a,n
330 NEXT n
340 NEXT a
```

Figura 52

## BANDIERA REGNO UNITO

Quello di Figura 53 è il primo di tre programmi che disegnano delle bandiere, a colori e molto vicine alle vere. Della prima viene anche dato il risultato finale (Figura 54).

```

10 REM bandiera REGNO UNITO
20 REM -----
100 BORDER 0: PAPER 0: INK 7: C
L3
120 FOR a=1 TO 640
130 PRINT ; PAPER 7;" ";
140 NEXT a: PRINT
150 PAPER 7: INK 2
160 FOR l=0 TO 19
170 PRINT AT l,14: PAPER 7: INK
0:" "
180 NEXT l
190 FOR c=0 TO 31
200 PRINT AT 9,c:" "
210 PRINT AT 10,c:" "
220 PRINT AT 11,c:" "
230 NEXT c
240 PRINT AT 11,14:" "
250 FOR x=0 TO 103
260 PLOT x,176-x*0.6: DRAW 0,-9
270 NEXT x
280 PRINT AT 8,11:" "
290 FOR y=76 TO 85 STEP -1
300 PLOT 148+(76-y)*1.80,y: DRA
L14,0
310 NEXT y
320 FOR x=242 TO 255
330 PLOT x,24: DRAW 255-x,-8
340 NEXT x
350 FOR x=15 TO 103
360 PLOT x,16+(x-15)*0.68: DRAW
-14,0
370 NEXT x
380 FOR x=148 TO 241
390 PLOT x,112+(x-148)*.68: DRA
L14,0
400 NEXT x
410 INK 1
420 FOR y=112 TO 155
430 PLOT 0,y: DRAW 72-(y-112)+1
.69,0
440 NEXT y
450 FOR y=112 TO 164
460 PLOT 255,y: DRAW -(164-y)*1
.5,0
470 NEXT y
480 FOR y=76 TO 86 STEP -1

```

Figura 53 (continua)

```

470 PLOT 0,y: DRAW 72-(76-y)*1.
480 NEXT y
490 FOR y=76 TO 34 STEP -1
500 PLOT 255,y: DRAW -(y-34)*1.
510 NEXT y
520 FOR x=16 TO 103
530 PLOT x,175: DRAW 0,-(x-16)*
540 NEXT x
550 FOR x=238 TO 148 STEP -1
560 PLOT x,16: DRAW 0,(238-x)*0
570 NEXT x
580 FOR x=29 TO 103
590 PLOT x,16: DRAW 0,(x-29)*0.
600 NEXT x
610 FOR x=225 TO 148 STEP -1
620 PLOT x,175: DRAW 0,-(225-x)
630 NEXT x

```

Figura 53

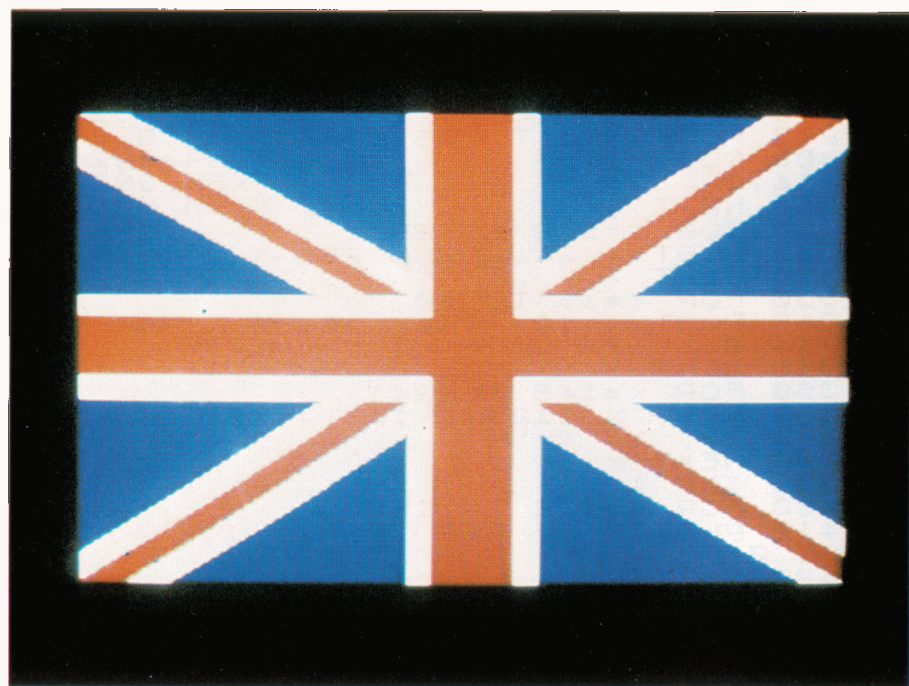


Figura 54

# BANDIERA STATI UNITI (Figura 55)

```

10 REM Bandiera USH
20 REM -----
100 BORDER 0: PAPER 0: INK 7: C
L3
110 LET U=PEEK 23675+256*PEEK 2
3675
120 DATA 16,16,166,124,56,40,68
,0
130 FOR a=U TO U+7
140 READ d: POKE a,d: NEXT a
150 LET a$="
160 LET b$="
170 LET c$="
180 REM 32 spazi
180 FOR l=0 TO 17 STEP 3
190 PRINT AT l,0: PAPER 7: INK
2;a$
200 PRINT AT l+1,0: PAPER 7: IN
K 2;b$
210 PRINT AT l+2,0: PAPER 7:c$
220 NEXT l
230 PRINT AT 18,0: PAPER 7: INK
20;a$
240 PRINT AT 19,0: INK 2;b$
250 LET p$=" "+CHR$ 144+" "+CHR$ 144+" "+CHR$
$ 144+" "+CHR$ 144+" "+CHR$ 144+" "+CHR$ 144
+ " "+CHR$ 144+" "+CHR$ 144+" "+CHR$ 144+" "
260 LET j$=" "+p$(1 TO 11)+" "
270 PRINT AT 0,0: PAPER 1:j$
280 REM 13 spazi
280 FOR l=1 TO 10 STEP 2
7;p$
290 PRINT AT l,0: PAPER 1: INK
300 IF l<9 THEN PRINT AT l+1,0:
PAPER 1: INK 7:j$
310 NEXT l
320 PRINT AT 10,0: PAPER 1: INK
7;"

```

Figura 55

## BANDIERA GIAPPONESE (Figura 56)

```
10  REM  Bandiera GIAPPONESE
20  REM  -----
100  BORDER 0: PAPER 0: INK 2: C
110  FOR a=1 TO 540
120  PRINT PAPER 7;" ";
130  NEXT a: PRINT
140  PRINT AT 5,14;" "
150  PRINT AT 6,10;" "
160  PRINT AT 7,12;" "
170  PRINT AT 8,11;" "
180  PRINT AT 9,11;" "
190  PRINT AT 10,11;" "
200  PRINT AT 11,11;" "
210  PRINT AT 12,12;" "
220  PRINT AT 13,12;" "
230  PRINT AT 14,14;" "
240  FOR r=48 TO 33 STEP -0.2
250  CIRCLE 128,96,r
260  IF r<48 THEN CIRCLE 127,95,
270  NEXT r
```

Figura 56

# PROGRAMMI VARI

## FISICA DEI REATTORI NUCLEARI

Il programma (Figura 57) simula il processo di 'Fissione Nucleare' che è alla base del funzionamento appunto delle centrali nucleari che producono energia. La Fissione Nucleare è una reazione provocata da neutroni veloci che colpiscono degli atomi di Uranio 235 i quali si spaccano producendo due atomi più piccoli di Bario e Kripto ed altri due neutroni che colpendo a loro volta degli atomi di Uranio provocano la cosiddetta 'reazione a catena'. La reazione produce anche energia che viene sfruttata per produrre elettricità e calore. La Figura 58 mostra una fase intermedia della reazione di fissione. In questo programma è molto importante inserire bene gli spazi nelle varie stringhe.

```
10 REM FISICA DEI REATTORI
20 REM NUCLEARI
30 REM -----
100 BORDER 0: PAPER 0: INK 7: C
LS
110 DATA 0,0,0,24,24,0,0,0,0,24
,60,126,126,60,24,0,60,126,255,2
55,255,255,126,60
120 LET U=PEEK 23675+256*PEEK 2
3676
130 FOR a=U TO U+23
140 READ c: POKE a,c: NEXT a
150 PRINT "SIMULATORE DI FISSI
ONE NUCLEARE"
160 PRINT AT 2,20: INK 6: CHR$ 1
44: INK 7: "_Neutrone"
170 PRINT AT 4,20: INK 4: CHR$ 1
46: INK 7: "_Uranio 235"
180 PRINT AT 6,20: INK 5: CHR$ 1
48: INK 7: "_Bario 139"
190 PRINT AT 8,20: INK 2: CHR$ 1
45: INK 7: "_Kripto 97"
200 INK 6: PRINT AT 2,15: CHR$ 1
44: INK 4: PRINT AT 6,15: CHR$ 14
6: AT 14,7: CHR$ 146: AT 14,23: CHR$
146: AT 18,3: CHR$ 146: AT 18,11: C
HR$ 146: AT 18,19: CHR$ 146: AT 18,
27: CHR$ 146
```

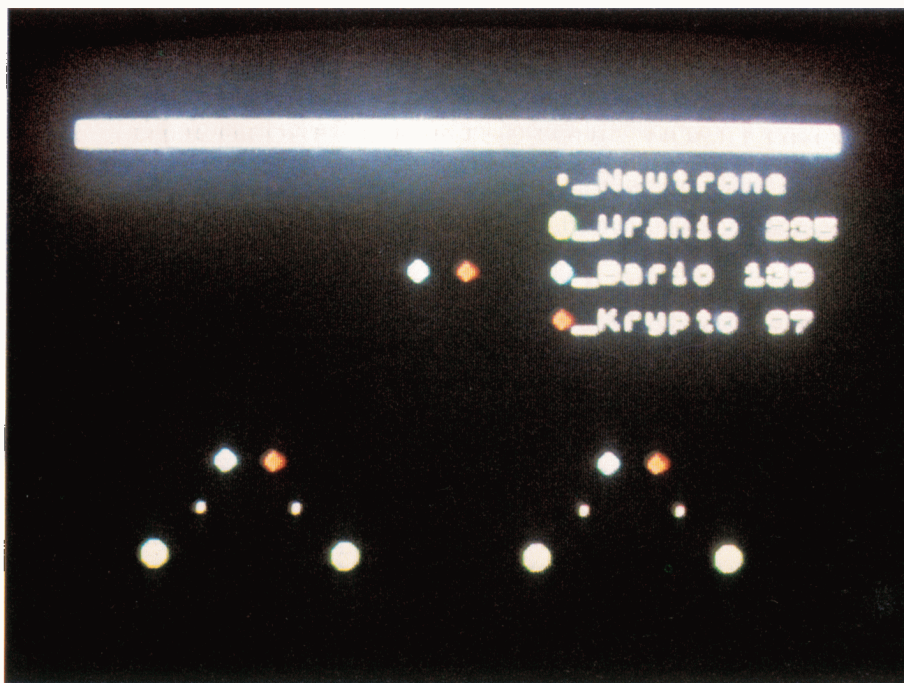
Figura 57 (continua)

```

210 LET a$=CHR$ 16+CHR$ 5+CHR$
145+CHR$ 128+CHR$ 16+CHR$ 2+CHR$
145
220 LET n$=CHR$ 16+CHR$ 6++CHR$
144
230 LET b$="
REM 32 spazi
240 PRINT AT 3,0; INK 3;"Premier
e ENTER"
250 INPUT LINE a$
260 PRINT AT 3,0;"
270 FOR L=3 TO 6
277 PAUSE 1
280 PRINT AT L-1,15;" ";AT L,15
) n$
3000 NEXT L
300 PRINT AT 6,14,a$
310 BEEP 0.1,-10
320 FOR L=1 TO 7
330 PRINT AT L+6,15-L;n$;AT L+6
15+L;n$;PAUSE 1;PRINT AT L+6,
15-L;" ";AT L+6,15+L;" "
340 NEXT L
350 PRINT AT 14,6,a$;AT 14,22;a
$
360 BEEP 0.1,-10
370 PRINT AT 15,6;n$;" ";n$;"
";n$;";n$
380 PAUSE 1;PRINT AT 15,0;b$
390 PRINT AT 16,6;n$;" ";n$;"
";n$;";n$
400 PAUSE 1;PRINT AT 16,0;b$
410 PRINT AT 17,4;n$;" ";n$
";n$;";n$
420 PAUSE 1;PRINT AT 17,0;b$
430 PRINT AT 18,3;n$;" ";n$;"
";n$;";n$
440 PAUSE 1
450 PRINT AT 18,2;a$;" ";a$
";a$;";a$
460 BEEP 0.1,-10
470 PRINT AT 19,2;n$;" ";n$;"
";n$;";n$;";n$;";n$;";n$
";n$;";n$;";n$;";n$;";n$
480 PAUSE 1;PRINT AT 19,0;b$
490 PRINT AT 20,1;n$;" ";n$;"
";n$;";n$;";n$;";n$;";n$
";n$;";n$;";n$;";n$;";n$
500 PAUSE 1;PRINT AT 20,0;b$
510 PRINT AT 21,0;n$;" ";n$;"
";n$;";n$;";n$;";n$;";n$
";n$;";n$;";n$;";n$;";n$
520 PAUSE 1;PRINT AT 21,0;b$

```

Figura 57



**Figura 58**

## LETTURA VELOCE

È possibile esercitarsi a riconoscere più velocemente parole e frasi, in modo da riuscire a leggere più rapidamente quando occorre, usando il programma di Figura 59. Per usare il programma si devono innanzitutto inserire le parole o le frasi (dalla cui lunghezza dipende anche il grado di difficoltà) in una serie di istruzioni DATA a partire dalla linea 1001. Ogni linea deve contenere una sola frase di massimo 32 caratteri e/o spazi, segni, numeri, ecc.

Le frasi non devono contenere delle keyword ma solo i caratteri con codice da 32 a 126 (vedere appendice A del manuale Spectrum); inoltre le linee devono essere numerate di seguito (1001, 1002, 1003, ecc.) senza saltarne nessuna e devono concludersi con la linea 9999 DATA "#" che serve al programma per contare le frasi. Le parole o le frasi possono essere scritte in qualsiasi lingua ed il loro numero può essere anche di alcune centinaia ed è limitato solo dalla capacità di memoria dello Spectrum (lo spectrum 16K può contenere oltre 500 parole e/o frasi di 10/15 caratteri ciascuna).

Naturalmente è bene che la persona che sceglie ed introduce le frasi sia diversa da quella che deve usare il programma, e questo per evitare che il test di lettura veloce si trasformi in realtà in un test di memoria (nel senso di ricordare le frasi che si sono introdotte).

Inserite le frasi si può far girare il programma. La prima frase viene sempre visualizzata per un tempo di 0.1 secondi che viene indicato in anticipo (Figura 60), quindi esce la frase scelta casualmente (Figura 61). Sparita la frase il computer chiede di scriverla (Figura 62), a questo punto si deve inserire la frase esattamente (rispettando anche gli spazi e i caratteri maiuscoli o minuscoli).

Se la frase inserita è uguale a quella scelta dal Computer esce la scritta di Figura 63, se invece è sbagliata esce la scritta di Figura 64.

Inoltre ogni volta che si riconosce esattamente una frase il tempo di visualizzazione della successiva viene dimezzato (fino ad un minimo di 0.02 sec.), mentre ogni volta che si sbaglia viene raddoppiato (fino ad un massimo di 1 sec.).

```
10 REM LETTURA VELOCE
20 REM -----
100 BORDER 0: PAPER 0: INK 6: C
LS
110 LET b$=""
120 LET k$=b$+b$+b$
130 PRINT ; PAPER 1; k$
140 FOR a=1 TO 10000
150 RESTORE 1000+a
160 READ c$: IF c$="#" THEN GO
TO 180
170 NEXT a
180 LET nf=a-1
```

Figura 59 (continua)

```

1900 DIM t$(6,12)
2000 LET t$(1)="01 0.00 0 sec."
2100 LET t$(2)="02 0.04 0 sec."
2200 LET t$(3)="05 0.1 0 sec."
2300 LET t$(4)="10 0.2 0 sec."
2400 LET t$(5)="20 0.0 0 sec."
2440 LET t$(6)="50 1 0 sec."
2500 LET tx=3
2600 PAUSE 100
2700 PRINT AT 9,0; INK 4; "Durata
Frases ";t$(tx,4 TO 5)
2800 PAUSE 30+RND*30
2900 LET p=VAL t$(tx, TO 2)
3000 RESTORE 1001+INT (RND*6)
3100 READ f$: IF LEN f$>32 THEN
LET f$=f$( TO 32)
3200 PRINT AT 1,0; PAPER 1; f$
3300 PAUSE p
3400 PRINT AT 1,0; PAPER 1; b$
3500 PRINT AT 9,0; INK 4; "Scrivi
te la frase che avete visto
e premete ENTER"
3600 INPUT LINE x$
3700 LET r=0: IF x$=f$ THEN LET
r=1
3800 IF r=1 THEN PRINT AT 9,0; I
NK 7; "Frases: ESATTA"; b$; b$
3900 IF r=0 THEN PRINT AT 9,0; I
NK 3; "Frases: SBAGLIATA !!"; b$; b$
4000 IF r=1 AND tx>1 THEN LET tx
=tx-1
4100 IF r=0 AND tx<6 THEN LET tx
=tx+1
4200 GO TO 260
9999 REM -----
10000 REM Catalogo Frasi
10001 DATA "Spectrum"
10002 DATA "123 PROVA"
10003 DATA "SINCLAIR"
10004 DATA "Frasi di max trentadu
e caratteri"
10005 DATA "in qualsiasi lingua"
10006 DATA "Los Angeles"
10007 DATA "GREEN"
99999 DATA "#": REM fine lista

```

Figura 59

Durata Frase 0.1 Sec.

Figura 60

GREEN

Durata Frase 0.2 Sec.

**Figura 61**

Scrivete la frase che avete  
visto e premete **ENTER**

**Figura 62**

Frase: **ESATTA**

**Figura 63**

Frase: **SBAGLIATA !!**

**Figura 64**

## CRONOMETRO

Il Cronometro (Figura 65) permette di misurare i tempi e di visualizzarli in ore, minuti, secondi e centesimi di secondo (Figura 66).

La precisione è buona poichè il clock dello Spectrum è al quarzo con una precisione dichiarata dello 0.01%.

Il Cronometro è controllato dai tasti 'Z' (Start), 'C' (Stop) e 'X' (Lap), il tasto 'X' in particolare fornisce i risultati intermedi e può essere premuto quante volte si vuole durante il cronometraggio.

Importante: in questo programma non va inserita l'istruzione 'BORDER' altrimenti non funziona, eventualmente il colore del bordo può essere scelto prima di scrivere il programma con un comando diretto.

```

10 REM CRONOMETRO
20 REM -----
30 REM      funzione      tasto
33 REM      -----
40 REM      START      Z
50 REM      LAP      X
60 REM      STOP      C
70 REM -----
99 PAPER 0: INK 7: CLS
100 DATA 33,120,92,0,62,254,219
,054,054,253,32,247
110 DATA 54,0,35,54,0,35,54,0,3
3,120,92,1,0,91,0,62,254,219,254
,254,251
120 DATA 40,7,254,247,40,9,0,24
,240,0,62,1,2,24,4,0,62,0
130 DATA 2,0,3,126,2,3,35,126,2
,3,35,126,2,201
140 LET U=PEEK 23296+256*PEEK 2
3676
150 FOR a=U TO U+64
160 READ d: POKE a,d: NEXT a
166 CLS
170 PRINT AT 0,11: INK 4: "hr mi
n sec"
180 PRINT AT 6,0: INK 5: "testi:
[Z]=START [X]=LAP [C]=STOP"
190 LET k=USR U
200 GO SUB 300
210 IF PEEK 23296=1 THEN LET k=
USR (U+20)
215 IF PEEK 23296=1 THEN GO TO
200
220 GO SUB 300
230 PRINT AT 8,2: "Premere ENTER
per continuare"

```

Figura 65 (continua)

```

240 INPUT e$
250 PRINT AT 0,2;"          REM 20 spazi
299 GO TO 166
300 PRINT AT 2,10;" 00: 00: 00.
000:
310 LET t=65536*PEEK 23299+256*
PEEK 23298+PEEK 23297
320 LET h=INT (t/180000)
330 LET t=t-h*180000
340 LET m=INT (t/3000)
350 LET t=t-m*3000
360 LET s=INT (t/50)
370 LET c=(t-s*50)*2
380 LET h$=STR$ h
390 LET m$=STR$ m
400 LET s$=STR$ s
410 LET c$=STR$ c
420 PRINT AT 0,10-LEN h$;h$
430 PRINT AT 0,17-LEN m$;m$
440 PRINT AT 0,21-LEN s$;s$
450 PRINT AT 0,29;c$
499 RETURN

```

Figura 65

```

hr min sec
00: 01: 25.44

```

tasti: ☒=START ☒=LAP ☒=STOP  
 Premere **ENTER** per continuare

Figura 66

## SCRITTE ROTANTI

Una delle applicazioni del programma 'Scritte Rotanti' (Figura 67) può essere nella pubblicità e nell'informazione in genere.

Dopo avere premuto RUN e ENTER si inserisce prima la frase (anche di molti caratteri) e poi la velocità di rotazione, dopodichè i caratteri, i numeri, ecc. della frase inserita inizieranno a ruotare da destra a sinistra.

Le Figure 68 e 69, 70 e 71, mostrano due esempi di scritte rotanti ciascuno dei quali visto in due differenti momenti; comunque il modo migliore per rendersi conto del funzionamento è quello di provare il programma in pratica.

Se si vuole rendere stabile la frase in modo da registrarla su nastro insieme al programma si può sostituire la linea 110 INPUT a\$ con la linea 110 LET a\$="frase" inoltre se nella frase si vogliono inserire anche dei caratteri speciali (Figura 72) si può inserire la relativa routine per caricarli in memoria tra la linea 1115 e la linea 2221.

Infine inserendo nel programma le linee di Figura 73 si può fermare la rotazione della frase ogni volta che si preme il tasto 'p' e fino a che non si preme ENTER.

```
10 REM SCRITTE ROTANTI
20 REM -----
99 GO SUB 1111
100 BORDER 0: PAPER 0: CLS
110 INPUT "Inserire frase e preme ENTER", LINE a$
120 FOR a=1 TO LEN a$
130 IF CODE a$(a)<32 OR CODE a$(a)>164 THEN LET a$(a)=" "
140 NEXT a
150 INPUT "Inserire velocità (1-10) e premere ENTER", t:
IF t<1 OR t>10 THEN GO TO 150
160 FOR a=3 TO 7
170 PRINT AT a,0: PAPER 1;"
REM 32 spazi
180 NEXT a
190 IF LEN a$<32 THEN DIM h$(32-LEN a$+10)
195 IF LEN a$>31 THEN DIM h$(10)
200 LET b$=a$+h$
210 LET c$=b$
220 PRINT AT 5,0: INK 6;c$(1 TO 32)
230 LET c$=c$(2 TO )
240 IF LEN c$<32 THEN LET c$=c$+b$
250 PAUSE t*5
```

Figura 67 (continua)

```

260 GO TO 220
1111 REM -----
1112 REM routine per eventuali
1113 REM caratteri speciali
1114 REM -----
2222 RETURN

```

Figura 67

QUESTA E' UNA SCRITTA ROTANTE...

Figura 68

TE.... QUESTA E' UNA SC

Figura 69

US. 8\_1495 SCAN.\_1230 2\_2

Figura 70



Figura 71



Figura 72

```
255 IF INKEY$="P" THEN GO TO 27
260 GO TO 220
270 INPUT LINE &$
280 GO TO 220
```

Figura 73

## CONTAPEZZI

Un uso insolito dello Spectrum può essere quello di Contapezzi (Figura 74) capace di contare da 0 a 99.999.999 pezzi.

Facendo girare il programma, ogni volta che si preme ENTER il numero viene incrementato di uno (Figura 75). Oltre a ciò, si può inserire anche un numero qualsiasi da aggiungere (per esempio 334925) che premendo ENTER viene sommato al numero presente sullo schermo oppure sottratto, se il numero inserito è preceduto dal segno '-'.  
Se si cerca di sommare o di sottrarre un numero che farebbe scendere il totale sotto lo 0 o lo farebbe salire oltre il massimo, il programma fa uscire sul video la frase 'NUMERO ERRATO O ECCESSIVO'.

```

10 REM      CONTAPEZZI CH 0 H
20 REM      99.999.999 PEZZI
30 REM      -----
100 BORDER 0: PAPER 0: INK 6: C
LS
110 PRINT INK 4;"Premere ENTER
per 1 solo pezzo"
120 PRINT
130 PRINT INK 4;"o introdurre N
pezzi in piu' o in"
140 PRINT
150 PRINT INK 4;"meno (con -) e
premere ENTER"
160 PRINT AT 10,0; INK 5;"L I I I
I I I I I I I I I I I I I I I I
170 PRINT AT 11,0; INK 5;"I I I I
I I I I I I I I I I I I I I I I
180 PRINT AT 12,0; INK 5;"I I I I
I I I I I I I I I I I I I I I I
190 LET n=0
200 LET c=0
210 INPUT LINE n$
220 IF n$="" THEN LET n$="1"
230 IF CODE n$ <> 45 THEN FOR a=1
TO LEN n$
240 IF CODE n$=45 THEN FOR a=2
TO LEN n$
250 IF CODE n$(a) < 48 OR CODE n$
(a) > 57 THEN GO TO 600
260 NEXT a
270 IF (n+VAL n$) > (1E8-1) OR (n
+VAL n$) < 0 THEN GO TO 600
280 LET n=n+VAL n$
290 DIM b$(1,9-LEN STR$ n)
300 LET a$=STR$ n+b$(1)
310 FOR p=1 TO LEN a$-1
320 LET c=p*2+14

```

Figura 74 (continua)

```

0330 LET J=CODE a$(P)
0340 IF J=00 THEN GO SUB 1997
0350 IF J>00 THEN GO SUB (J-48)*
1000+1000
0360 NEXT P
0370 GO TO 210
0000 PRINT AT 15,5; INK 7;"NUMER
O ERRATO O ECCES6100"
0610 PAUSE 100
0000 PRINT AT 15,5;"
      REM 25 spazi
0630 GO TO 210
10000 PRINT AT 10,c;" 0 "
10010 PRINT AT 11,c;"  "
10020 PRINT AT 12,c;"  "
10030 RETURN
11000 PRINT AT 10,c;" 1 "
11010 PRINT AT 11,c;"  "
11020 PRINT AT 12,c;"  "
11030 RETURN
12000 PRINT AT 10,c;" 2 "
12010 PRINT AT 11,c;"  "
12020 PRINT AT 12,c;"  "
12030 RETURN
13000 PRINT AT 10,c;" 3 "
13010 PRINT AT 11,c;"  "
13020 PRINT AT 12,c;"  "
13030 RETURN
14000 PRINT AT 10,c;" 4 "
14010 PRINT AT 11,c;"  "
14020 PRINT AT 12,c;"  "
14030 RETURN
15000 PRINT AT 10,c;" 5 "
15010 PRINT AT 11,c;"  "
15020 PRINT AT 12,c;"  "
15030 RETURN
16000 PRINT AT 10,c;" 6 "
16010 PRINT AT 11,c;"  "
16020 PRINT AT 12,c;"  "
16030 RETURN
17000 PRINT AT 10,c;" 7 "
17010 PRINT AT 11,c;"  "
17020 PRINT AT 12,c;"  "
17030 RETURN
18000 PRINT AT 10,c;" 8 "
18010 PRINT AT 11,c;"  "
18020 PRINT AT 12,c;"  "
18030 RETURN
19000 PRINT AT 10,c;" 9 "
19010 PRINT AT 11,c;"  "
19020 PRINT AT 12,c;"  "
19030 RETURN
19997 PRINT AT 10,c;"  "
19998 PRINT AT 11,c;"  "
19999 PRINT AT 12,c;"  "
20000 RETURN

```

Figura 74

Premere **ENTER** per 1 solo pezzo  
o introdurre N° pezzi in più o in  
meno (con -) e premere **ENTER**

**N°PEZZI 53774923**

**Figura 75**

## **SIMULATORE DI PORTE LOGICHE**

Il programma (Figura 76) simula il funzionamento delle sei porte logiche che sono alla base di qualsiasi circuito digitale incluso i Computer. Inoltre lo stesso microprocessore Z80 usato dallo Spectrum ha nel suo set di istruzioni, tre istruzioni logiche (AND, OR, XOR) il cui funzionamento è simile a quello delle corrispondenti porte digitali.

La scelta delle porte e la selezione di '1' e '0' ai due ingressi 'A' e 'B' avviene premendo i seguenti tasti:

<b>Tasto</b>	<b>Funzione</b>
<b>1</b>	<b>ingresso 'A' a '1'</b>
<b>2</b>	<b>ingresso 'A' a '0'</b>
<b>3</b>	<b>ingresso 'B' a '1'</b>
<b>4</b>	<b>ingresso 'B' a '0'</b>
<b>5</b>	<b>porta AND</b>
<b>6</b>	<b>porta NAND</b>
<b>7</b>	<b>porta OR</b>
<b>8</b>	<b>porta NOR (Figura 77)</b>
<b>9</b>	<b>porta EX-OR</b>
<b>0</b>	<b>porta EX-NOR</b>

```

10 REM      SIMULATORE DI
20 REM      PORTE LOGICHE
30 REM      -----
77 BORDER 1: PAPER 0
88 INK 7: CLS
98 PRINT AT 1,1; INK 2;"■";" " =
100 PRINT AT 3,1; INK 5;"■";" " =
110 PRINT AT 4,9; INK 6;"A";AT
120 INK 6;"B";AT 5,23; INK 4;"O
130 LET P=0: LET a=0: LET b=0
140 LET K=CODE INKEY$-47
150 IF K>0 AND K<11 THEN GO TO
160 K*5000
170 GO TO 120
180 REM      -----
190 REM      EX-NOR
200 LET P=5
210 FOR I=1 TO 7: PRINT AT I,11
220 NEXT I
230 PRINT AT 1,13;"EX-NOR"
240 PLOT 112,112
250 DRAW 38,20,PI/3
260 DRAW -38,20,PI/3
270 DRAW 0,-40,-PI/3
280 PLOT 104,112: DRAW 0,40,PI/3
290 PLOT 96,143: DRAW 19,0
300 PLOT 96,119: DRAW 19,0
310 CIRCLE 154,132,3: PLOT 158,
320 DRAW 10,0
330 REM      -----
340 IF K=0 THEN LET a=1
350 IF K=2 THEN PRINT AT 4,11;
360 INK 2;"■";" "
370 IF K=3 THEN LET a=0
380 IF K=3 THEN PRINT AT 4,11;
390 INK 5;"■";" "
400 IF K=4 THEN LET b=1
410 IF K=4 THEN PRINT AT 6,11;
420 INK 6;"■";" "
430 IF K=5 THEN LET b=0
440 IF K=5 THEN PRINT AT 6,11;
450 INK 4;"■";" "
460 GO TO 3600+500*P
470 PRINT AT 5,21; INK 5;"■";" "
480 IF a=1 AND b=1 THEN PRINT A
490 T 5,21; INK 2;"■";" "
500 GO TO 120
510 PRINT AT 5,21; INK 2;"■";" "
520 IF a=1 AND b=1 THEN PRINT A
530 T 5,21; INK 5;"■";" "
540 GO TO 120
550 PRINT AT 5,21; INK 2;"■";" "

```

Figura 76 (continua)

```

2760 IF a=0 AND b=0 THEN PRINT A
T 5,21; INK 5; "■"
2770 GO TO 120
2800 PRINT AT 5,21; INK 5; "■"
2810 IF a=0 AND b=0 THEN PRINT A
T 5,21; INK 2; "■"
2820 GO TO 120
2850 PRINT AT 5,21; INK 2; "■"
2860 IF a=b THEN PRINT AT 5,21;
INK 5; "■"
2870 GO TO 120
2900 PRINT AT 5,21; INK 5; "■"
2910 IF a=b THEN PRINT AT 5,21;
INK 2; "■"
2920 GO TO 120
3000 REM AND
3010 FOR l=1 TO 7: PRINT AT l,11
...: NEXT l
3015 PRINT AT 1,13; "AND"
3020 LET p=1
3022 PLOT 104,143: DRAW -9,0
3024 PLOT 104,119: DRAW -9,0
3026 PLOT 168,131: DRAW -16,0
3030 PLOT 104,112: DRAW 0,38: DR
AW 28,0: DRAW 0,-38,-PI: DRAW -2
8,0
3030 GO TO 1000
3500 REM NAND
3510 LET p=2
3515 FOR l=1 TO 7: PRINT AT l,11
...: NEXT l
3520 PRINT AT 1,13; "NAND"
3530 PLOT 104,143: DRAW -9,0
3540 PLOT 104,119: DRAW -9,0
3550 PLOT 168,131: DRAW -16,0
3560 PLOT 104,112: DRAW 0,38: DR
AW 28,0: DRAW 0,-38,-PI: DRAW -2
8,0: CIRCLE 155,131,3
3560 GO TO 1000
4000 REM OR
4010 LET p=3
4020 FOR l=1 TO 7: PRINT AT l,11
...: NEXT l
4030 PRINT AT 1,13; "OR"
4040 PLOT 112,112
4050 DRAW 38,20,PI/3
4060 DRAW -38,20,PI/3
4070 DRAW 0,-40,-PI/3
4080 PLOT 96,143: DRAW 19,0
4082 PLOT 96,119: DRAW 19,0
4084 PLOT 151,132: DRAW 16,0
4090 GO TO 1000
4500 REM NOR
4510 LET p=4
4520 FOR l=1 TO 7: PRINT AT l,11
...: NEXT l

```

Figura 76 (continua)

```

4530 PRINT AT 1,13;"NOR"
4540 PLOT 112,112
4550 DRAW 38,20,PI/3
4560 DRAW -38,20,PI/3
4570 DRAW 0,-40,-PI/3
4580 PLOT 96,143: DRAW 19,0
4590 PLOT 96,119: DRAW 19,0
4594 CIRCLE 154,132,3: PLOT 158,
132: DRAW 10,0
4599 GO TO 1000
5000 REM EX-OR
5010 LET P=5
5020 FOR L=1 TO 7: PRINT AT L,11
: " : NEXT L
5030 PRINT AT 1,13;"EX-OR"
5040 PLOT 112,112
5050 DRAW 38,20,PI/3
5060 DRAW -38,20,PI/3
5070 DRAW 0,-40,-PI/3
5072 PLOT 104,112: DRAW 0,40,PI/
3
5080 PLOT 96,143: DRAW 19,0
5082 PLOT 96,119: DRAW 19,0
5084 PLOT 151,132: DRAW 16,0
5555 GO TO 1000

```

Figura 76

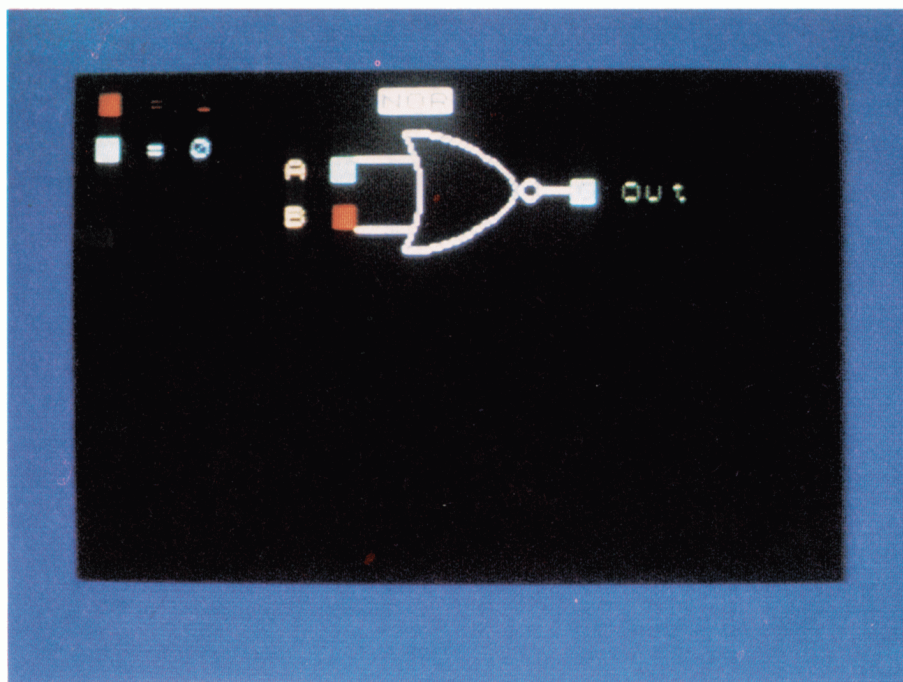


Figura 77

## PROGETTAZIONE DI MULTIVIBRATORE MONOSTABILE CON 555

Una delle applicazioni più interessanti del Computer è nella progettazione di circuiti elettronici. Un esempio pratico è il programma di Figura 78 che permette di calcolare i valori di un Multivibratore Monostabile con il timer 555.

Durante il funzionamento il programma chiede prima il calcolo che si desidera effettuare (Figura 79), quindi dopo aver inserito i dati, fornisce il progetto completo (Figura 80); infine premendo il tasto 'C' si può avere anche una copia su carta del circuito.

Per quanto riguarda i valori di Ra, C, t inseriti, questi possono avere la seguente escursione minima e massima:

Ra	da 1k $\Omega$ a 10000k $\Omega$
C	da 0.00001 $\mu$ F a 1000 $\mu$ F
t	da 0.01ms a 10000ms

```

10 REM
20 REM
30 REM
40 REM
99 BORDER 0: PAPER 0: INK 6: C
LS
100 LET U=PEEK 23675+256*PEEK 2
3676
8110 DATA 56,66,130,130,66,40,23
8,0,0,18,18,18,28,32,64
1120 DATA 16,254,130,130,130,130
,130,130,130,130,130,130,130,130
,254,16,16,66,66,56,16,16,16
120 FOR a=U TO U+39
130 READ d: POKE a,d: NEXT a
140 PLOT 95,136: DRAW 57,0: DRA
U 0,-57: DRAW -57,0: DRAW 0,57
150 PLOT 152,100: DRAW 37,0
160 PLOT 152,116: DRAW 3,0
170 PLOT 124,79: DRAW 0,-23
180 PLOT 118,55: DRAW 10,0
190 PLOT 118,34: DRAW 10,0
200 PLOT 118,53: DRAW 10,0
210 PLOT 67,167: DRAW 0,-96
220 PRINT AT 3,0:CHR$ 140:AT 5,
0:CHR$ 146:AT 6,0:CHR$ 147:AT 7,
0:CHR$ 148:AT 9,0:CHR$ 148
230 PLOT 72,148: DRAW 60,0
240 PLOT 72,118: DRAW 23,0
250 PLOT 72,100: DRAW 23,0
260 PLOT 62,71: DRAW 10,0
270 PLOT 62,66: DRAW 10,0

```

Figura 78 (continua)

```

0080 PLOT 67,67: DRAW 0,-11
0090 PLOT 61,66: DRAW 10,0
0100 PLOT 61,64: DRAW 10,0
0110 PLOT 61,63: DRAW 10,0
0120 PLOT 107,70: DRAW 0,-35
0130 PLOT 107,40: DRAW -10,0
0140 PLOT 116,106: DRAW 0,100
0150 PLOT 132,100: DRAW 0,100
0160 PLOT 114,149: PLOT 116,149:
T 116,149: PLOT 114,147: PLO
T 116,147: BRIGHT 1
0180 PLOT 192,106: DRAW 3,0: DRA
W 0,10: DRAW 10,0: DRAW 0,-12: D
W 0,0
0190 PRINT AT 5,9: "R"
0200 PRINT AT 10,9: "C"
0210 PRINT AT 10,9: "Trig. IN"
0220 PRINT AT 9,24: "OUT"
0230 INK 3
0240 PRINT AT 6,25: "t"
0250 PRINT AT 0,0: "+0"
0260 INK 5
0270 PRINT AT 11,15: "1": AT 11,13
: "2": AT 9,10: "3": AT 9,12: "6": AT
7,10: "5": AT 7,12: "7": AT 5,14: "8"
4
0280 INK 7: BRIGHT 0
0290 PRINT AT 0,14: "555"
0300 PRINT AT 0,13: "Multivibrato
re"
0310 PRINT AT 1,13: "Monostabile
cod 555"
0320 GO SUB 4000
0330 PRINT AT 18,0: INK 5: "Pre
mere per avere"
0340 PRINT AT 19,0: INK 6: "il t
asto 1 t da R"
0350 PRINT AT 20,0: INK 6: "il t
asto 2 R da t e C"
0360 PRINT AT 21,0: INK 6: "il t
asto 3 C da t e R"
0370 PRINT AT 18,20: INK 3: "test
o 555=COPY"
0380 LET K=CODE INKEY$
0390 IF K=67 OR K=99 THEN GO TO
0400
0410 IF K>48 AND K<52 THEN GO TO
(K-48)*1000
0420 GO TO 0400
0430 GO SUB 4000
0440 PRINT AT 20,0: "Tossire R"
0450 : CHR$(144):
: "e"
: "Premere 11111"
0460 INPUT R: IF R<1 OR R>100
00 THEN GO TO 1020

```

Figura 78 (continua)

```

10000 GO SUB 4000
10040 PRINT AT 10,0;"Inserire C ("
10077 INPUT C$ 145;"
10080 ENTER
10080 INPUT t: IF t<0.01 OR t>100.000001 OR C>
10090 THEN GO TO 10090
10090 GO SUB 4000
10090 LET C=t/11.1*Ra*0
10111 GO TO 9999
10090 PRINT AT 10,0;"Inserire t ("
10120 INPUT t: IF t<0.01 OR t>100.000001 OR C>
10130 THEN GO TO 10130
10130 GO SUB 4000
10130 LET C=t/11.1*Ra*0
10150 GO TO 9999
10150 PRINT AT 10,0;"Inserire t ("
10180 INPUT t: IF t<0.01 OR t>100.000001 OR C>
10190 THEN GO TO 10190
10190 GO SUB 4000
10190 LET C=t/11.1*Ra*0
10210 GO TO 9999
10210 PRINT AT 10,0;"Inserire t ("
10240 INPUT t: IF t<0.01 OR t>100.000001 OR C>
10250 THEN GO TO 10250
10250 GO SUB 4000
10250 LET C=t/11.1*Ra*0
10270 GO TO 9999
10270 PRINT AT 10,0;"Inserire Ra ("
10300 INPUT Ra: IF Ra<1 OR Ra>100.000001 OR C>
10310 THEN GO TO 10310
10310 GO SUB 4000
10310 LET C=t/11.1*Ra*0
10330 GO TO 9999
10330 LET C$=""
10350 REM 22 spazi
10410 PRINT AT 10,0;"J$:" AT 10,0;"J$
10420 RETURN
00000 GO SUB 4000: COPY GO TO 9
01
7777 LET X=0: IF Ra<1 OR Ra>1000
0 OR C<0.000001 OR C>1000 OR t<0.
01 OR t>1000000 THEN LET X=1
7778 IF X=1 THEN PRINT AT 14,19,
"UHFLOREI ERRATTI"
7779 IF X=1 THEN PAUSE 100
7780 PRINT AT 14,19;"
": REM 13 spazi

```

Figura 78 (continua)

```

7781 IF X=1 THEN RETURN
7790 LET X$=" k"+CHR$ 144: LET Y
$=" " +CHR$ 145+"F": LET Z$=" m$e
c.
7800 IF Ra>=1000 THEN LET X$=" M
"+CHR$ 144
7802 IF Ra>=1000 THEN LET Ra=Ra/
1000
7808 LET r$=STR$ Ra: IF LEN r$>4
THEN LET r$=r$(1 TO 4)
7810 IF C<1 THEN LET Y$=" nF"
7812 IF C<1 THEN LET C=C*1000
7818 LET c$=STR$ c: IF LEN c$>4
THEN LET c$=c$(1 TO 4)
7820 IF t>=1000 THEN LET z$=" se
c. "
7822 IF t>=1000 THEN LET t=t/100
0
7827 LET t$=STR$ t: IF LEN t$>4
THEN LET t$=t$(1 TO 4)
7829 PRINT AT 5,1;" ";AT 1
2,1;" ";AT 11,22;"
7830 PRINT AT 5,5-LEN r$;r$;X$
7840 PRINT AT 12,5-LEN c$;c$;Y$
7850 PRINT AT 11,22;t$;Z$
8888 RETURN

```

Figura 78

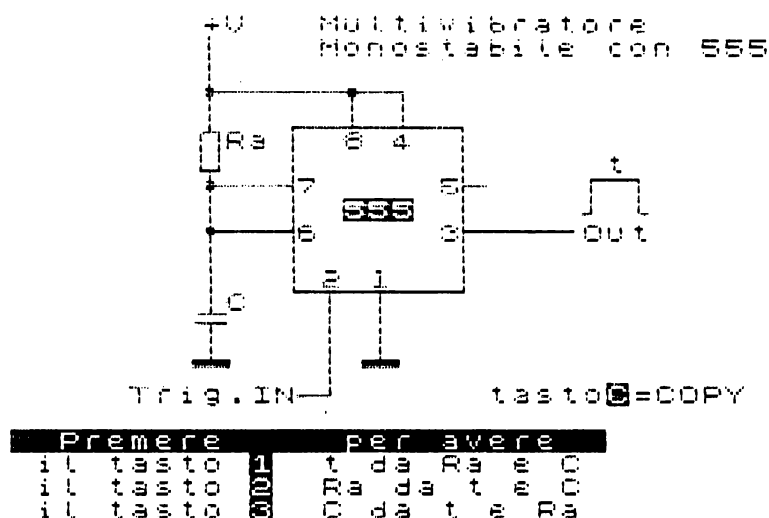
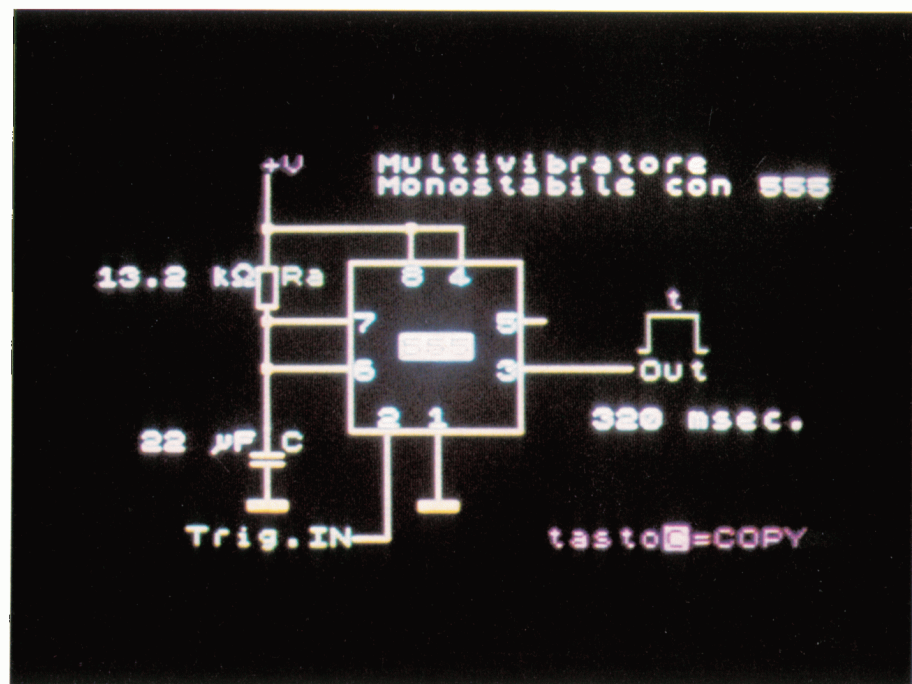


Figura 79



**Figura 80**

## PROGETTAZIONE DI OSCILLATORE CON 555

Anche un Oscillatore con il 555 può essere progettato con un apposito programma (Figura 81). L'uso è simile a quello del programma sul multivibratore monostabile; viene infatti prima chiesto il tipo di calcolo che si vuole effettuare (Figura 82), quindi vengono inseriti i dati chiesti dal computer il quale poco dopo fornisce il circuito finito (Figura 83); anche in questo caso premendo il tasto 'C' si può avere la copia su carta.

Da notare che questo programma calcola anche il 'duty cycle' e visualizza quindi la forma d'onda esatta che si avrà in uscita. I valori Ra, Rb, C, f inseriti possono avere la seguente escursione:

Ra	da 1k $\Omega$ a 10000k $\Omega$
Rb	da 1k $\Omega$ a 10000k $\Omega$
C	da 0.00001 $\mu$ F a 1000 $\mu$ F
f	da 0.01hz a 200000hz

```

10 REM      Oscillatore con 555
20 REM      -----
99 BORDER 0: PAPER 0: INK 6: C
LS
100 LET U=PEEK 23675+256*PEEK 2
36780
110 DATA 55,55,130,130,55,40,23
0,0,0,0,0,16,16,16,28,32,64
1110 DATA 16,254,130,130,130,130
,130,130,130,130,130,130,130,130
,254,16,16,16,55,55,16,16,16
1200 FOR a=0 TO U+30
130 READ d: POKE a,d: NEXT a
140 PLOT 95,136: DRAW 57,0: DRA
150,57: DRAW 57,0: DRAW 0,57
150 PLOT 152,100: DRAW 37,0
160 PLOT 152,116: DRAW 8,0
170 PLOT 154,70: DRAW 0,100
180 PLOT 118,55: DRAW 10,0
190 PLOT 118,54: DRAW 10,0
200 PLOT 118,53: DRAW 10,0
210 PLOT 57,100: DRAW 0,95
220 PLOT 140,0: CHIRP# 140: AT 4,
0: CHIRP# 140: AT 6,
0: CHIRP# 140: AT 8,
0: CHIRP# 140: AT 10
,0: CHIRP# 140
230 PLOT 70,140: DRAW 60,0
240 PLOT 70,104: DRAW 20,0
250 PLOT 70,100: DRAW 0,0
260 PLOT 70,92: DRAW 20,0
270 PLOT 70,71: DRAW 10,0
280 PLOT 70,60: DRAW 10,0

```

Figura 81 (continua)

```

0000 PLOT 07,07: DRADU 0,-11
0000 PLOT 01,05: DRADU 10,0
0000 PLOT 01,04: DRADU 10,0
0100 PLOT 01,03: DRADU 10,0
0400 PLOT 115,135: DRADU 0,12
0500 PLOT 130,130: DRADU 0,12
0600 PLOT 114,140: PLOT 115,140:
T PLOT 116,140: PLOT 114,147: PLO
0000 INK 4: BRIGHT 1
0000 PRINT AT 4,0: "Da"
0000 PRINT AT 7,0: "De"
4000 PRINT AT 10,0: "Ce"
4000 PRINT AT 9,24: "Out"
4400 INK 0
4500 PRINT AT 0,5: "+U"
4700 INK 5
4800 PRINT AT 11,15: "1": AT 10,12
7000 AT 9,13: "3": AT 9,12: "6": AT
7100 "5": AT 5,12: "7": AT 5,14: "8"
4000 INK 7: BRIGHT 0
5000 PRINT AT 8,14: "555"
5000 PRINT AT 0,10: "Oscillatore"
0000 GOTO SUB 4000
0001 PRINT AT 10,0: INK 5: "Pre
mere per avere"
0100 PRINT AT 20,0: INK 6: "il t
asto f da Ra Rb e C"
0200 PRINT AT 21,0: INK 6: "il t
asto C da f Ra e Rb"
0300 PRINT AT 16,20: INK 3: "tast
o"
0400 LET K=CODE INKEY$
0440 IF K=67 OR K=99 THEN GO TO
0500
0500 IF K>48 AND K<51 THEN GO TO
(K-48)*1000
0600 GO TO 940
1000 GO SUB 4000
1010 PRINT AT 20,0: "Inserire Ra
(in k";CHR$ 144:)"
premere ENTER
1020 INPUT Ra: IF Ra<1 OR Ra>100
00 THEN GO TO 1020
1030 GO SUB 4000
1032 PRINT AT 20,0: "Inserire Rb
(in k";CHR$ 144:)"
premere ENTER
1034 INPUT Rb: IF Rb<1 OR Rb>100
00 THEN GO TO 1034
1036 GO SUB 4000
1040 PRINT AT 20,0: "Inserire C (
in ";CHR$ 145: "F)"
premere ENTER

```

Figura 81 (continua)

```

1050 INPUT C: IF C<0.000001 OR C>
1060 THEN GO TO 1050
1060 GO SUB 4000
1070 LET f=1.44/((Ra+2*Rb)*C)*10
00
1080 GO SUB 7777
1111 GO TO 900
2000 GO SUB 4000
2010 PRINT AT 20,0:"Insereire f (
in Hz) "
2020 INPUT f: IF f<0.01 OR f>200
000 THEN GO TO 2020
2030 GO SUB 4000
2040 PRINT AT 20,0:"Insereire Ra
(in k",CHR$(144)," "
2050 INPUT Ra: IF Ra<1 OR Ra>100
000 THEN GO TO 2050
2060 GO SUB 4000
2070 PRINT AT 20,0:"Insereire Rb
(in k",CHR$(144)," "
2080 INPUT Rb: IF Rb<1 OR Rb>100
000 THEN GO TO 2080
2090 GO SUB 4000
2100 LET C=1.44/((Ra+2*Rb)+f/100
0)
2110 GO SUB 7777
2120 GO TO 900
4000 LET J#=""
4010 PRINT AT 19,0:J#;AT 20,0:J#
;AT 21,0:J#
4020 RETURN
6600 GO SUB 4000: COPY GO TO 9
01
7777 LET x=0: IF Ra<1 OR Ra>1000
0 OR Rb<1 OR Rb>10000 OR C<0.0000
01 OR C>10000 OR f<0.01 OR f>20000
00 THEN LET x=1
7778 IF x=1 THEN PRINT AT 14,10:
"VALORI ERRATI"
7779 IF x=1 THEN PAUSE 100
7780 PRINT AT 14,10:"
: REM 13 spazi
7781 IF x=1 THEN RETURN
7785 GO TO 7860
7790 LET x#=""x"+CHR$(144) LET y
#=""x"+CHR$(144) LET y#=""x"+CHR$(
144)+"f": LET z#=""Hz "
7792 IF Rb>=10000 THEN LET h#=""M
"+CHR$(144)
7794 IF Rb>=1000 THEN LET Rb=Rb/
1000
7796 LET m#=STR$ Rb: IF LEN m#>4
THEN LET m#="m$(1 TO 4)

```

Figura 81 (continua)

```

7800 IF Ra >= 1000 THEN LET X$=" M
"+CHR$ 144
7802 IF Ra >= 1000 THEN LET Ra=Ra/
1000
7808 LET r$=STR$ Ra: IF LEN r$>4
THEN LET r$=r$(1 TO 4)
7810 IF C<1 THEN LET u$=" DF"
7812 IF C<1 THEN LET C=C*1000
7818 LET c$=STR$ c: IF LEN c$>4
THEN LET c$=c$(1 TO 4)
7820 IF f >= 1000 THEN LET z$=" KH
N
7822 IF f >= 1000 THEN LET f=f/100
0
7827 LET f$=STR$ f: IF LEN f$>4
THEN LET f$=f$(1 TO 4)
7829 PRINT AT 4,1;" " AT 7
1,1;" " AT 12,1;" " A
t 11,20;"
7830 PRINT AT 4,5-LEN r$;r$;X$
7835 PRINT AT 7,5-LEN u$;u$;h$
7840 PRINT AT 12,5-LEN c$;c$;u$
7850 PRINT AT 11,20; INK 3; BRIG
HT 1; "f="; INK 7; BRIGHT 0; f$;z
$
7855 RETURN
7860 LET dt=Rb/(Ra+0*Rb)
7877 PRINT AT 6,23;" " AT
7,23;" " AT 8,23;" "
7880 PLOT 164,105: DRAW 0,22: DR
AW (1-dt)*50,0: DRAW 0,-22: DRAW
dt*50,0: DRAW 0,22
8888 GO TO 7788

```

Figura 81

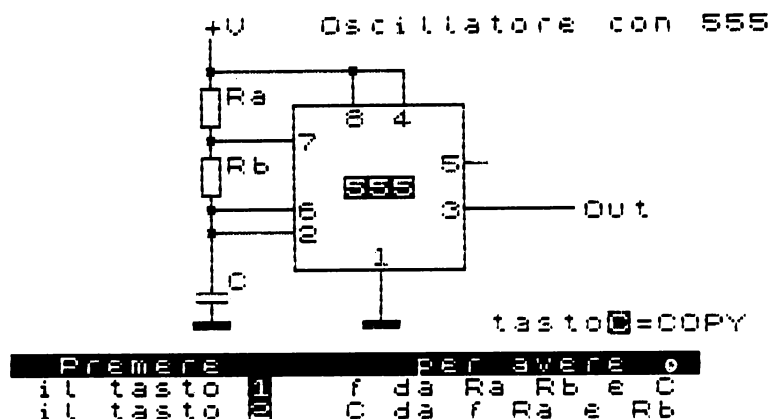


Figura 82

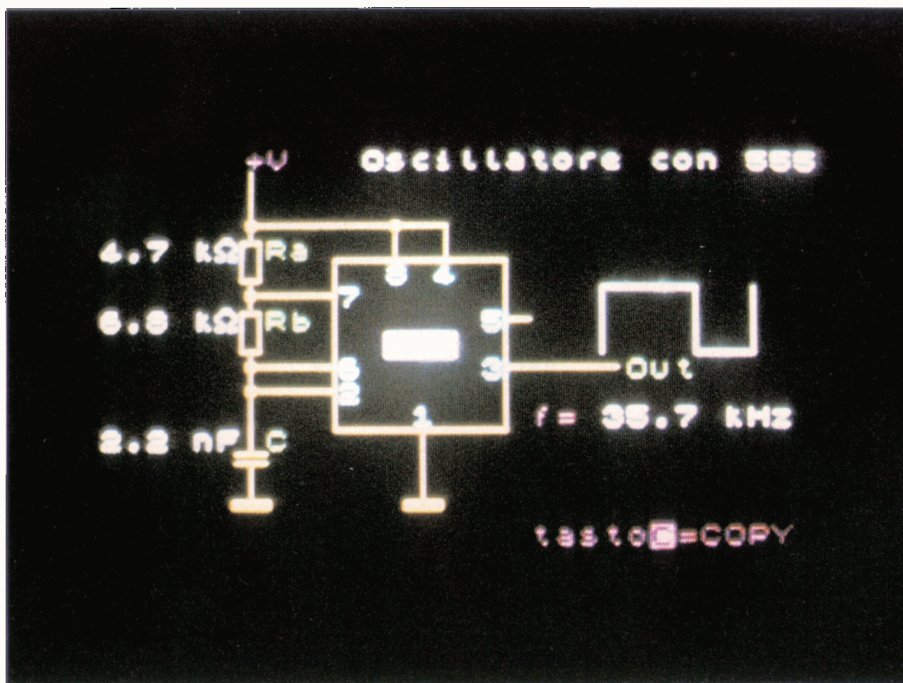


Figura 83

## LEGGE DI OHM

Il terzo programma di progettazione elettronica, ed ultimo di questo Capitolo, è soprattutto un programma didattico sulla legge di OHM (Figura 84). Inserendo infatti i dati come vengono richiesti dal computer si ottiene un risultato come quello di Figura 85.

Da notare che il computer chiede sempre tutti e tre i dati e che bisogna perciò inserire 0 quando il computer chiede il valore che deve essere calcolato.

Le escursioni dei valori di V, R, I inseriti possono essere le seguenti:

V	da 0.000001V a 1000000V
R	da 0.001 $\Omega$ a 1000 M $\Omega$
I	da 0.000001A a 1000A

```
10 REM      Legge di OHM
20 REM      -----
99 BORDER 0: PAPER 0: INK 6: C
LS
100 LET U=PEEK 23675+256*PEEK 2
3675
110 DATA 56,68,130,130,68,40,23
0,0,16,254,130,130,130,130,130,1
30,130,130,130,130,130,130,254,1
0
120 FOR a=U TO U+23
130 READ d: POKE a,d: NEXT a
140 PLOT 83,132: DRAW 0,16: DRA
W 56,0: DRAW 0,-40: DRAW -56,0:
DRAW 0,16
150 PRINT AT 5,17:CHR$ 145;AT 6
,17:CHR$ 146
160 PLOT 71,131: DRAW 24,0
170 PLOT 77,137: DRAW 12,0
180 PLOT 77,126: DRAW 12,0
190 PLOT 77,135: DRAW 12,0
200 PLOT 74,138: DRAW 0,0
210 PLOT 71,139: DRAW 0,0
220 PLOT 130,154: DRAW -36,0
230 PLOT 130,154: DRAW -3,3
240 PLOT 130,154: DRAW -3,-3
250 INK 5: BRIGHT 1
260 PRINT AT 7,11: "U";AT 5,18: "
R";AT 1,13: "I"
270 INK 3
280 PRINT AT 2,21: "U_";AT 4,21:
"R_";AT 6,21: "I_"
290 INK 7: BRIGHT 0
300 LET c=0
```

Figura 84 (continua)

```

310 INPUT "Inserire U (in Volt)
oppure 0 e premere ENTER ";v
311 IF v=0 THEN GO TO 320
315 IF v<1E-6 OR v>1E6 THEN GO
TO 310
320 INPUT "Inserire I (in Amper
) oppure 0 e premere ENTER ";i
322 IF i=0 THEN GO TO 330
325 IF i<1E-6 OR i>1E3 THEN GO
TO 320
330 INPUT "Inserire R (in Ohm)
oppure 0 e premere ENTER ";r
333 IF r=0 THEN GO TO 340
335 IF r<1E-3 OR r>1E9 THEN GO
TO 330
340 IF v<>0 THEN LET c=c+1
342 IF i<>0 THEN LET c=c+1
344 IF r<>0 THEN LET c=c+1
350 IF c<>2 THEN GO TO 777
360 IF v=0 THEN LET v=i*r
370 IF r=0 THEN LET r=v/i
380 IF i=0 THEN LET i=v/r
390 IF v<1E-6 OR v>1E6 THEN GO
TO 777
400 IF i<1E-6 OR i>1E3 THEN GO
TO 777
410 IF r<1E-3 OR r>1E9 THEN GO
TO 777
420 LET x$="U": LET y$=CHR$ 144
: LET z$="A"
430 IF v<1 THEN LET x$="mU"
440 IF v<1 THEN LET v=v*1000
450 IF v>=1000 THEN LET x$="kU"
460 IF v>=1000 THEN LET v=v/100
0
470 IF r>=1E6 THEN LET y$="M"+C
HR$ 144
480 IF r>=1E6 THEN LET r=r/1E6
490 IF r>=1000 THEN LET y$="k"+
CHR$ 144
500 IF r>=1000 THEN LET r=r/100
0
510 IF i<1 THEN LET z$="mA"
520 IF i<1 THEN LET i=i*1000
530 LET v$=STR$ v: IF LEN v$>5
THEN LET v$=v$(1 TO 5)
540 LET r$=STR$ r: IF LEN r$>5
THEN LET r$=r$(1 TO 5)
550 LET i$=STR$ i: IF LEN i$>5
THEN LET i$=i$(1 TO 5)
560 PRINT AT 2,23;" ";AT
4,23;" ";AT 6,23;" "
570 PRINT AT 2,23;v$; INK 4;x$
580 PRINT AT 4,23;r$; INK 4;y$

```

Figura 84 (continua)

```

590 PRINT AT 6,23;i$; INK 4;z$
600 GO TO 300
777 PRINT AT 11,7;"VALORI ERRAT
I"
888 PAUSE 100: PRINT AT 11,7;"
999 GO TO 300

```

Figura 84

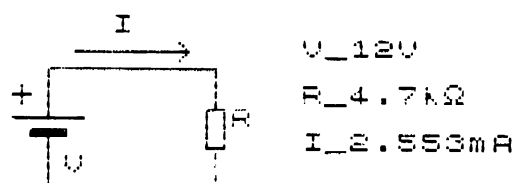


Figura 85

# UTILITY

## ROUTINE 'PAINT' IN BASIC

Una istruzione di cui si sente la mancanza quando si usa l'alta risoluzione grafica dello Spectrum per disegnare, è l'istruzione 'PAINT' (o FILL su altri computer) che serve per colorare l'interno di figure chiuse senza dover inventare ogni volta una routine apposita. La routine PAINT (Figura 86) può essere usata a tale scopo; basta infatti inserire nelle variabili 'xp' ed 'yp' le coordinate di un punto all'interno della figura da colorare (il più possibile al centro) e nella variabile 'ip' il colore desiderato, e quindi chiamare la routine con l'istruzione 'GOSUB 7000'.

Naturalmente se la figura da colorare è molto complessa oppure rimangono degli spazi non colorati, è possibile indirizzare più punti all'interno della figura e chiamare altrettante volte la routine.

È importante che all'interno della figura non vi siano già delle scritte o altro che possa fermare il pennello della routine, inoltre nel programma non devono essere utilizzate le variabili 'xp', 'yp', 'ip', 'o' e 'q' che vanno riservate alla routine stessa.

In Figura 86 insieme alla routine è compreso anche un breve programma esempio che disegna due figure (Figura 87) e le colora (Figura 88) mostrando in pratica il funzionamento di 'PAINT'.

```
10 REM  programma esempio
20 BORDER 0: PAPER 0: CLS
30 INK 7: PLOT 0,0: DRAW 255,0
: DRAW 0,175: DRAW -255,0: DRAW
0,-175
40 REM  CERCHIO azzurro
50 INK 5: CIRCLE 50,50,30
60 LET ip=5: LET xp=50: LET yp
=50: GO SUB 7000
70 REM  QUADRATO giallo
80 INK 6: PLOT 150,100: DRAW 4
5,0: DRAW 0,45: DRAW -45,0: DRAW
0,-45
90 LET ip=6: LET xp=170: LET y
p=120: GO SUB 7000
100 STOP
999 REM  -----
7000 REM  routine PAINT in Basic
```

Figura 86 (continua)

```

7005 INK ip
7010 FOR q=xp TO 255
7020 FOR o=yp TO 175
7030 PLOT q,o
7040 IF POINT (q,o+1)=0 THEN NEX
T o
7050 IF POINT (q+1,yp)=0 THEN NE
XT q
7060 FOR q=xp TO 255
7070 FOR o=yp-1 TO -175 STEP -1
7080 PLOT q,o
7090 IF POINT (q,o-1)=0 THEN NEX
T o
7100 IF POINT (q+1,yp-1)=0 THEN
NEXT q
7110 FOR q=xp-1 TO -255 STEP -1
7120 FOR o=yp TO 175
7130 PLOT q,o
7140 IF POINT (q,o+1)=0 THEN NEX
T o
7150 IF POINT (q-1,yp)=0 THEN NE
XT q
7160 FOR q=xp-1 TO -255 STEP -1
7170 FOR o=yp-1 TO -175 STEP -1
7180 PLOT q,o
7190 IF POINT (q,o-1)=0 THEN NEX
T o
7200 IF POINT (q-1,yp-1)=0 THEN
NEXT q
7222 RETURN

```

Figura 86

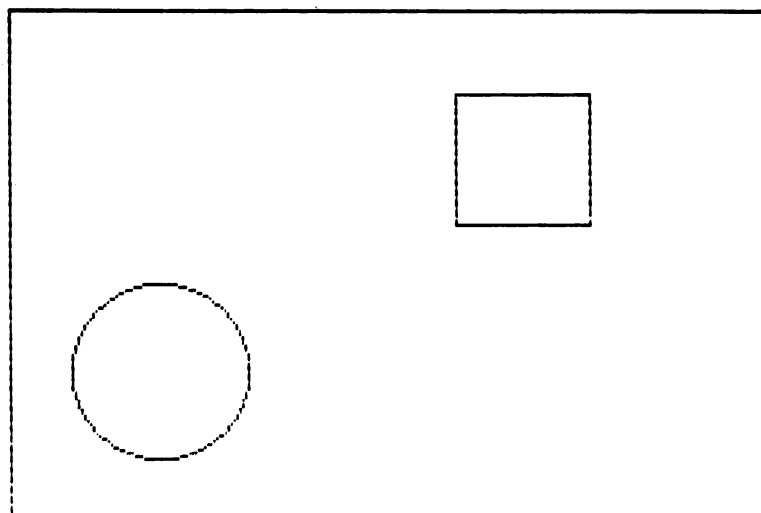
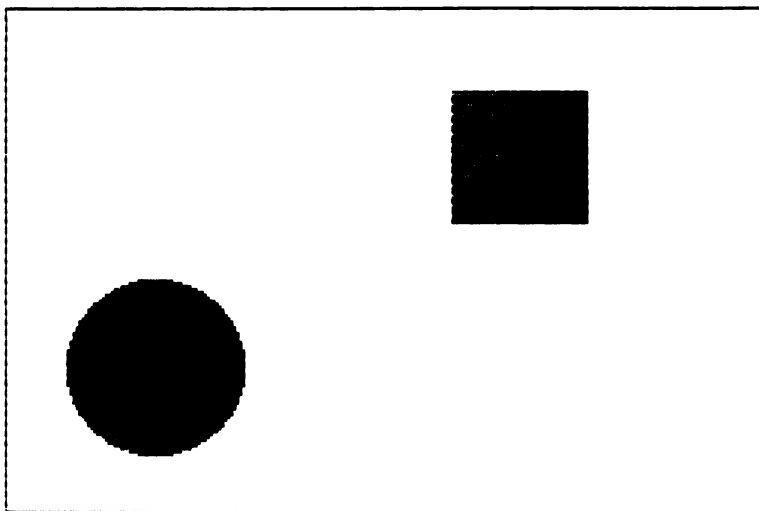


Figura 87



**Figura 88**

## **NOTE SULLE ROUTINES 'SCROLL'**

Le prossime quattro utilities sono le routines 'SCROLL'.

Si tratta di quattro routines in linguaggio macchina, le quali, ogni volta che vengono chiamate da una istruzione 'USR' spostano di un carattere tutto il contenuto dello schermo rispettivamente verso sinistra, verso destra, verso l'alto o verso il basso.

Tali routines sono molto utili, inserite in appositi programmi, per estendere le dimensioni virtuali dello schermo, che diventa una finestra su uno schermo più ampio non visibile il quale può avere anche per esempio 50 righe per 80 colonne.

Altri usi sono per spostare in un dato senso una serie di dati o delle immagini (una versione modificata degli Scroll a sinistra ed a destra è usata per esempio nel gioco 'Tiro alla fune', in questo libro, per spostare rapidamente tutta la parte superiore dello schermo a destra o a sinistra secondo le necessità del programma).

Da notare che lo Spectrum ha già lo Scroll verso l'alto che però, a differenza della routine in linguaggio macchina, si blocca ogni 22 linee visualizzando la scritta 'scroll?' che chiede se lo scroll deve continuare.

Importante: in tutte e quattro le routines (che comprendono anche un esempio di funzionamento) le istruzioni in linguaggio macchina sono caricate nella zona di memoria riservata ai caratteri speciali, naturalmente se tale zona serve al programma utilizzatore, la o le routines possono essere memorizzate tra la fine dello stack e l'inizio della zona degli UDG usando il comando CLEAR nel modo indicato al capitolo 26 del manuale dello Spectrum.

## SCROLL A SINISTRA

La prima routine (Figura 89) che è lunga 24 Bytes, scrolla tutto lo schermo verso sinistra di un carattere ogni volta che il programma incontra un'istruzione tipo: LET s=USR n dove 'n' è l'indirizzo della RAM dove comincia la routine e che nel programma esempio è (u+144).

```

66 REM ROUT. SCROLL A SINISTRA
77 REM -----
99 BORDER 0: PAPER 0: INK 6: C
LS
100 DATA 6,192,33,1,64,229,209,
27,197,1,31,0,237,176,43,54,0,19
,35,35,193,16,241,201
104 LET U=PEEK 23675+256*PEEK 2
3676
110 FOR a=U+144 TO U+167
120 READ d: POKE a,d: NEXT a
999 REM -----

1000 REM      programma esempio
1010 REM -----
1015 FOR x=1 TO 100
1020 FOR a=0 TO 359
1030 PRINT AT 10+9*SIN (PI/9*a),
31; " "
1040 LET s=USR (U+144)
1044 PAUSE S
1050 NEXT a
1060 NEXT x
```

Figura 89

## SCROLL A DESTRA

Anche questa routine (Figura 90) è lunga 24 bytes e scrolla tutto lo schermo verso destra di un carattere ad ogni istruzione tipo LET s=USR n dove 'n' è di nuovo l'indirizzo di partenza della routine nella RAM, nell'esempio quindi (u+144).

```

66 REM      rout. SCROLL H DESTRA
77 REM      -----
99 BORDER 0: PAPER 0: INK 7: C
LS
100 DATA 6,192,33,254,67,229,20
9,19,197,1,31,0,237,184,35,54,0,
27,43,43,193,16,241,201
104 LET U=PEEK 23675+256*PEEK 2
3676
110 FOR a=U+144 TO U+167
120 READ d: POKE a,d: NEXT a
999 REM      -----

1000 REM      programma esempio
1010 REM      -----
1011 LET t=1
1015 FOR x=1 TO 100
1020 FOR a=0 TO 21
1030 IF t=1 THEN PRINT AT a,0;".
"
1033 IF t=-1 THEN PRINT AT 21-a,
0;".
1040 LET s=USR (U+144)
1044 PAUSE 5
1050 NEXT a
1055 LET t=t*-1
1060 NEXT x
```

Figura 90

## SCROLL IN ALTO

Altra routine (Figura 91), lunga 121 Bytes, questa volta per lo scroll in alto di un carattere ad ogni istruzione LET s=USR 'indirizzo routine', indirizzo che nell'esempio è (u+47).

```

66 REM      ROUT. SCROLL IN ALTO
77 REM      -----
99 BORDER 0: PAPER 0: INK 2: C
LS
100 DATA 6,8,33,32,64,17,0,64,1
97,1,224,0,237,176,62,32,35,19,6
1,32,251,193,16,240
102 DATA 6,8,33,0,72,17,224,64,
97,1,32,0,237,176,62,224,35,19,
61,32,251,193,16,240
104 DATA 6,8,33,32,72,17,0,72,1
97,1,224,0,237,176,62,32,35,19,6
1,32,251,193,16,240
108 DATA 6,8,33,32,80,17,0,80,1
97,1,224,0,237,176,62,32,35,19,6
1,32,251,193,16,240
116 DATA 201
120 LET U=PEEK 23675+256*PEEK 2
3675
130 FOR a=U+47 TO U+167
140 READ d: POKE a,d
150 NEXT a
999 REM      -----

1000 REM      programma esempio
1010 REM      -----
1020 FOR a=1 TO 1000
1030 PRINT AT 21,INT (RND*32);"■"
1040 LET s=USR (U+47)
1044 PAUSE 5
1050 NEXT a

```

Figura 91

## SCROLL IN BASSO

Ultima routine (Figura 92), lunga 140 Bytes, scolla verso il basso di un carattere le prime 22 linee sempre ad ogni istruzione tipo LET s=USR 'indirizzo routine', istruzione che nell'ultimo esempio (che produce una pioggia di Dollari e Sterline) è USR u.

```

66 REM      ROUT. SCROLL IN BASSO
77 REM      -----
99 BORDER 0: PAPER 0: INK 4: C
LS
100 DATA 6,8,33,223,87,17,255,8
7,197,1,2024,0,2037,104,62,32,43,2
7,61,32,2051,103,16,240
101 DATA 17,2024,0,66,8,33,192,80
62,32,54,0,35,81,32,250,25,16,2
45
102 DATA 6,8,33,223,87,17,31,87
,197,1,32,10,2037,104,62,204,43,27
,61,32,2051,103,16,240
104 DATA 6,8,33,223,87,17,255,7
7,197,1,2024,0,2037,104,62,32,43,2
7,61,32,2051,103,16,240
106 DATA 6,8,33,223,87,1,17,31,70
,197,1,32,10,2037,104,62,204,43,27
,61,32,2051,103,16,240
108 DATA 6,8,33,223,87,1,17,255,7
7,197,1,2024,0,2037,104,62,32,43,2
7,61,32,2051,103,16,240
110 DATA 17,2024,0,66,8,33,0,64,6
2,32,54,0,35,81,32,250,25,16,245
112 DATA 101
120 LET U=PEEK 23675+256*PEEK 2
3675
130 FOR a=U TO U+156
140 READ d: POKE a,d
150 NEXT a
999 REM      -----

1000 REM      programma esempio
1010 REM      -----
1020 FOR a=1 TO 1000
1030 PRINT AT 0,INT (RND*32);CHR
$(36+(60*INT (RND*2)))
1040 LET s=USR u
1044 PRUSR 20
1050 NEXT a

```

Figura 92

## RENUMBER

Un programma di Renumber è molto utile nella programmazione per riordinare automaticamente tutti i numeri di linea appena cominciano ad essere un po' confusi.

La versione di Figura 93 realizza tale riordino escluso i numeri dopo GOTO e GO-SUB che vanno modificati singolarmente.

Il programma va fatto girare con un RUN 9020 o con un GOTO 9020. Dopo di ciò appariranno le scritte in Input che chiedono l'inserimento prima del numero da cui iniziare la rinumerazione (per esempio 3), poi del nuovo numero di linea iniziale (per esempio 100) ed infine della distanza desiderata tra linea e linea (per esempio 10). Naturalmente è possibile effettuare anche più di una rinumerazione per esempio per avere le prime 16 linee rinumerate con 100, 110, 120, ecc... e le altre 27 linee rinumerate con 1000, 1050, 1100, ecc.

La Figura 94 mostra come esempio un programma che ha bisogno di essere rinumerato, mentre la Figura 95 mostra una delle possibili rinumerazioni dello stesso ottenuta con il programma 'Renumber' appena descritto.

Per comodità il Renumber può essere memorizzato su una cassetta con il comando SAVE "renumber" e caricato in memoria quando occorre tramite il comando MERGE "renumber" che consente di inserire il Renumber nella RAM senza cancellare il programma che state scrivendo. Naturalmente per usare il Renumber il programma da rinumerare non deve usare le linee da 9000 a 9140, inoltre vanno riservate al funzionamento del Renumber le variabili 'im', 'nir', 'nli', 'dtl', 'nx'.

```
9000 REM          RENUMBER
9010 STOP
9020 INPUT "Inserire N° linea ini
zio renumber",,nir
9030 INPUT "Inserire nuovo N° lin
ea iniziale",,nli
9040 INPUT "Inserire distanza tr
a linee",,dtl
9050 LET im=PEEK 23635+256*PEEK
23636-1
9060 LET nx=PEEK (im+1)*256+PEEK
(im+2)
9070 IF nx<nir THEN GO TO 9120
9080 IF nx=9000 THEN GO TO 9140
9090 POKE (im+1),INT (nli/256)
9100 POKE (im+2),nli-256*INT (nli
/256)
9110 LET nli=nli+dtl
9120 LET im=im+4+PEEK (im+3)+256
*PEEK (im+4)
9130 GO TO 9060
9140 LIST
```

Figura 93

```

1 REM Programma esempio
3 REM per prova RENUMBER
7 REM linee 1---8888
11 REM -----
20 LET x=235*678
43 FOR k=1 TO 100
44 PRINT "yt";
45 NEXT k
2000 PRINT "ffff"
2005 CIRCLE 40,40,10
5500 DATA 10,4,7,68,40,99,0,3
9000 GO SUB 3333
10000 READ d: PRINT d
15000 PLOT 5,5: DRAW 80,80
20000 STOP
30000 RANDOMIZE
35000 LET g=RND*10
40000 PRINT PEEK 23678
55000 LET r$="ffffgjtU"
60000 PRINT LEN r$
75000 RETURN
80000 REM fine programma esempio

```

Figura 94

```

100 REM Programma esempio
110 REM per prova RENUMBER
120 REM linee 1---8888
130 REM -----
140 LET x=235*678
150 FOR k=1 TO 100
160 PRINT "yt";
170 NEXT k
180 PRINT "ffff"
190 CIRCLE 40,40,10
2000 DATA 10,4,7,68,40,99,0,3
2100 GO SUB 3333
2200 READ d: PRINT d
2300 PLOT 5,5: DRAW 80,80
2400 STOP
2500 RANDOMIZE
2600 LET g=RND*10
2700 PRINT PEEK 23678
2800 LET r$="ffffgjtU"
2900 PRINT LEN r$
3000 RETURN
310 REM fine programma esempio

```

Figura 95

## LETTORE MEMORIA RAM/ROM

Un'altra utility interessante è il lettore della memoria del Computer (Figura 96) con il quale si può leggere il contenuto di una parte qualsiasi della ROM o della RAM a partire da un indirizzo inserito all'inizio.

La Figura 97 mostra un esempio di lettura della RAM ed in particolare proprio di una zona nella quale è scritto il programma di Figura 96; come si vede il lettore memoria visualizza l'indirizzo, il contenuto decimale ed il corrispondente carattere di ciascun Byte esaminato. Se il carattere ha un codice inferiore a 32 esce la scritta '—', mentre, se il carattere è uno di quelli grafici definibili, viene visualizzata la lettera corrispondente preceduta da una 'G' inversa.

Il programma ha un funzionamento continuo premendo ENTER ogni volta che esce la scritta 'scroll?' e può essere fermato con un Break. Un'altra possibilità è quella di numerare il programma per esempio con i numeri da 8000 a 8140 in modo da caricarlo da una cassetta nella RAM con un comando 'MERGE' insieme ad un altro programma che si vuole esaminare; in questo caso il lettore memoria va fatto girare con un RUN 8000 o con un GOTO 8000.

Le variabili riservate al funzionamento di questo programma sono: 'ip', 'w\$', 'w'.

```
10 REM LETTORE MEMORIA ROM/ROM
20 REM -----
99 BORDER 0: PAPER 0: INK 7: C
LS
100 INPUT "Inserire indirizzo d
i partenza ( 0--65535 ) e preme
re ENTER",ip
110 IF ip<0 OR ip>65535 THEN GO
TO 100
120 FOR w=ip TO 65535-ip
130 IF w=ip OR w-INT (w/10)*10=
0 THEN PRINT " Indirizzo Conten
uto Carattere"
140 LET w$=" " + STR$ w+"
150 PRINT w$(1 TO 15);
160 LET w$=STR$ PEEK w+"
170 PRINT w$(1 TO 8);
180 IF PEEK w>31 AND PEEK w<144
OR PEEK w>164 THEN PRINT CHR$ P
EEK w
188 IF PEEK w>143 AND PEEK w<16
5 THEN PRINT "G ";CHR$ (PEEK w-7
9)
190 IF PEEK w<32 THEN PRINT "--
-
200 NEXT w
```

Figura 96



## CONVERSIONE DECIMALE/ESADECIMALE ED INVERSO

Spesso le routines in linguaggio macchina usate in alcuni programmi vengono caricate in memoria a partire da stringhe o istruzioni DATA che contengono i loro codici scritti in esadecimale.

In questo ed in molti altri casi è comodo disporre di un programma come quello di Figura 98 che converte numeri decimali in esadecimali e viceversa. Il programma può convertire numeri decimali da 0 a 16777215 e numeri esadecimali da 0 a FFFFFFF con una capacità quindi di 24 Bit sufficiente perciò per essere usato anche nella programmazione dei più recenti microprocessori a 16 Bit.

```

10 REM CONVERSIONE DECIMALE/
20 REM ESADECIMALE E INVERSO
30 REM -----
99 BORDER 0: PAPER 0: INK 7: C
LS
100 PRINT AT 0,0;"tasto Q per i
nserire N° Decimale"
110 PRINT AT 2,0;"tasto H per i
ns. N° Esadecimale"
120 LET t=CODE INKEY$
130 IF t=68 OR t=100 THEN GO TO
150
140 IF t=72 OR t=104 THEN GO TO
270
150 GO TO 120
160 INPUT "Inserire N° Decim. (0.
/16777215.) e premere ENTER ";
d
170 LET f=d
180 IF d<0 OR d>(2↑24-1) THEN G
O TO 160
190 LET c$="0123456789ABCDEF"
200 LET h$=""
210 FOR a=5 TO 0 STEP -1
220 LET h$=h$+c$(1+INT (f/16↑a))
)
230 LET f=f-(INT (f/16↑a)*16↑a)
240 NEXT a
241 LET s=1
242 FOR a=1 TO LEN h$
244 IF h$(a)="0" THEN LET s=s+1
246 IF CODE h$(a)>48 THEN GO TO
250
248 NEXT a
250 LET h$=h$(s TO ); GO TO 370
270 INPUT "Inserire N° Esadecim.
(0./FFFFFF.) e premere ENTER ";
LINE h$

```

Figura 98 (continua)

```

277 IF LEN h$ > 6 THEN GO TO 270
280 LET x$ = h$
290 FOR a = 1 TO LEN x$
299 IF CODE x$(a) > 57 AND CODE x$(a) < 65 THEN GO TO 270
300 IF CODE x$(a) > 96 AND CODE x$(a) < 103 THEN LET x$(a) = CHR$ ((CODE x$(a)) - 32)
310 IF CODE x$(a) > 84 AND CODE x$(a) < 71 THEN LET x$(a) = CHR$ ((CODE x$(a)) - 7)
320 IF CODE x$(a) < 48 OR CODE x$(a) > 63 THEN GO TO 270
330 NEXT a
333 LET d = 0
340 FOR a = 1 TO LEN x$
350 LET d = d + (CODE x$(a) - 48) * 16a
360 NEXT a
370 CLS
377 PRINT AT 6,0; INK 6; "N° Dec
imale"; d
380 PRINT AT 6,0; INK 5; "N° Ess
decimale"; h$
399 GO TO 100

```

Figura 98



# ANIMAZIONI

## VIAGGIO SPAZIALE

La più bella delle animazioni è senzaltro il Viaggio Spaziale (Figura 99) che simula appunto il viaggio di una astronave nello spazio con le stelle che dal centro dell'immagine fuggono verso l'esterno. Prima di ottenere l'effetto occorre aspettare circa venti minuti durante i quali il programma esegue una serie di operazioni che servono al funzionamento poi dell'animazione. Importante: Per settare correttamente gli attributi di colore il programma funziona con INK 0 e PAPER 0 durante l'esecuzione delle linee da 500 a 1000 per cui se fermate il programma in questa fase può sembrare che si sia cancellato, in realtà il programma c'è e basta eseguire in modo diretto l'istruzione 'INK 7' per rivederlo.

```

10 REM VIAGGIO SPAZIALE
20 REM -----
77 BORDER 1: PAPER 0
88 INK 7: CLS
99 CLEAR 30000
100 DATA 33,86,117,17,0,88,1,19
2,2,237,176,201,33,23,120,17,0,88
0,1,192,2,237,176,201,33,216,122
,17,0,88,1,192,2,237,176,201,0
120 FOR a=30001 TO 30037
130 READ d: POKE a,d: NEXT a
140 LET attr=22528
150 GO TO 500
180 INK 7: CLS
182 FOR a=0 TO 360 STEP 4
190 LET k=PI/180*a
200 LET l=10+RND*10
210 FOR s=1 TO 5
230 LET x=128+l*COS k
240 LET y=88+l*SIN k
250 IF x<0 OR x>255 THEN GO TO
260 IF y<0 OR y>175 THEN GO TO
270 IF RND<0.2 THEN PLOT INK 6;
X,Y
280 LET l=l+30
290 NEXT s
300 NEXT a
300 LET k=USR 30001

```

Figura 99 (continua)

```

310 PAUSE 7
320 LET k=USR 30013
330 PAUSE 7
340 LET k=USR 30025
350 PAUSE 7
444 GO TO 300
500 INK 0: CLS
502 FOR r=0.5 TO 20 STEP 4
510 FOR a=0 TO 360 STEP (10/r)
520 LET k=PI/180*a
530 LET l=10.5+r*SIN k
540 LET c=15.5+r*COS k
550 IF l<0 OR l>21 THEN GO TO 5
60 IF c<0 OR c>31 THEN GO TO 5
60 PRINT AT l,c: INK 7;"■"
600 NEXT a: NEXT r
600 LET mem=30038
600 FOR a=0 TO 703
610 POKE (mem+a),PEEK (attr+a)
620 NEXT a
630 CLS
700 FOR r=1.5 TO 20 STEP 4
710 FOR a=0 TO 360 STEP (10/r)
720 LET k=PI/180*a
730 LET l=10.5+r*SIN k
740 LET c=15.5+r*COS k
750 IF l<0 OR l>21 THEN GO TO 7
80 IF c<0 OR c>31 THEN GO TO 7
80 PRINT AT l,c: INK 7;"■"
800 NEXT a: NEXT r
800 LET mem=30743
800 FOR a=0 TO 703
810 POKE (mem+a),PEEK (attr+a)
820 NEXT a
830 CLS
900 FOR r=2.5 TO 20 STEP 4
910 FOR a=0 TO 360 STEP (10/r)
920 LET k=PI/180*a
930 LET l=10.5+r*SIN k
940 LET c=15.5+r*COS k
950 IF l<0 OR l>21 THEN GO TO 9
960 IF c<0 OR c>31 THEN GO TO 9
970 PRINT AT l,c: INK 7;"■"
980 NEXT a: NEXT r
990 LET mem=31448
992 FOR a=0 TO 703
994 POKE (mem+a),PEEK (attr+a)
996 NEXT a
998 CLS
1000 GO TO 180

```

Figura 99

## LANCIO SPACE SHUTTLE

Il programma (Figura 100) simula il conteggio alla rovescia e la partenza della navetta spaziale americana (o Space Shuttle). Dopo la partenza, la navetta (che è riprodotta in maniera molto realistica con i caratteri definibili) sale, scomparendo nella parte alta dello schermo. In Figura 101 si vede la navetta all'inizio del conteggio alla rovescia.

```

10 REM LANCIO SPACE SHUTTLE
20 REM -----
100 BORDER 0: PAPER 0: INK 7: C
L$
110 DATA 0,0,0,0,0,0,2,7,96,96,
240,144,80,208,208,208
120 DATA 6,14,14,14,14,14,14,14,14
,208,208,208,208,208,208,208,208
130 DATA 14,30,31,62,124,96,193
,128,208,208,240,224,192,192,224
,0
140 LET U=PEEK 23675+256*PEEK 2
3676
150 FOR a=U TO U+47
160 READ d: POKE a,d: NEXT a
170 LET a$=CHR$ 16+CHR$ 7+CHR$
144+CHR$ 16+CHR$ 2+CHR$ 146
180 LET b$=CHR$ 16+CHR$ 7+CHR$
146+CHR$ 16+CHR$ 2+CHR$ 147
190 LET c$=CHR$ 16+CHR$ 7+CHR$
148+CHR$ 16+CHR$ 2+CHR$ 149
200 PRINT AT 18,20: INK 5: "☒": A
T 19,20: INK 5: "☒☒": AT 20,20: IN
K 5: "☒☒"
210 PRINT AT 21,0: INK 4: "-----"
220 PRINT AT 18,18: a$: AT 19,18:
b$: AT 20,18: c$
230 RANDOMIZE
240 FOR s=1 TO 30
250 LET l=1+INT (RND*14)
260 LET c=INT (RND*32)
270 IF c>15 AND c<21 THEN GO TO
280
280 PRINT AT l,c: INK 2+INT (RN
D*6): ". "
290 NEXT s
300 PRINT AT 19,26: INK 7: "00:0
310 FOR t=9 TO 0 STEP -1
320 PAUSE 45
330 PRINT AT 19,30: INK 7: t
340 NEXT t

```

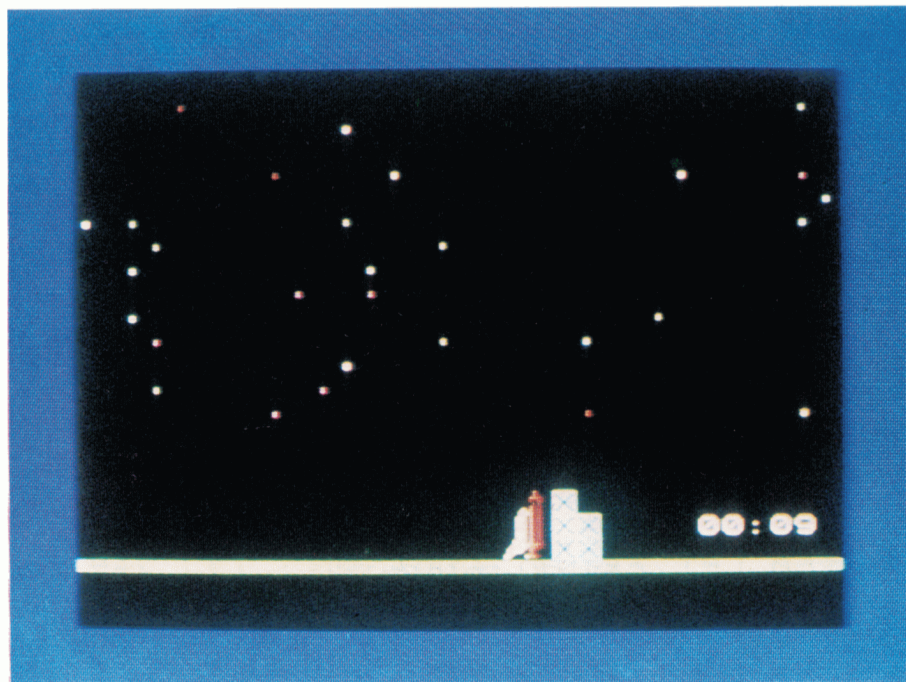
Figura 100 (continua)

```

350 FOR l=17 TO 0 STEP -1
360 LET g=l+1: LET h=l+2: LET s
=l+3
370 PRINT AT l,18;a$;AT g,18;b$
;AT h,18;c$;AT s,18;" "
380 NEXT l
390 PRINT AT 0,18;b$;AT 1,18;c$
;AT 2,18;" "
400 PRINT AT 0,18;c$;AT 1,18;"
"
410 PRINT AT 0,18;" "

```

**Figura 100**



**Figura 101**

## ANIMAZIONE ESPLOSIONE

In Figura 102 il programma che disegna un asteroide (Figura 103) il quale viene colpito e ridotto in tanti pezzi da un missile.

```

10 REM ANIMAZIONE ESPLOSIONE
20 REM -----
100 BORDER 0: PAPER 0: CLS
110 DATA 128,66,49,47,22,30,95,
50,34,4,34,81,4,41,16,133,136,4,
80,130,32,82,8,162
120 DATA 18,4,82,137,18,165,0,1
7,80,132,18,164,73,32,132,16
130 LET U=PEEK 23675+255*PEEK 2
3676
140 FOR a=U TO U+39
150 READ d: POKE a,d: NEXT a
160 LET a$=CHR$ 16+CHR$ 6+CHR$
144
170 LET b$=CHR$ 16+CHR$ 2+CHR$
145+CHR$ 146
180 LET c$=CHR$ 16+CHR$ 2+CHR$
147+CHR$ 148
190 FOR a=0 TO 6 STEP 0.5
200 INK 5: CIRCLE 56,128,a+1
202 CIRCLE 57,128,a
210 NEXT a
220 PRINT AT 21,22;a$
230 PAUSE 100
240 FOR a=20 TO 6 STEP -1
250 PRINT AT a+1,a+2;" ";AT a,a
+1;a$
260 PAUSE 1
270 NEXT a
280 PRINT AT 5,6;b$;AT 6,6;c$
290 BEEP 2,-40
300 PRINT AT 5,6;" ";AT 6,6;" "

```

Figura 102



Figura 103

## CALEIDOSCOPIO

Nel programma 'Caleidoscopio' (Figura 104) dodici diversi oggetti di colori casuali appaiono e scompaiono continuamente e velocemente sul video, simmetricamente nelle quattro zone in cui è diviso (Figura 105).

```

10 REM      CALEIDOSCOPIO
20 REM      -----
100 BORDER 0: PAPER 0: CLS
110 DATA 60,60,0,231,231,0,60,6
0,231,231,231,0,0,231,231
120 DATA 36,36,231,0,0,231,36,3
6,0,126,66,30,90,66,126,0
130 DATA 24,36,66,120,120,66,36
,24,0,126,126,24,24,126,0
140 DATA 126,126,0,60,60,0,126,
126,66,219,36,66,66,36,219,66

```

Figura 104 (continua)

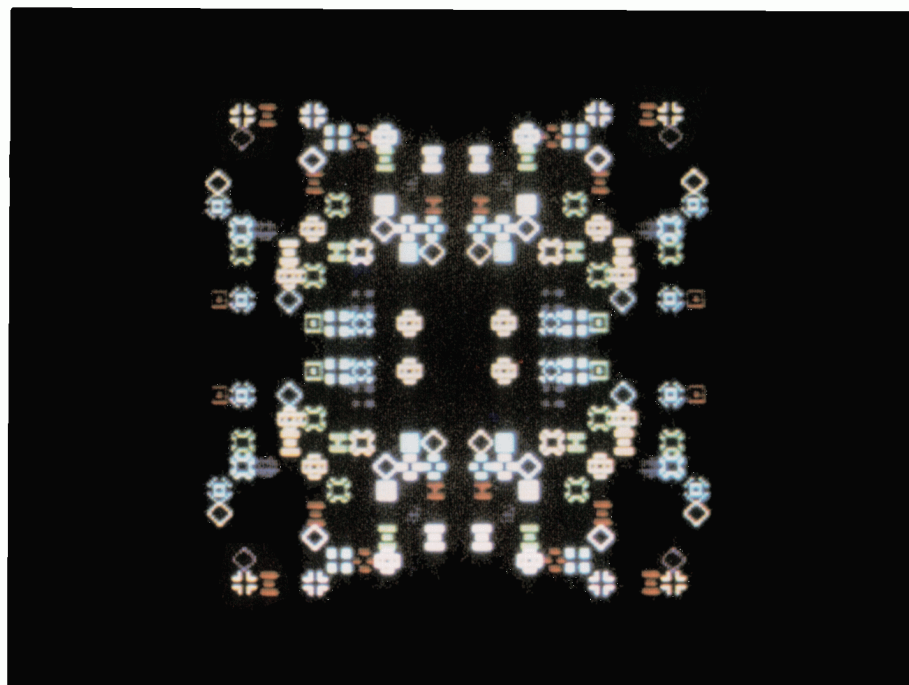


Figura 105

```

150 DATA 60,36,255,165,165,255,
36,60,102,189,195,66,66,195,189,
102
160 DATA 36,66,189,36,36,189,66
,36,0,126,126,126,126,126,126,0
170 LET U=PEEK 23675+256*PEEK 2
3676
180 FOR a=0 TO U+95
190 READ d: POKE a,d: NEXT a
200 RANDOMIZE
210 LET x=144+INT (RND*30): IF
x>155 THEN LET x=128
220 LET l=1+INT (RND*10)
230 LET c=6+INT (RND*10)
240 INK 1+INT (RND*7)
250 PRINT AT l,c;CHR$ x;AT l,32
-c;CHR$ x;AT 22-l,32-c;CHR$ x;AT
22-l,c;CHR$ x
260 GO TO 210

```

**Figura 104**

## ROMBOSPIRALE

Altra animazione è 'Rombospirale' (Figura 106) che disegna e cancella alternativamente in continuazione, una spirale a forma di rombo formata da caratteri e colori casuali (Figura 107).

```
10 REM      ROMBOSPIRALE
20 REM      -----
100 BORDER 0: PAPER 0: CLS
110 LET C=1
120 GO SUB 370
130 FOR P=1 TO 11
140 PRINT AT 12-P,P+S;CHR$
150 NEXT P
160 FOR N=1 TO 11 STEP 2
170 GO SUB 370
180 FOR P=1 TO 12-N
190 PRINT AT -1+P+N,15+P,CHR$ X
200 NEXT P
210 GO SUB 370
220 FOR P=1 TO 12-N
230 PRINT AT 10+P,28-P-N;CHR$ X
240 NEXT P
250 GO SUB 370
```

Figura 106 (continua)

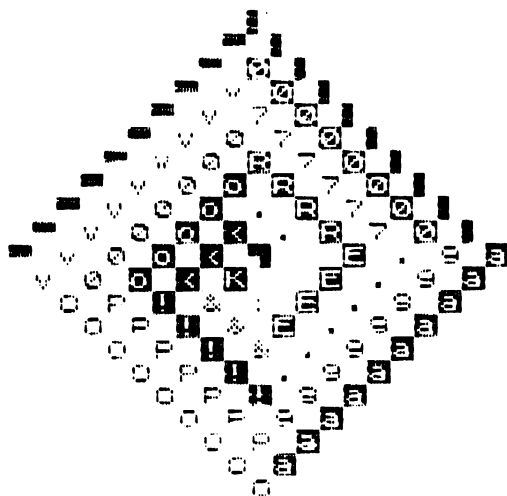


Figura 107

```

260 FOR p=1 TO 11-n
270 PRINT AT 23-p-n,17-p;CHR$ X
280 NEXT p
290 GO SUB 370
300 FOR p=1 TO 11-n
310 PRINT AT 13-p,5+p+n;CHR$ X
320 NEXT p
330 NEXT n
340 LET c=c*(-1)
350 LET x=32
360 GO TO 120
370 IF c=1 THEN LET x=33+INT (R
ND*11): INVERSE INT (RND*2): IN
K INT (RND*7)+1
380 RETURN

```

**Figura 106**

## PROGRAMMA UNIVERSALE ANIMAZIONE CON PRINT AT

Una complessa animazione può richiedere decine di linee con PRINT AT. Il programma di Figura 108 permette di condensare tutte le istruzioni necessarie all'animazione in sei stringhe.

Per ogni singolo movimento le sei stringhe contengono in codice i seguenti dati:

l\$	codice linea
c\$	codice colonna
i\$	codice INK
p\$	codice PAPER
x\$	carattere (anche grafico o speciale)
t\$	codice tempo tra un PRINT AT e il successivo

I codici ed i rispettivi valori per, linee, colonne, tempi e colori INK e PAPER, sono nella Tabella 2; per quanto riguarda i caratteri, nella stringa x\$ è presente direttamente il carattere e non un suo codice.

Per un corretto funzionamento è essenziale che le sei stringhe abbiano la stessa lunghezza.

Nel programma è anche prevista una zona (a partire dalla linea 1000), dove inserire eventuali routines (come nell'esempio) per caricare in memoria caratteri speciali o per disegnare all'inizio figure o altre cose necessarie all'animazione.

I dati contenuti nelle stringhe del programma di Figura 108, mostrano, come esempio di animazione, il movimento di un insetto in una tana con una entrata ed un'uscita, in Figura 109 è visibile l'inizio di tale animazione.

```

10 REM Programma Universale
20 REM Animazione con PRINT AT
30 REM con dati esempio
40 REM -----
99 BORDER 4: PAPER 0: CLS
100 GO SUB 1000
110 REM Codici Linee in l$
120 LET l$="5555555555555444333
333444455555444433333333"
130 REM Codici Colonne in c$
140 LET c$="778899::;<<<<==>
>????@#@AABBBBCCCCDDEEFFG"
150 REM Colori in i$
160 LET i$="7060500616360646062
60616065636061606460650607"
165 REM Fondi in p$

```

Figura 108 (continua)





# MUSICA ED EFFETTI SONORI

## MINIORGANO

Nonostante alcune limitazioni, dovute al fatto che lo Spectrum non possiede un integrato indipendente per la generazione dei suoni ma è lo Z80 che li produce, è possibile realizzare con il programma di figura 110 un Miniorgano a 5 ottave (61 note) che usa 18 dei 40 tasti dello Spectrum.

Nel programma è utilizzata una routine in linguaggio macchina lunga 89 Bytes che serve a ridurre al minimo il tempo di lettura della tastiera ed il calcolo del codice nota dell'istruzione BEEP. La funzione dei tasti usati dal Miniorgano è la seguente;

Tasto	Funzione
1	prima ottava
2	seconda ottava
3	terza ottava
4	quarta ottava
5	quinta ottava
Z	DO
S	DO#
X	RE
D	RE#
C	MI
V	FA
G	FA#
B	SOL
H	SOL#
N	LA
J	LA#
M	SI
'SYMBOL SHIFT'	DO (ottava superiore)

Per usare il Miniorgano occorre premere contemporaneamente uno dei cinque tasti di ottava ('1'...'5') ed uno dei tredici tasti di nota (per es. 'G').

Importante: Non inserire nel programma l'istruzione BORDER altrimenti non fun-

ziona; se si vuole un dato colore del bordo lo si può inserire con un comando diretto prima di scrivere il programma.

```

10 REM MUNDOREINO
20 REM
30 REM -----
40 INPUT "Inserire durata nota (0.1--1) ";d: IF d<.1 OR d>1 THEN GO TO 99
50 LET f=0
60 FOR n=1 TO 1000
70 LET r=INT (RND*20) -10
80 LET f=f+r
90 IF f>69 THEN LET f=f-20
100 IF f<-60 THEN LET f=f+20
110 BEEP d,f
120 NEXT n
130
140 LET u=PEEK 23675+256*PEEK 2
150 FOR a=u TO u+88
160 READ d: POKE a,d: NEXT a
170 POKE (u+26),INT (u/256)
180 POKE (u+27),u-(INT (u/256))
190 BEEP 0.3,USR (u+26)-36: GO TO 180

```

Figura 110

## GENERATORE DI MUSICA CASUALE – 130 NOTE

Il programma di Figura 111 produce musica utilizzando in modo casuale le cento-trenta note diverse dello Spectrum. All'inizio occorre inserire la durata richiesta per ogni nota.

```

10 REM GENERATORE DI MUSICA
20 REM CASUALE - 130 NOTE
30 REM -----
40 INPUT "Inserire durata nota (0.1--1) ";d: IF d<.1 OR d>1 THEN GO TO 99
50 LET f=0
60 FOR n=1 TO 1000
70 LET r=INT (RND*20) -10
80 LET f=f+r
90 IF f>69 THEN LET f=f-20
100 IF f<-60 THEN LET f=f+20
110 BEEP d,f
120 NEXT n
130
140 LET u=PEEK 23675+256*PEEK 2
150 FOR a=u TO u+88
160 READ d: POKE a,d: NEXT a
170 POKE (u+26),INT (u/256)
180 POKE (u+27),u-(INT (u/256))
190 BEEP 0.3,USR (u+26)-36: GO TO 180

```

Figura 111

## METRONOMO

Il Metronomo (Figura 112) può produrre da 60 a 300 battute al minuto a seconda del valore inserito all'inizio nella richiesta di Input fatta dal programma.

```
10 REM METRONOMO
20 REM -----
100 INPUT "Inserire N° battute
al minuto (60--300) e premere
ENTER "; b
110 IF b<60 OR b>300 THEN GO TO
100
120 LET t=3000/b-1
130 BEEP .02,-15
140 PAUSE t
150 GO TO 130
```

Figura 112

## SCALE MUSICALI

Il programma di Figura 113 produce, continuamente le scale musicali alternativa-mente a salire ed a scendere. All'inizio può essere stabilita la durata delle note delle scale.

```
10 REM SCALE MUSICALI
20 REM -----
100 INPUT "Inserire durata note
(0.1--2) "; d: IF d<.1 OR d>2 TH
EN GO TO 100
110 DIM a$(12,4)
120 LET a$(1)="DO "; LET a$(2)
="DO# "; LET a$(3)="RE "; LET a
$(4)="RE# "; LET a$(5)="MI "; L
ET a$(6)="FA "; LET a$(7)="FA#
"; LET a$(8)="SOL "; LET a$(9)="
SOL#"; LET a$(10)="LA "; LET a$
(11)="LA# "; LET a$(12)="SI "
130 FOR o=-5 TO 4
140 FOR n=0 TO 11
150 INK 6: PRINT AT 10,10;a$(n+
1); " Ottava ";o+5; " "
160 BEEP d,o*12+n
170 NEXT n: NEXT o
180 FOR o=4 TO -5 STEP -1
190 FOR n=11 TO 0 STEP -1
200 INK 5: PRINT AT 10,10;a$(n+
1); " Ottava ";o+5; " "
210 BEEP d,o*12+n
220 NEXT n: NEXT o
230 GO TO 130
```

Figura 113

## PROGRAMMA UNIVERSALE PER MUSICA

Suonare con il Computer il vostro pezzo musicale preferito può voler dire realizzare un programma con, nel caso dello Spectrum, centinaia di linee con istruzioni BEEP.

Tali istruzioni possono essere compattate usando il programma di Figura 114 nel quale le note sono inserite in codice in una stringa (a\$). Ad ogni nota tra la '-36' e la '+56' corrisponde un carattere ASCII da inserire nella stringa a\$ secondo la Tabella 3.

In a\$ le Pause sono inserite con '!'.

La durata delle note è stabilita all'inizio del programma dal valore (in secondi) inserito nella variabile 'd' (linea 99); se nel pezzo musicale occorrono a volte note di durata maggiore basta inserire nella stringa più codici della stessa nota di seguito.

```

10 REM          PROGRAMMA UNIVERSALE
20 REM          PER MUSICA
30 REM          -----
99 LET d=0.5: REM durata note
in secondi
100 REM          NOTE NELLA STRINGA a$
110 LET a$="#$%&'()*+,-./012345
6789:;<=>!!!!?@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~"
120 FOR n=1 TO LEN a$
130 IF CODE a$(n TO ) < 33 OR CODE
E a$(n TO ) > 127 OR CODE a$(n TO
)=34 THEN PRINT "NOTA ";n;" ERRA
TA"
140 NEXT n
150 FOR n=1 TO LEN a$
160 IF CODE a$(n TO )=33 THEN G
O TO 200
170 BEEP d, CODE a$(n TO )-71
180 NEXT n
190 STOP
200 PAUSE 50*d
210 NEXT n

```

Figura 114

CHARIT. ASCII	NOTA	DURATA	CODICE BEEP
!	PAUSA		
#	DO	2	-36
\$	DO#	2	-35

Tabella 3 (continua)





## OROLOGIO (Figura 116)

```
1000 REM OROLOGIO
1010 REM -----
1020 FOR n=1 TO 60
1030 BEEP .01,-5
1040 PAUSE 24
1050 BEEP .01,-12
1060 PAUSE 24
1070 NEXT n
```

Figura 116

## MITRA (Figura 117)

```
1000 REM MITRA
1010 REM -----
1020 FOR n=1 TO 10
1030 FOR c=1 TO 3+INT (RND*10)
1040 BEEP .02,-24
1050 FOR t=1 TO 5: NEXT t
1060 NEXT c
1070 PAUSE 20+INT (RND*30)
1080 NEXT n
```

Figura 117

## SIRENA AMERICANA (Figura 118)

```
1000 REM SIRENA AMERICANA
1010 REM -----
1020 FOR a=1 TO 100
1030 BEEP .01,12
1040 BEEP .01,13
1050 BEEP .01,14
1060 BEEP .01,15
1070 BEEP .01,16
1080 BEEP .01,17
1090 BEEP .01,18
1100 BEEP .01,19
1110 BEEP .01,20
1120 BEEP .01,19
1130 BEEP .01,18
1140 BEEP .01,17
1150 BEEP .01,16
1160 BEEP .01,15
1170 BEEP .01,14
1180 BEEP .01,13
1190 NEXT a
```

Figura 118

### SIRENA A 2 TONI (Figura 119)

```
1000 REM      SIRENA A 2 TONI
1010 REM      -----
1020 FOR a=1 TO 30
1030 BEEP .5,2: REM tono 1
1040 BEEP .5,7: REM tono 2
1050 NEXT a
```

Figura 119

### SIRENA MULTITONO (Figura 120)

```
1000 REM      SIRENA MULTITONO
1010 REM      -----
1020 FOR a=1 TO 30
1030 BEEP .1,-7: REM tono 1
1040 BEEP .1,-4: REM tono 2
1050 BEEP .1,8: REM tono 3
1060 BEEP .1,12: REM tono 4
1070 BEEP .1,0: REM tono 3
1080 BEEP .1,-4: REM tono 2
1090 NEXT a
1100 REM (anche piu' di 4 toni)
```

Figura 120

### SQUILLO DEL TELEFONO (Figura 121)

```
1000 REM      SQUILLO DEL TELEFONO
1010 REM      -----
1020 FOR n=1 TO 8
1030 FOR f=1 TO 40
1040 BEEP .02,25
1050 FOR t=1 TO 1: NEXT t
1060 NEXT f
1070 PAUSE 80
1080 NEXT n
```

Figura 121

## SEGNALE TELEFONICO DI LINEA LIBERA (Figura 122)

```
1000 REM      SEGNALE TELEFONICO
1010 REM      DI LINEA LIBERA
1020 REM      -----
1030 FOR n=1 TO 10
1040 BEEP 1.4,16
1050 PAUSE 70
1060 NEXT n
```

Figura 122

## SEGNALE TELEFONICO DI LINEA OCCUPATA (Figura 123)

```
1000 REM      SEGNALE TELEFONICO
1010 REM      DI LINEA OCCUPATA
1020 REM      -----
1030 FOR n=1 TO 30
1040 BEEP .2,14
1050 PAUSE 10
1060 NEXT n
```

Figura 123

## GENERATORE DI SUONI CASUALI (Figura 124)

```
1000 REM      GENERATORE DI SUONI
1010 REM      CASUALI
1020 REM      -----
1030 FOR n=1 TO 1000
1040 LET t=10: REM o t=5 o t=20
1050 BEEP RND/t,RND*60-20
1060 NEXT n
```

Figura 124

## DIN-DON (Figura 125)

```
1000 REM      DIN-DON
1010 REM      -----
1020 FOR n=1 TO 10
1030 BEEP .02,23
1040 PAUSE 5
1050 BEEP .3,19
1060 PAUSE 50
1070 NEXT n
```

Figura 125

**GRILLI (Figura 126).**

```
1000 REM GRILLI
1010 REM -----
1020 FOR n=1 TO 50
1030 FOR k=1 TO 3
1040 BEEP .02,36
1050 FOR t=1 TO 1: NEXT t
1060 NEXT k
1070 PAUSE 6
1080 NEXT n
```

**Figura 126**

# GIOCHI

## CACCIA AL SOMMERGIBILE

Il gioco (Figura 127) consiste nel trovare un sommergibile muovendo tramite i tasti 5, 6, 7, 8 un simbolo che rappresenta la nave o l'aereo antisommergibile.

Naturalmente finchè la nave non gli passa sopra il sommergibile viaggia nascosto sott'acqua spostandosi casualmente in tutte le direzioni, il cacciatore ha però la possibilità di scovarlo seguendo il suono di una sorta di 'SONAR' prodotto dal computer.

Infatti tale suono AUMENTA di frequenza se ci si sta avvicinando e diminuisce se ci si sta allontanando o se il sommergibile si sposta in senso opposto.

Appena la nave o aereo trova il sommergibile, questo appare (Figura 128) insieme ad una scritta che indica in quante mosse ci si è riusciti.

Ovviamente ogni volta che si inizia il gioco la nave parte dallo stesso angolo ed il sommergibile viene nascosto nello stesso punto.

Il gioco può essere reso più difficile facendo spostare più spesso il sommergibile sostituendo la linea 310 con:

```
310 GO SUB 350
```

Un'ulteriore difficoltà può essere introdotta facendo spostare il sommergibile di spazi più ampi, sostituendo le linee 350 e 360 con:

```
350 LET k=INT (RND*5) -2
```

```
360 LET h=INT (RND*5) -2
```

```
10 REM CHOCCH HL SOMMERGIBILE
20 REM -----
99 BORDER 0: PAPER 1: INK 7: C
LS
100 DATA 1,130,76,48,4,28,50,19
3,0,48,16,48,56,252,255,0
110 LET U=PEEK 23675+256*PEEK 2
3676
111 FOR w=U TO U+15
112 READ g: POKE w,g: NEXT w
120 FOR s=1 TO 176
130 PRINT CHR$ 144;CHR$ 144;CHR$ 144;CHR
$ 144;CHR$ 144;
140 NEXT s
150 LET l=21: LET c=0
160 LET x=20: LET y=8
```

Figura 127 (continua)

```

170 LET M=0: LET f=-60
180 INK 4: PRINT AT L,C;"#"
185 BEEP 0.3,f
190 INK 7: PRINT AT L,C;CHR$ 14
4
200 IF C>0 AND INKEY$="5" THEN
LET C=C-1
210 IF L<21 AND INKEY$="6" THEN
LET L=L+1
220 IF L>0 AND INKEY$="7" THEN
LET L=L-1
230 IF C<31 AND INKEY$="8" THEN
LET C=C+1
240 IF CODE INKEY$>52 AND CODE
INKEY$<57 THEN GO TO 300
250 IF L=4 AND C=X THEN GO TO 2
70
260 GO TO 180
270 INK 6: PRINT AT 9,X;CHR$ 14
5
277 BEEP 2,-30

```

Figura 127 (continua)

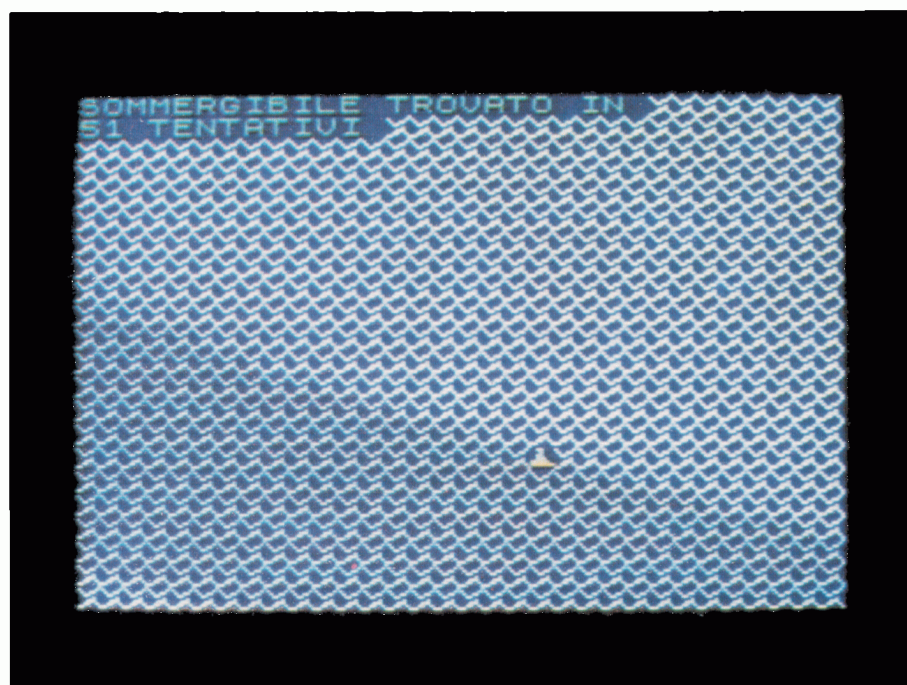


Figura 128

```

280 INK 4: PRINT AT 0,0;"SOMMER
GIBILE TROVATO IN ";AT 1,0;m);" T
ENTATIVI "
290 PAUSE 50: STOP
300 LET m=m+1
310 IF RAND<0.5 THEN GO SUB 350
320 LET J=53-ABS (y-l)-ABS (c-x
)
330 LET f=-55+J*2
340 GO TO 250
350 LET k=INT (RAND*3)-1
360 LET h=INT (RAND*3)-1
370 IF x+k<=31 AND x+k>=0 THEN
LET x=x+k
380 IF y+h<=21 AND y+h>=0 THEN
LET y=y+h
390 RETURN

```

**Figura 127**

## **TIRO ALLA FUNE**

Altro gioco è il Tiro alla Fune (Figura 129) con il quale devono giocare due persone.

All'inizio appaiono sullo schermo due omini (Figura 130) con una fune.

L'omino di sinistra è Giallo e viene comandato dal tasto 'A' mentre l'omino di destra è Bianco e viene comandato dal tasto 'L'. Ogni volta che appare il rettangolo rosso al centro dello schermo il giocatore che preme per primo il proprio tasto ha la possibilità di tirare i due uomini dalla propria parte, se però uno dei giocatori preme il proprio tasto prima che appaia il rettangolo, i due omini si sposteranno nel senso opposto favorendo l'avversario.

Il gioco continua finchè uno dei due giocatori non riesce a trascinare l'avversario completamente nella propria zona (Figura 131).

È importante durante il gioco che i giocatori non premano tasti diversi da 'A' e 'L'.

Il tempo di apparizione del rettangolo rosso ed il ritardo tra una apparizione e l'altra sono casuali e vengono stabiliti rispettivamente dalle linee 290 e 280 che possono essere eventualmente modificate per rendere il gioco più difficile.

Con le linee attuali il tempo di apparizione ed il ritardo sono rispettivamente di 200-600 msec. e di 1-3 sec.

Importante: Non inserire in questo programma l'istruzione BORDER altrimenti non funziona più. Se si vuole un colore per il bordo lo si può stabilire con un comando diretto prima di inserire il programma.



```

280 POKE 20480,INT (RND*100)+50
290 POKE 20481,INT (RND*20)+10
300 GO SUB USR (U+40)*500
310 POKE 22831,0: POKE 22832,0
320 GO TO 260
500 PRINT AT 2,1: INK 6;"Vince
il GIALLO"
510 PRINT AT 12,2: INK 5;"Preme
re ENTER per continuare"
520 INPUT e$
530 RUN
6400 RETURN
6500 PRINT AT 2,16: INK 7;"Vince
il BIANCO"
6600 PRINT AT 12,2: INK 5;"Preme
re ENTER per continuare"
6700 INPUT e$
6800 RUN

```

Figura 129



Figura 130

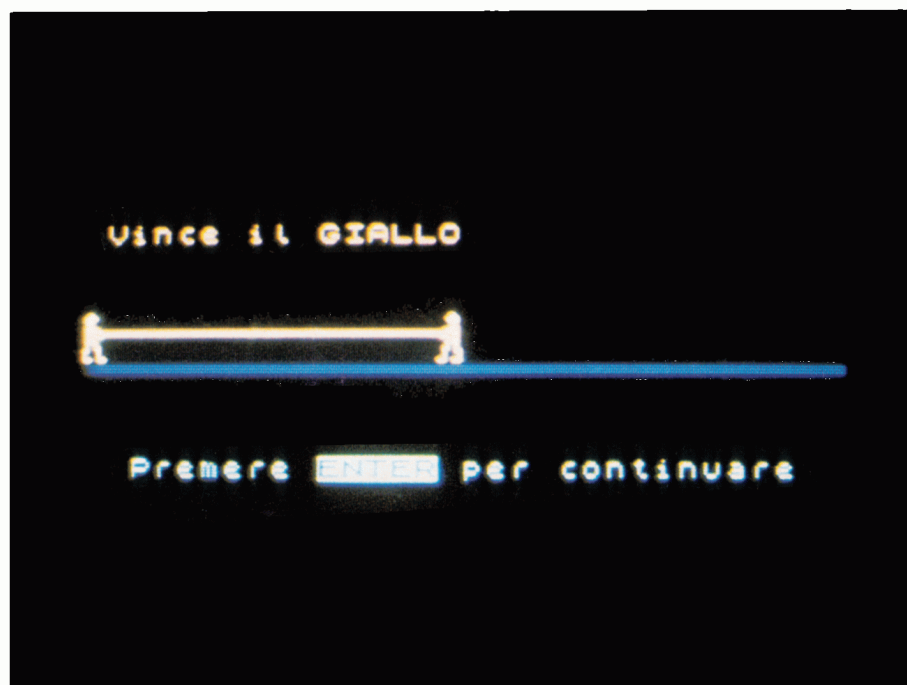


Figura 131

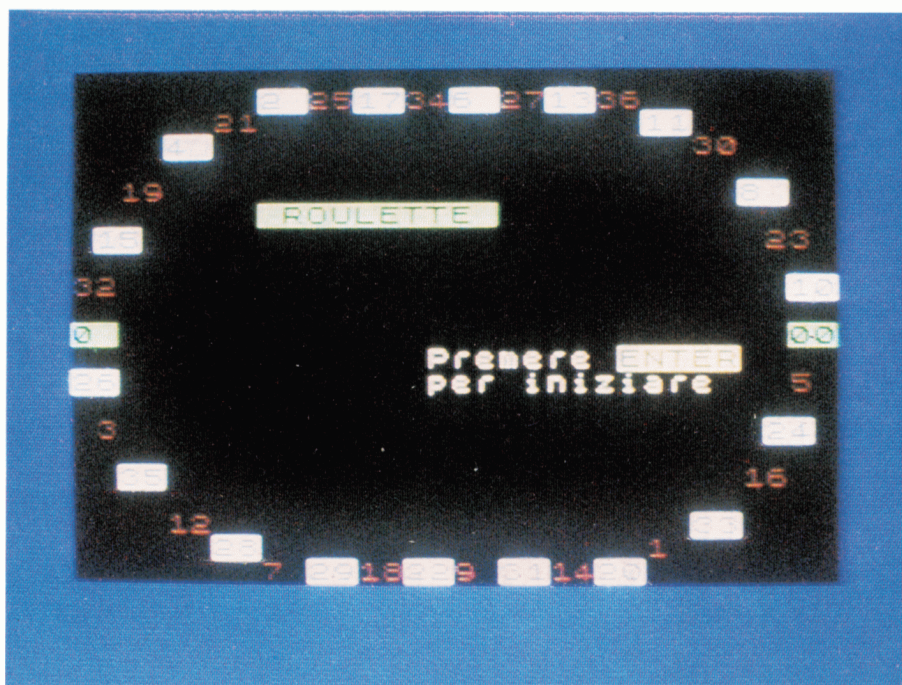


```

350 LET K$=SCREEN$ (L,C): LET J
$=SCREEN$ (L,C+1): LET K$=K$+J$
360 LET P=INT ATTR (L,C)/8
370 PRINT AT L,C: PAPER 0: INK
6:K$
377 PAUSE 3
380 PRINT AT L,C: PAPER P: INK
i:k$
390 NEXT X
400 PRINT AT L,C: FLASH 1: PAPE
R P: INK i:k$
410 INK 6: PRINT AT 12,15;"Prem
ere ENTER"; AT 13,15;"per continu
are"
420 INPUT e$
430 FLASH 0: GO TO 100

```

**Figura 132**



**Figura 133**

## CAMPO MINATO

Questo programma (Figura 134) simula un campo in cui vi sono 20 mine nascoste casualmente. Il giocatore deve spostare, usando i tasti 5, 6, 7, 8, l'omino dalla posizione di partenza nell'angolo in basso a sinistra alla posizione di arrivo nell'angolo in alto a destra senza incappare in una delle mine nascoste. Per attraversare il campo minato senza danni, il giocatore può contare sul proprio intuito e sulla propria fortuna, se riesce appare sul video la scritta 'SALVO !!!'. Aggiungendo la linea 245 (Figura 135) e facendo girare il programma, si può avere un'idea di come vengono disposte le mine sul campo (Figura 136). Naturalmente durante il gioco le mine devono essere nascoste per cui si deve prima togliere dal programma la linea 245.

Ora con le mine nascoste può capitare di terminare il gioco senza danno (Figura 137) oppure può succedere prima dell'arrivo di passare su una delle mine (Figura 138).

Il numero delle mine può essere diminuito o aumentato per rendere più facile o più difficile il gioco. Se per esempio si vogliono nascondere 12 mine invece di 20 basta modificare il contenuto delle linee 190, 210 e 400, sostituendo, all'interno delle linee stesse, tutti i 20 con dei 12.

```

10 REM CAMPO MINATO
20 REM -----
30 LET U=256*PEEK 23676+PEEK 2
3675
40 FOR a=U TO U+31: READ w: PO
KE a,w: NEXT a
50 DATA 170,85,170,85,170,85,1
70,85,85,170,85,170,85,170,85,17
0,85,85,16,254,186,40,40,238,0,1
6,68,16,146,56,124,254
100 BORDER 0: PAPER 0: CLS
110 FOR a=1 TO 88
120 INK 4: PRINT CHR$(144);CHR$
145;CHR$(144);CHR$(145);CHR$(144);C
HR$(145);CHR$(144);CHR$(145);
130 NEXT a
140 PRINT
150 LET l=21
160 LET c=0
170 INK 6: PRINT AT 0,31;"A"
180 INK 2: PRINT AT l,c;"B"
188 PAUSE 30
190 DIM m(20,2)
200 RANDOMIZE

```

Figura 134 (continua)

```

210 FOR a=1 TO 20
220 LET g=INT (32*RND)
230 LET f=INT (22*RND)
240 IF g<6 AND f>15 THEN GO TO
250 LET m(a,1)=f
260 LET m(a,2)=g
270 NEXT a
280 INK 6: PRINT AT l,c;CHR$ 14
6
290 BEEP .1,30
300 INK 0: PRINT AT l,c;"■"
320 LET x=l
330 LET y=c
340 IF c>0 AND INKEY$="5" THEN
LET c=c-1
350 IF l<21 AND INKEY$="6" THEN
LET l=l+1
360 IF l>0 AND INKEY$="7" THEN
LET l=l-1
370 IF c<31 AND INKEY$="8" THEN
LET c=c+1
380 IF x<>l OR y<>c THEN GO TO
400
390 GO TO 280
400 FOR a=1 TO 20
410 IF m(a,1)=l AND m(a,2)=c TH
EN GO TO 450
420 NEXT a
430 IF l=0 AND c=31 THEN GO TO
530
440 GO TO 280
510 INK 7: PRINT AT l,c;CHR$ 14
7
520 BEEP 2,-40: STOP
530 PRINT AT 0,31;CHR$ 146
540 INK 6: PRINT AT 8,0;"
SALVO !!!"

```

Figura 134

```

245 INK 5: PRINT AT f,g;CHR$ 14
7

```

Figura 135

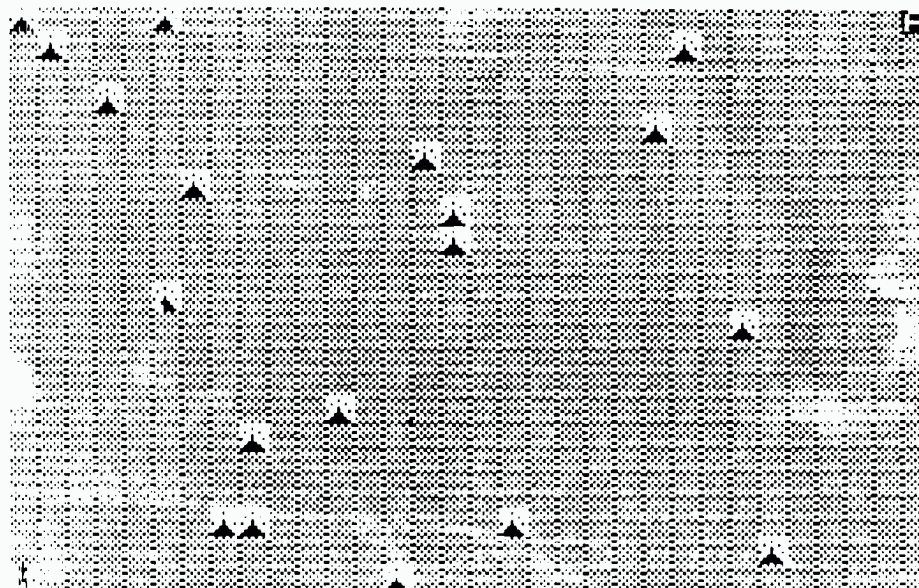


Figura 136

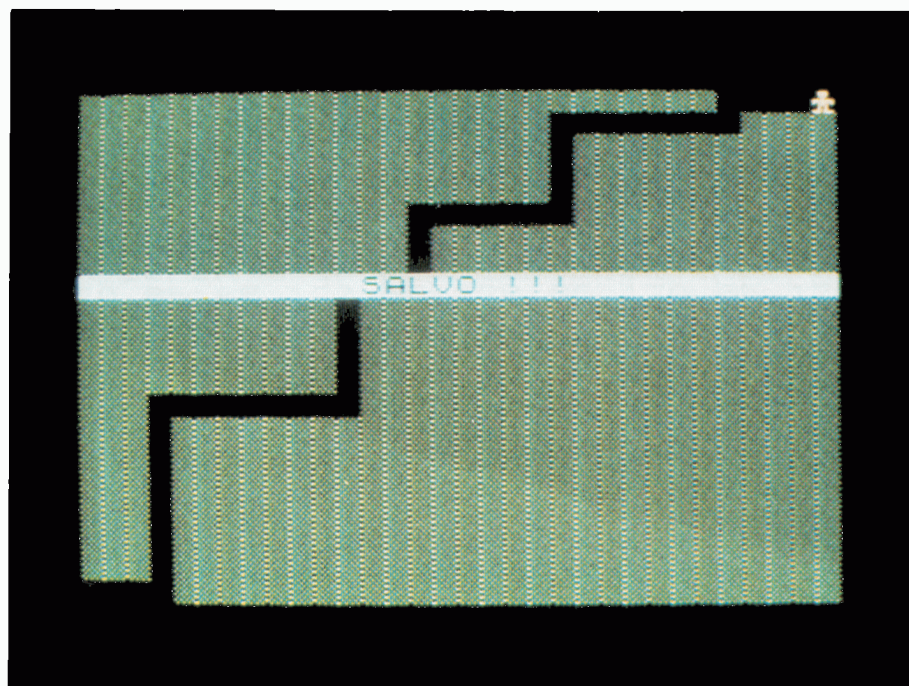
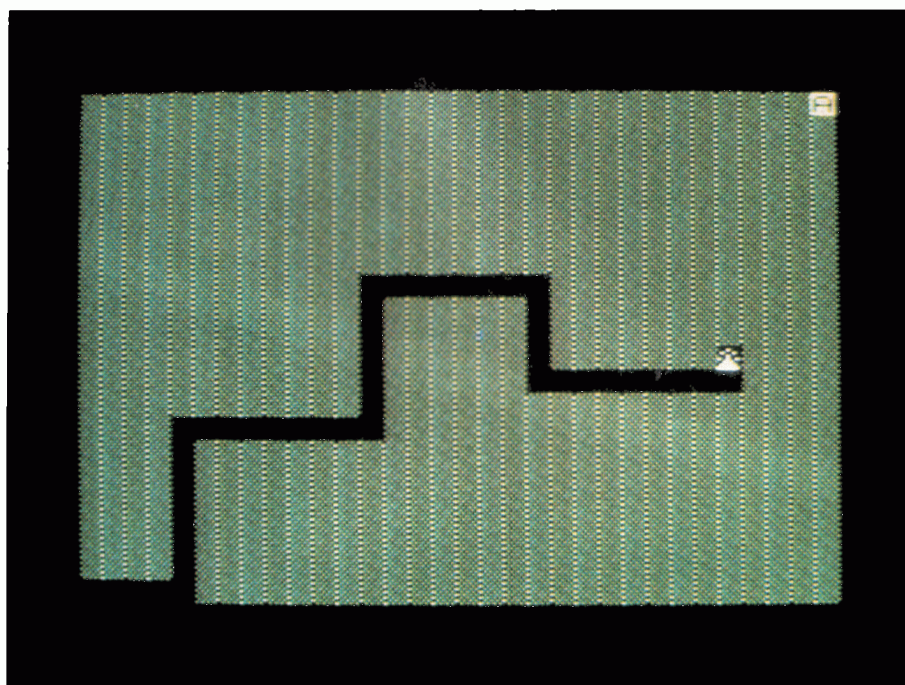


Figura 137



**Figura 138**

## FLIPPER GIAPPONESE

In questo 'Flipper Giapponese' (Figura 139), trenta palline cadono dall'alto attraverso una serie di chiodini, finendo in modo casuale su delle basi che indicano dei punteggi diversi, punteggi che vengono automaticamente sommati al totale (Figura 140).

Il numero delle palline può essere modificato cambiando la cifra '30' nella linea 290.

```

10 REM FLIPPER GIAPPONESE
20 REM -----
100 BORDER 4: PAPER 1: CLS
110 DATA 24,24,0,0,0,0,0,0,0,0,
24,60,126,126,60,24
120 LET U=PEEK 23675+256*PEEK 2
3676
130 FOR W=U TO U+15
140 READ Z: POKE W,Z: NEXT W
150 INK 2: PRINT AT 2,10;"
160 LET A$=CHR$ 144+CHR$ 128
170 FOR A=1 TO 4: LET A$=A$+A$:
NEXT A
180 LET A$=" "+A$
190 FOR A=0 TO 12
200 INK 7: PRINT AT 4+A,13-A;A$
(1 TO (A+3)*2+1)
210 NEXT A
220 INK 6: INVERSE 1: PRINT AT
17,3;"921
0:
230 PRINT AT 18,3;"95021
12059"
240 PRINT AT 19,3;"900526
55625009"
244 INVERSE 0
250 INK 7: PRINT AT 21,0;"Preme
re ENTER"
260 INPUT LINE E$
266 PRINT AT 21,16;"PUNTEGGIO
270 PRINT AT 21,0;"
280 LET S=0
290 FOR A=1 TO 30
300 LET L=3: LET C=16
310 INK 6: PRINT AT 3,3;A;"
320 PRINT AT 2,16;CHR$ 145: PAU
SE 20: PRINT AT 2,16;" "/AT 3,16
;CHR$ 145;AT 3,16;"
330 FOR C=1 TO 13
340 LET H=RND

```

Figura 139 (continua)

```

344 IF h<0.5 THEN LET h=-1: IF
h>=0.5 THEN LET h=1
346 IF h>=0.5 THEN LET h=1
350 PRINT AT (L,C) " "
355 LET L=L+1
360 PRINT AT (L,C+h)CHR$ 145
370 LET C=C+h
380 BEEP 0.05,-20
390 NEXT r
400 IF C=15 OR C=17 THEN LET S=
S+5
410 IF C=13 OR C=19 THEN LET S=
S+6
420 IF C=11 OR C=21 THEN LET S=
S+12
430 IF C=9 OR C=23 THEN LET S=S
+25
440 IF C=7 OR C=25 THEN LET S=S
+50
450 IF C=5 OR C=27 THEN LET S=S
+250
460 IF C=3 OR C=29 THEN LET S=S
+500
470 INK 7: PRINT AT (21,26):S
480 PAUSE 10: PRINT AT (L,C) " "
490 NEXT a
500 GO TO 250

```

Figura 139

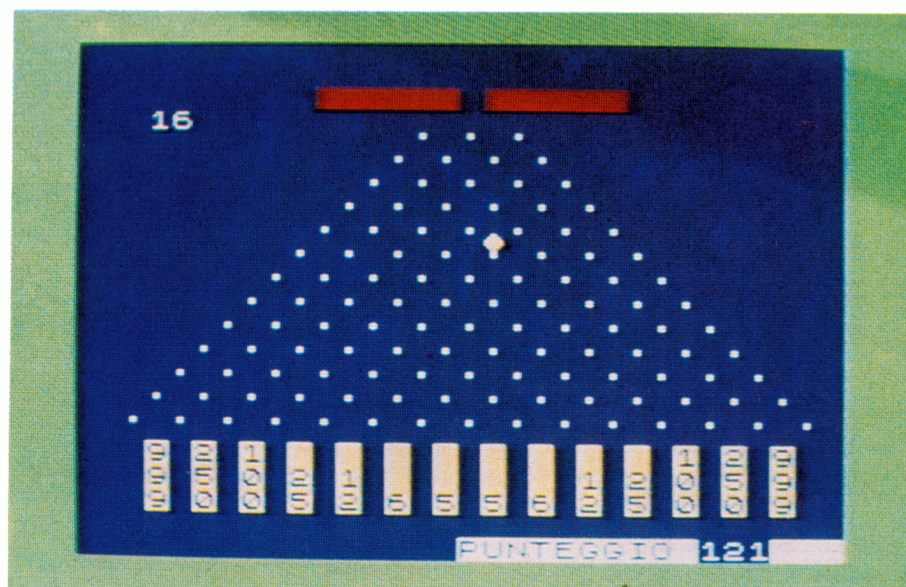


Figura 140

## TAVOLA DI NUMERI E LETTERE

La prima parte del programma di Figura 141 disegna sul video una tavola di sedici caselle di cui quindici occupate dai numeri da 0 a 9 più le lettere da A a E. Subito dopo, le quindici caselle vengono mescolate per apparire alla fine sullo schermo nel modo indicato in Figura 142.

A questo punto, usando i tasti 5, 6, 7, 8 si possono spostare le varie caselle con i numeri e le lettere per cercare di ritornare alla disposizione iniziale (Figura 143) nel minor numero possibile di mosse. Durante il gioco o alla fine, ogni volta che si preme il tasto 'p', si può avere il numero aggiornato delle mosse fatte.

```

10 REM          TAVOLA DI NUMERI
20 REM          E LETTERE
30 REM          -----
99 BORDER 1: PAPER 0: INK 2: C
LS
110 DIM a$(4,4)
120 LET a$(1)="1234"
130 LET a$(2)="5678"
140 LET a$(3)="90AB"
150 LET a$(4)="CDE"
160 PRINT AT 3,12;""
170 PRINT AT 4,12;""
180 PRINT AT 5,12;""
190 PRINT AT 6,12;""
200 PRINT AT 7,12;""
210 PRINT AT 8,12;""
220 LET p=0
230 LET l=4
240 LET c=4
250 INK 0
260 FOR m=1 TO 500
270 LET x=1+INT (4*RND)
280 IF c<4 AND x=1 THEN GO SUB
300
290 IF l>1 AND x=2 THEN GO SUB
440
300 IF l<4 AND x=3 THEN GO SUB
500
310 IF c>1 AND x=4 THEN GO SUB
560
310 NEXT m
315 LET p=p+1
320 IF INKEY$="p" THEN PRINT AT
10,8; INK 4;"TOTALE ";p;" MOSSE
": INK 0
330 IF c<4 AND INKEY$="5" THEN
GO SUB 380
340 IF l>1 AND INKEY$="6" THEN
GO SUB 440
350 IF l<4 AND INKEY$="7" THEN
GO SUB 500

```

Figura 141 (continua)

```

360 IF c>1 AND INKEY$="8" THEN
GOTO 370
370 GO TO 320
380 PRINT AT l+3,c+13;" "
390 LET a$(l,c)=a$(l,c+1)
400 LET a$(l,c+1)=" "
410 PRINT AT l+3,c+12;a$(l,c)
420 LET c=c+1
430 LET p=p+1
440 RETURN
450 PRINT AT l+2,c+12;" "
460 LET a$(l,c)=a$(l-1,c)
470 LET a$(l-1,c)=" "
480 PRINT AT l+3,c+12;a$(l,c)
490 LET l=l-1
500 LET p=p+1
510 RETURN
520 PRINT AT l+4,c+12;" "
530 LET a$(l,c)=a$(l+1,c)
540 LET a$(l+1,c)=" "
550 PRINT AT l+3,c+12;a$(l,c)
560 LET l=l+1
570 LET p=p+1
580 RETURN
590 PRINT AT l+3,c+11;" "
600 LET a$(l,c)=a$(l,c-1)
610 LET a$(l,c-1)=" "
620 PRINT AT l+3,c+12;a$(l,c)
630 LET c=c-1
640 LET p=p+1
650 RETURN

```

Figura 141

```

5190
004E
0070
030

```

Figura 142

```

1234
5678
90AB
CDE

```

TOTALE 508 MOSSE

Figura 143

## 1 - 40 DADI

L'ultimo programma di questo libro e dei giochi (Figura 144) disegna sullo schermo fino a quaranta dadi (Figura 145) insieme al punteggio totale. Il numero di dadi desiderato può essere inserito all'inizio, quando il programma lo richiede con un Input.

```

10 REM      1-40 DADI
20 REM      -----
100 INPUT "Inserire N° dadi ( 1-
-40 ) e premere ENTER ";n
120 IF n<1 OR n>40 THEN GO TO 1
00
150 BORDER 0: PAPER 0: CLS
160 LET k=0
170 LET y=0
180 RANDOMIZE
190 FOR l=0 TO 19 STEP 4
200 FOR c=0 TO 31 STEP 4
210 LET y=y+1
220 LET x=1+INT (RND*6)
222 INK 2+INT (RND*6)
230 PRINT AT l,c;CHR$ 132;CHR$
140;CHR$ 140;CHR$ 140
240 IF x=1 THEN PRINT AT l+1,c;
CHR$ 133;CHR$ 143;CHR$ 143;CHR$
143
250 IF x=2 OR x=3 THEN PRINT AT
l+1,c;CHR$ 133;CHR$ 141;CHR$ 14
3;CHR$ 143
260 IF x=4 OR x=5 THEN PRINT AT
l+1,c;CHR$ 133;CHR$ 141;CHR$ 14
3;CHR$ 141
270 IF x=6 THEN PRINT AT l+1,c;
CHR$ 133;CHR$ 141;CHR$ 141;CHR$
141
280 IF x=1 OR x=3 OR x=5 THEN P
RINT AT l+2,c;CHR$ 133;CHR$ 143;
CHR$ 141;CHR$ 143
290 IF x=2 OR x=4 OR x=6 THEN P
RINT AT l+2,c;CHR$ 133;CHR$ 143;
CHR$ 143;CHR$ 143
300 IF x=1 THEN PRINT AT l+3,c;
CHR$ 133;CHR$ 143;CHR$ 143;CHR$
143
310 IF x=2 OR x=3 THEN PRINT AT
l+3,c;CHR$ 133;CHR$ 143;CHR$ 14
3;CHR$ 141
320 IF x=4 OR x=5 THEN PRINT AT
l+3,c;CHR$ 133;CHR$ 141;CHR$ 14
3;CHR$ 141

```

Figura 144 (continua)

```

330 IF X=6 THEN PRINT AT (L+3,C)
CHR$ 133;CHR$ 141;CHR$ 141;CHR$
141
340 LET K=K+X
350 IF Y=0 THEN GO TO 380
360 NEXT C
370 NEXT L
380 PRINT
390 PRINT "          "); INK 6; PRINT
N;" dadi";"          punteggio = ";K
400 REM

```

Figura 144

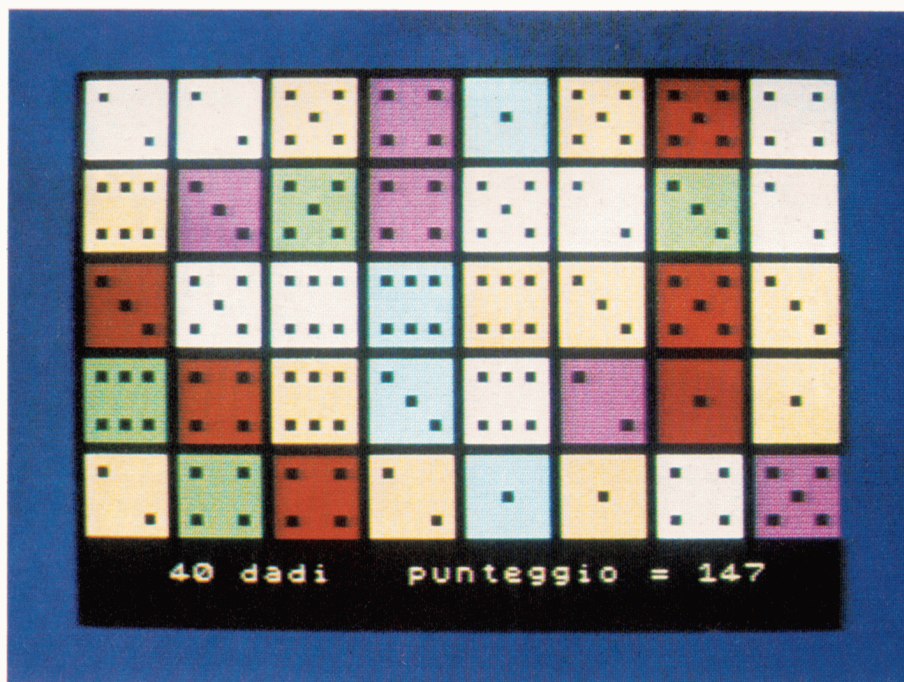


Figura 145









La particolarità dei 77 programmi per lo Spectrum contenuti in questo libro è' oltre alla qualità, soprattutto la varietà delle applicazioni e degli argomenti per la realizzazione dei quali si è cercato di utilizzare al massimo le nuove interessanti prestazioni dello Spectrum quali le nuove istruzioni BASIC, la velocità, il Suono, e soprattutto il Colore e la Grafica tra cui specialmente l'altra risoluzione e la possibilità di creare dei Caratteri Speciali. Le applicazioni vanno dalla Grafica (30) a, Musica (5), Utility (8), Animazioni (6), Giochi (7), Effetti Sonori (12), Elettronica (4) ed altri programmi (5).

Un cenno particolare merita senz'altro la 'Business Grafica', presente con 9 programmi, e realizzata con una efficacia visiva e una utilizzabilità pratica precedentemente possibile solo su personal e business computers molto più costosi.

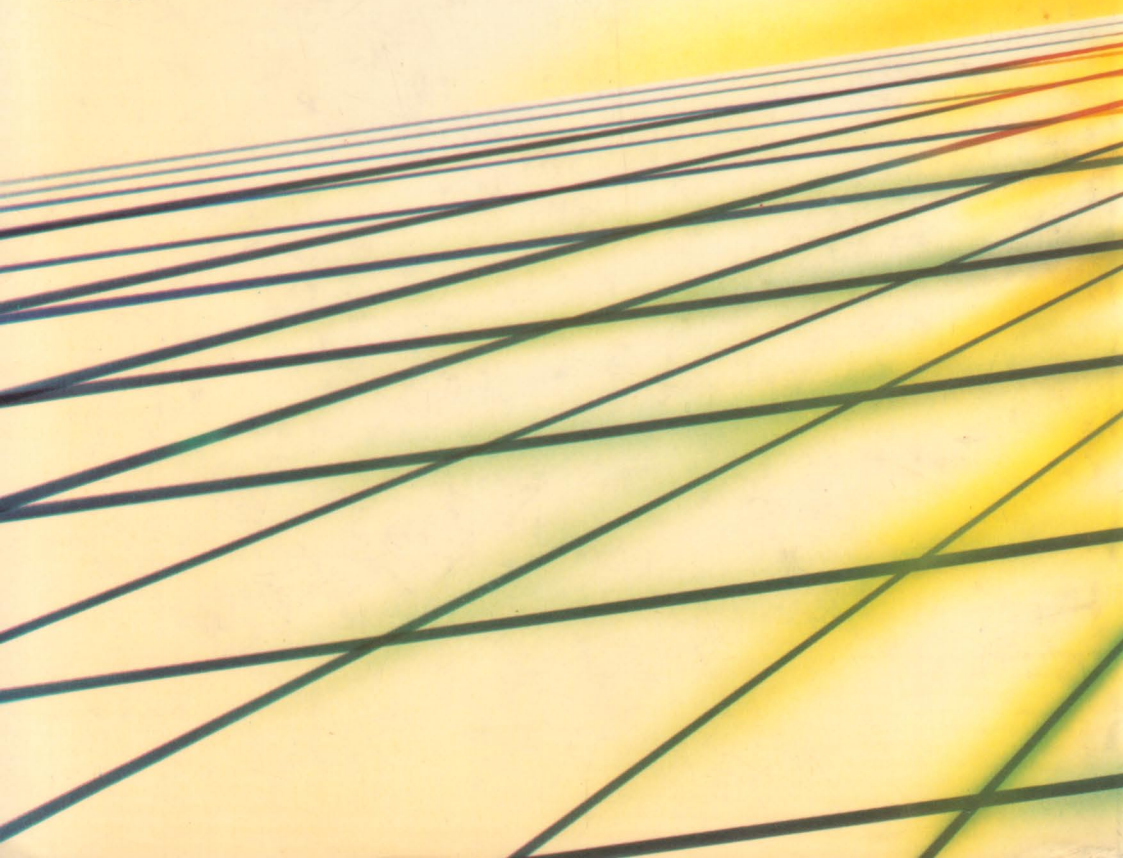
Dello stesso Autore:

- 66 PROGRAMMI PER ZX81 E ZX80 Con Nuova ROM + Hardware.
- GLI AMPLIFICATORI DI NORTON QUADRUPLI LM3900 & LM359 con Esperimenti.

**L. 16.000**

Cod. 555A

ISBN 88-7056-143-7





**77**  
**PRONG**  
**PER**  
**SPER**  
**CE**

**di Gaetano Marano**

**GRUPPO  
EDITORIALE  
JACKSON**

