

CADEL ✓

Spediz. in abbonamento postale GR. 11/70 L. 2.000
(...)

15 CORSO PRATICO COL COMPUTER

42167

F4 F5 F6 F7 F8

diretto da **GIANNI DEGLI ANTONI**

è una iniziativa
FABBRI EDITORI

in collaborazione con
BANCO DI ROMA

e **OLIVETTI**



BATTERY LOW

**IN OMAGGIO
IL SESTO POSTER
"LA STORIA
DELL'INFORMATICA"**



**FABBRI
EDITORI**

LIBRERIA DI SOFTWARE

Spediz. in abbonamento postale GR. 11/70 L. 8.000 (...)

Personal Computer Olivetti M 10
Commodore 64 • ZX Sinclair Spectrum

6

a cura di EIDOS

Programmi di statistica - 1

PARAM: valutazione dei parametri statistici

FABBRI EDITORI

**A partire dal 10 luglio sarà in edicola
il sesto numero di LIBRERIA DI
SOFTWARE dedicato a Programmi
di statistica - 1 PARAM:
valutazione dei parametri statistici**

Direttore dell'opera
GIANNI DEGLI ANTONI

Comitato Scientifico
GIANNI DEGLI ANTONI
Docente di Teoria dell'informazione, Direttore dell'Istituto di Cibernetica
dell'Università degli Studi di Milano

UMBERTO ECO
Ordinario di Semiotica presso l'Università di Bologna

MARIO ITALIANI
Ordinario di Teoria e Applicazione delle Macchine Calcolatrici presso
l'Istituto di Cibernetica dell'Università degli Studi di Milano

MARCO MAIOCCHI
Professore Incaricato di Teoria e Applicazione delle Macchine Calcolatrici
presso l'Istituto di Cibernetica dell'Università degli Studi di Milano

DANIELE MARINI
Ricercatore universitario presso l'Istituto di Cibernetica dell'Università
degli Studi di Milano

Curatori di rubriche
TULLIO CHERSI, ADRIANO DE LUCA (Professore di Architettura dei
Calcolatori all'Università Autonoma Metropolitana di Città del Messico)
GOFFREDO HAUS, MARCO MAIOCCHI, DANIELE MARINI, GIANCARLO
MAURI, CLAUDIO PARMELLI, ENNIO PROVERA

Testi
Eidos (TIZIANO BRUGNETTI), GOFFREDO HAUS, ENNIO PROVERA,
Etnoteam (ADRIANA BICEGO)

Tavole
Logical Studio Communication
Il Corso di Programmazione e BASIC è stato realizzato da Etnoteam
S.p.A., Milano
Computergrafica è stato realizzato da Eidos, S.c.r.l., Milano
Usare il Computer è stato realizzato in collaborazione con PARSEC S.M.
- Milano

Direttore Editoriale
ORSOLA FENGHI

Coordinatore settore scientifico
UGO SCAIONI

Redazione
MARINA GIORGETTI
LOGICAL STUDIO COMMUNICATION

Art Director
CESARE BARONI

Impaginazione
BRUNO DE CHECCHI
PAOLA ROZZA

Programmazione Editoriale
ROSANNA ZERBARINI
GIOVANNA BREGGÉ

Segretarie di Redazione
RENATA FRIGOLI
LUCIA MONTANARI

Corso Pratico col Computer - Copyright © sul fascicolo 1984 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Etas S.p.A., Milano - Copyright © sull'opera 1984 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Etas S.p.A., Milano - Prima Edizione 1984 - Direttore responsabile GIOVANNI GIOVANNINI - Registrazione presso il Tribunale di Milano n. 00082, del marzo 1984 - Iscrizione al Registro Nazionale della Stampa n. 00082, del 3, Foglio 489 del 20.9.1982 - Stampato presso lo Stabilimento Grafico del Gruppo Editoriale Fabbri S.p.A., Milano - Distribuzione per l'Italia: A. & G. Marco s.a.s., via Fortezza 27 - tel. 2526 - Milano - Pubblicazione periodica settimanale - Anno I - n. 15 - esce il giovedì - Speditezza in abb. postale - Gruppo 11/70. L'Editore si riserva la facoltà di modificare il prezzo nel corso della pubblicazione, se costretto da mutue condizioni di mercato.

CORSO PRATICO COL COMPUTER

Volume Secondo

CORSO PRATICO

Diretto da GIANNI DEGLI ANTONI



COL COMPUTER

FABBRI EDITORI

U
4

I
5

O
6

P

=
,
_

ENTER

J
1

L
3

%
'
`
~

?
,
0

:
/

NUM

Direttore dell'opera
GIANNI DEGLI ANTONI

Comitato Scientifico
GIANNI DEGLI ANTONI
Docente di Teoria dell'Informazione, Direttore dell'Istituto
di Cibernetica dell'Università degli Studi di Milano

UMBERTO ECO
Ordinario di Semiotica presso l'Università di Bologna

MARIO ITALIANI
Ordinario di Teoria e Applicazione
delle Macchine Calcolatrici presso
l'Istituto di Cibernetica dell'Università
degli Studi di Milano

MARCO MAIOCCHI
Professore Incaricato di Teoria e Applicazione delle Macchine
Calcolatrici presso l'Istituto di Cibernetica
dell'Università degli Studi di Milano

DANIELE MARINI
Ricercatore universitario presso l'Istituto di Cibernetica
dell'Università degli Studi di Milano

Curatori di rubriche
ADRIANO DE LUCA
GOFFREDO HAUS
MARCO MAIOCCHI
DANIELE MARINI
GIANCARLO MAURI
CLAUDIO PARPELLI
ENNIO PROVERA

Direttore Editoriale
ORSOLA FENGHI

Redazione
MARINA GIORGETTI
LOGICAL STUDIO COMMUNICATION

Art Director
CESARE BARONI

Impaginazione
BRUNO DE CHECCHI
PAOLA ROZZA

Programmazione Editoriale
ROSANNA ZERBARINI
GIOVANNA BREGGÉ

Segretarie di Redazione
RENATA FRIGOLI
LUCIA MONTANARI

Corso Pratico col Computer
Copyright © 1984 Gruppo Editoriale Fabbri, Bompiani,
Sonzogno, Etas S.p.A., Milano
Prima Edizione 1984

IL MODO MAGGIORE

La modulazione, la progressione, la regione tonale e alcuni brevi cenni sul ritmo.

Modulazione e progressione

Il passaggio da una tonalità ad un'altra all'interno di una melodia è chiamato *modulazione*. Questo procedimento è tanto più facile (e più breve) quanto più sono vicine le tonalità di partenza e di arrivo sul circolo delle quinte, come è già stato detto in precedenza. Ad esempio, è semplice passare dalla tonalità di *do maggiore* alla tonalità di *sol maggiore* (*sol-la-si-do-re-mi-fa diesis*) o di *fa maggiore* (*fa-sol-la-si bemolledo-re-mi*) poiché, a parte la nota alterata (*fa diesis* o *si bemol-*

le, rispettivamente), le tonalità sono formate dallo stesso insieme di altezze.

Una funzione musicale simile alla modulazione è la *progressione*, cioè il trasporto di una melodia su un altro grado della scala, mantenendo quindi una tonalità ma cambiando la sequenza di intervalli della melodia; un esempio ci chiarisce meglio il concetto di *progressione*.

Nel programma rappresentato è codificato il tema dal quarto movimento della *Sinfonia N° 9* di L. van Beethoven. Vediamo brevemente come è architettato questo programma.



pentagramma del tema dal quarto movimento della sinfonia n. 9 di L. van Beethoven

```

010 clear
020 REM programma per la generazio
ne del tema del quarto movimento
025 t=1
030 dim nt(13)
040 nt(0)=8368
050 nt(1)=7456
060 nt(2)=6642
070 nt(3)=6269
080 nt(4)=5586
090 nt(5)=4968
100 nt(6)=4433
101 nt(7)=4184
102 nt(8)=3728
103 nt(9)=3321
104 nt(10)=3134
105 nt(11)=2793
106 nt(12)=2484
107 nt(13)=2216
110 REM scelta della tonalita'
12)";tn
120 input "tonalita' (tra 1 e
121 if tn=2 then t=1.059463
122 if tn=3 then t=1.122462
123 if tn=4 then t=1.189207
124 if tn=5 then t=1.259921

```

```

125 if tn=6 then t=1.334840
126 if tn=7 then t=1.414212
127 if tn=8 then t=1.498307
128 if tn=9 then t=1.597401
129 if tn=10 then t=1.681793
130 if tn=11 then t=1.781797
131 if tn=12 then t=1.887749
132 REM richiesla progressione
133 input "progressione (tra 0 e
    6)";pr
137 REM codifica del tema musicale
138 input "tempo (tra 0 e 5)";tm
139 tmp=2^tm
140 REM struttura del tema: A (bat
    tute 1,2,3) B (4) A (5,6,7) C (8)
141 gosub 150
142 gosub 310
143 gosub 150
144 gosub 380
145 end
150 REM battuta 1 o 5
160 y=4*32/tmp
170 sound nt(2+pr)/t,y
180 y=2*32/tmp
190 sound nt(3+pr)/t,y
200 sound nt(4+pr)/t,y
205 REM battuta 2 o 6
210 sound nt(4+pr)/t,y
220 sound nt(3+pr)/t,y
230 sound nt(2+pr)/t,y
240 sound nt(1+pr)/t,y
250 REM battuta 3 o 7
260 y=4*32/tmp
270 sound nt(0+pr)/t,y
280 y=2*32/tmp
290 sound nt(1+pr)/t,y
300 sound nt(2+pr)/t,y
301 return
310 REM battuta 4
320 y=3*32/tmp
330 sound nt(2+pr)/t,y
340 y=32/tmp
350 sound nt(1+pr)/t,y
360 y=4*32/tmp
370 sound nt(1+pr)/t,y
371 return
380 REM battuta 8
390 y=3*32/tmp
400 sound nt(1+pr)/t,y
410 y=32/tmp
420 sound nt(0+pr)/t,y
430 y=4*32/tmp
440 sound nt(0+pr)/t,y
441 return

```

Le istruzioni 030-107 servono a memorizzare in un vettore i gradi di una scala per un'estensione di due ottave. Le istruzioni 110-132 predispongono il parametro *t* per l'effettuazione della trasposizione necessaria sulla base della tonalità che scegliamo (1 per il *re*, 2 per il *re diesis*, 3 per il *mi*, fino a 12 per il *do diesis*). Le istruzioni 132-133 memorizzano l'eventuale progressione che richiediamo: 0 nessuna progressione, 1 progressione di un grado, 2 di due gradi, fino a 6, cioè una progressione di sei gradi; indicando 1, per esempio, in una tonalità di *re* (come nel programma) da una sequenza *re-mi-fa diesis* otteniamo la sequenza *re-fa diesis-sol*, in cui si mantiene la tonalità originale ma si cambia la sequenza di intervalli. Le istruzioni 138-139 memorizzano il tempo che scegliamo. Le istruzioni 140-145 richiamano le subroutine secondo la struttura del tema musicale: *A B A C*, dove *A* indica il gruppo delle prime tre battute o il gruppo di battute 5-6-7 (sono uguali), *B* indica la quarta battuta e *C* indica l'ultima. Le istruzioni che seguono sono le subroutine corrispondenti alla codifica delle battute del tema musicale; all'interno di ogni istruzione SOUND viene calcolata l'eventuale progressione mediante lo spostamento del posizionamento all'interno del vettore in cui sono memorizzati i valori di altezza dei gradi della scala; l'operazione dipende dal parametro *pr* da noi impostato. Mediante la divisione per il parametro *t* viene calcolata l'eventuale trasposizione. Il valore delle durate viene calcolato a parte, di volta in volta.

Il programma ci permette di trasporre in qualunque tonalità e di effettuare progressioni di quanti gradi vogliamo (al tempo che vogliamo) del tema del quarto movimento.

Il concetto di regione

Abbiamo iniziato ad esplorare il concetto di tonalità e per meglio comprendere i significati più ampi è bene che consideriamo un concetto che estende la concezione classica di tonalità, il concetto di *regione*; questa impostazione ci permetterà di comprendere meglio l'unità dell'armonia di un brano musicale.

Ne parla A. Schoenberg in "Funzioni strutturali dell'armonia" (Il Saggiatore, 1967):

"Il mescolare note e accordi estranei in successioni che sarebbero altrimenti diatoniche era considerato dagli antichi teorici come una modulazione. Ma questa è una concezione ristretta e dunque antiquata della tonalità: non si dovrebbe parlare di modulazione finché una tonalità è lasciata del tutto per un certo periodo e finché non viene determinata, sia armonicamente sia tematicamente, un'altra tonalità. Il concetto di *regione* è una conseguenza logica del principio della *monotonalità*, secondo il quale ogni digressione dalla tonica viene considerata sempre nell'ambito della tonalità in base a un rapporto che può essere diretto o indiretto, vicino o lon-

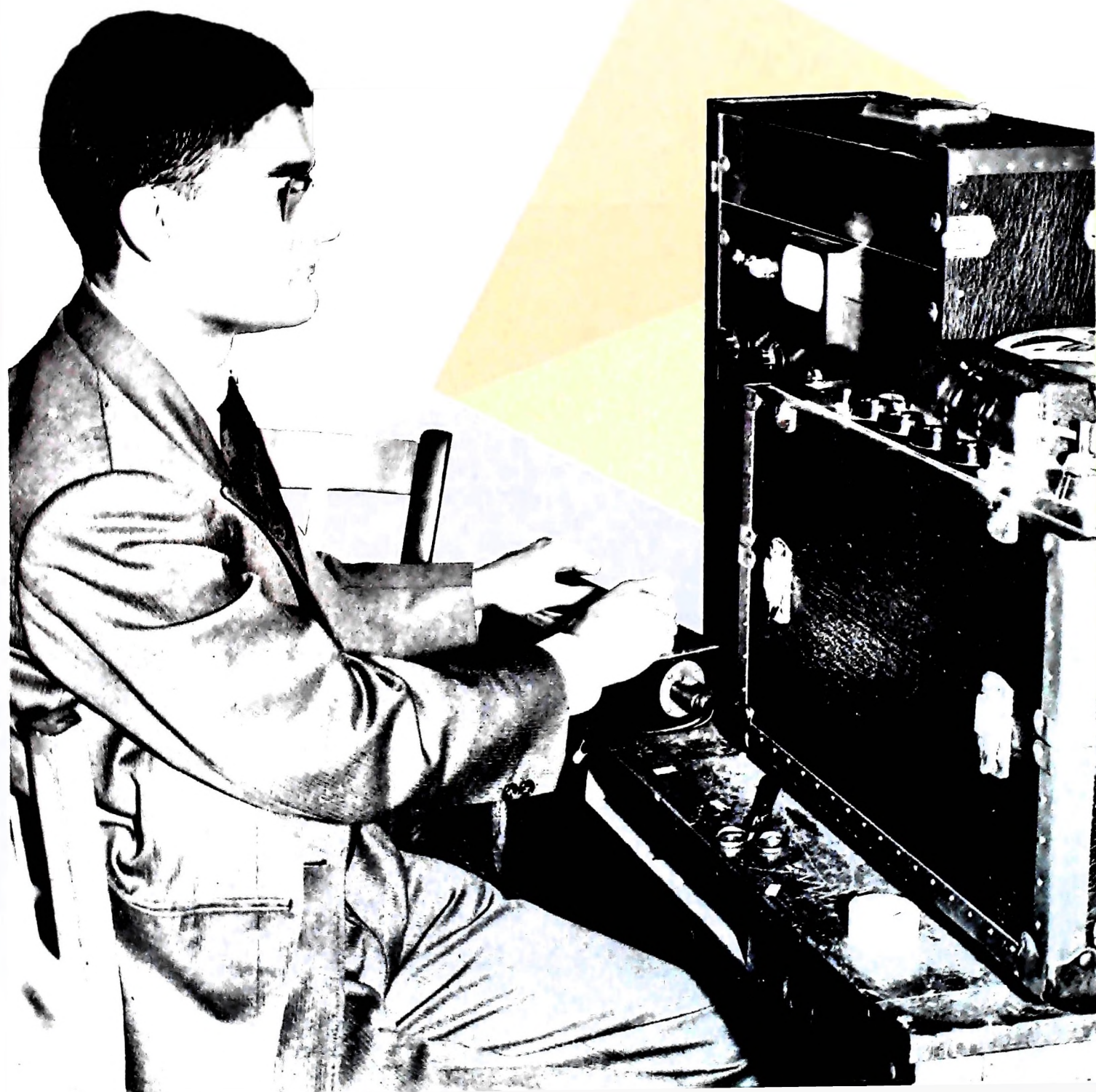
tano. In altre parole: in un pezzo di musica esiste *solo una tonalità*, e ogni sua parte che un tempo veniva considerata come tonalità diversa è soltanto una regione, un contrasto armonico nell'ambito della stessa tonalità. La monotonalità comprende la modulazione, cioè il movimento verso un altro tono che può essere anche svolto a sé stante, ma considera tali deviazioni come regioni della tonalità, subordinate al potere centrale di una tonica. È possibile così comprendere l'unità armonica nell'ambito di uno stesso pezzo".

Così abbiamo che le tonalità più *vicine* nel modo maggiore sono quelle della *dominante maggiore*, della *sottodominante maggiore*, della *sopradominante minore*, della *tonica minore*;

ad esempio, nella tonalità di *do maggiore* le regioni più vicine sono il *sol maggiore*, il *fa maggiore*, il *la minore*, il *do minore*. Queste quattro regioni sono dette *vicine e dirette*; allontanandosi armonicamente dalla tonica abbiamo regioni *indirette ma vicine, indirette, indirette e lontane, lontanissime*.

Il portarsi dalla tonica ad una regione (e ritorno) è una delle operazioni tipiche del compositore, realizzata mediante l'introduzione graduale di note alterate rispetto alla scala della tonalità di partenza fino allo stabilizzarsi della nuova regione (cioè dei legami armonici che caratterizzano un'altra tonalità). Non approfondiremo la trattazione delle regioni, ma riprenderemo l'argomento nella trattazione del *modo minore*.

Pierre Henry, uno dei pionieri della musica elettronica, con un sistema di registratori e mixer per sovrapporre ed elaborare i suoni.



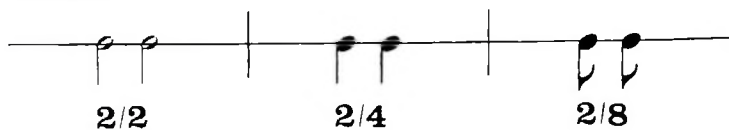
Un programma per la generazione del ritmo

Non abbiamo fino ad ora trattato un argomento di fondamentale importanza: il ritmo. Il ritmo è presente in moltissimi fenomeni naturali; nella musica esso ha la precisa funzione di regolare la pulsazione del fluire di eventi sonori.

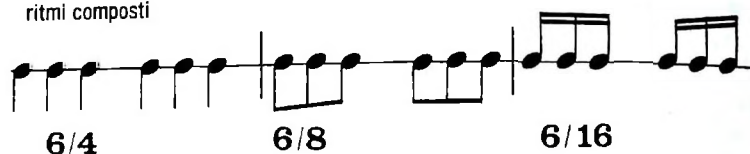
I ritmi più importanti sono:

- binari (cioè formati da due battiti);
- ternari (cioè formati da tre battiti);
- combinazioni di binari e/o ternari.

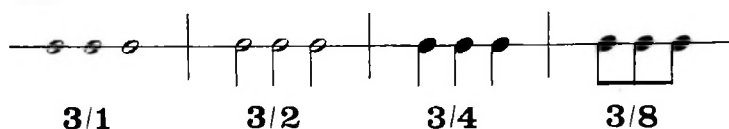
ritmi binari



ritmi composti



ritmi ternari



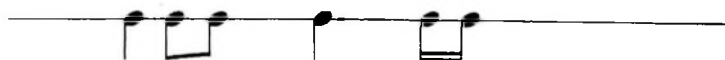
ritmi irregolari



ritmi a quattro tempi



Nell'illustrazione vediamo i ritmi più comuni; il primo battito di ogni unità ritmica è accentato; la prima nota di ogni battuta deve quindi differenziarsi dalle successive; per realizzare questo aspetto in modo soddisfacente dobbiamo poter controllare sia l'intensità che il timbro del suono; possiamo però simularlo ugualmente anche senza queste possibilità, specie se si tratta di un ritmo che prevede unità ritmiche in cui le note iniziali sono più lunghe delle successive; ad esempio, unità ritmiche come la seguente:



Possiamo allora costruire un programma che genera sequenze di altezze pseudocasuali organizzate secondo la struttura ritmica qui a lato indicata; nel vettore n poniamo i valori di durata relativi degli elementi della struttura ritmica e alla richiesta del tempo rispondiamo con un intero da 0 (lento) a 4 (veloce); il ciclo FOR scandisce il vettore ciclicamente assegnando di volta in volta i valori di durata memorizzati. Anche questo programma non termina spontaneamente, genera cioè una sequenza infinita di note sempre con la struttura ritmica contenuta nel vettore; se volessimo cambiare ritmo sarebbe quindi sufficiente modificare i valori relativi di durata memorizzati nel vettore n. Ecco il programma:

```

010 clear
015 REM memorizzazione di una
    struttura ritmica in 4/4
020 dim n(6)
030 n(1)=64
040 n(2)=32
050 n(3)=32
060 n(4)=96
070 n(5)=16
080 n(6)=16
083 REM generazione pseudocasuale
085 input "dammi un intero":INIT

```

```

090 rn=rnd(INIT)
091 REM scelta del ritmo
092 input "tempo (da 0 a 4)":tm
093 t=2^tm
095 REM generazione delle note
100 for i=1 to 6
102 rn=rnd(rn*100)
110 y=n(i)/t
120 x=rn*16383
122 sound x,y
125 next i
130 goto 100

```

Lezione 14

Gli array a due dimensioni

Vogliamo adesso costruire una tavola pitagorica. Se la limitiamo, per motivi di spazio, ai primi 5 interi si presenterà così:

	1	2	3	4	5
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25

Abbiamo così costruito una tabella che ha 5 righe e 5 colonne oltre alle intestazioni. In BASIC si può dichiarare così:

```
10 DIM P(5,5)
```

Pertanto con l'espressione $P(2,3)$ indichiamo l'elemento della seconda riga e della terza colonna, che contiene il valore del prodotto 2×3 .

In generale, l'elemento della riga I-esima e della colonna J-esima si indica con l'espressione $P(I,J)$ e nel nostro caso contiene il valore del prodotto $I \times J$.

Se pertanto vogliamo conoscere il prodotto di 5×3 dovremo semplicemente fare riferimento all'elemento $P(5,3)$, che contiene effettivamente il valore 15. Possiamo pertanto costruire il programma BASIC che costruisce la tavola pitagorica e la visualizza:

```
10 DIM T(5,5)
20 REM Facciamo scorrere tutte le righe
30 FOR I=1 TO 5
40 REM Per la riga di posto I, facciamo scorrere
tutte le colonne
50 FOR J=1 TO 5
60 REM Calcoliamo il valore della casellina
70 LET T(I,J)=I*J
80 NEXT J
90 NEXT I
100 REM Visualizziamo
110 REM Intestazione
115 CLS
120 PRINT "          TAVOLA PITAGORICA"
125 PRINT
130 PRINT "  ",1;2;3;4;5
135 PRINT "  ","-----"
140 FOR I=1 TO 5
145 PRINT I,
150 FOR J=1 TO 5
160 PRINT T(I,J);
```

Completata questa quattordicesima lezione del Corso di Programmazione e BASIC, siete in grado di eseguire gli esercizi:

VENDT.DO
VENDP.BA
contenuti nella cassetta "5 esercizi di programmazione" lato A.

I titoli seguiti dal suffisso DO corrispondono a testi, quelli seguiti da BA a programmi in BASIC.

Caricateli secondo le modalità che avete appreso.

```

170 NEXT J
180 PRINT
190 NEXT I
200 REM Fine programma

```

Si noti nel programma su riportato l'uso delle virgole e punti e virgole nell'istruzione PRINT. In particolare, l'effetto della virgola all'istruzione 145 è di far proseguire la visualizzazione sulla stessa riga lasciando uno spazio pari a 14 caratteri. L'effetto invece del punto e virgola all'istruzione 160 è ugualmente di far proseguire la visualizzazione sulla stessa riga ma senza l'effetto di tabulazione.

Si noti anche l'istruzione 180 che ha l'effetto di annullare il punto e virgola dell'istruzione 160, consentendo così di riprendere la visualizzazione dei valori relativi al numero successivo da riga nuova.

Array a N dimensioni

La tabella che abbiamo così costruito è un array a due dimensioni e abbiamo già visto esempi di array monodimensionali.

È anche possibile costruire array a tre dimensioni. Pensiamo per esempio a uno schedario che fornisca la temperatura media giornaliera per tutti i giorni di tutti i mesi degli ultimi 5 anni.

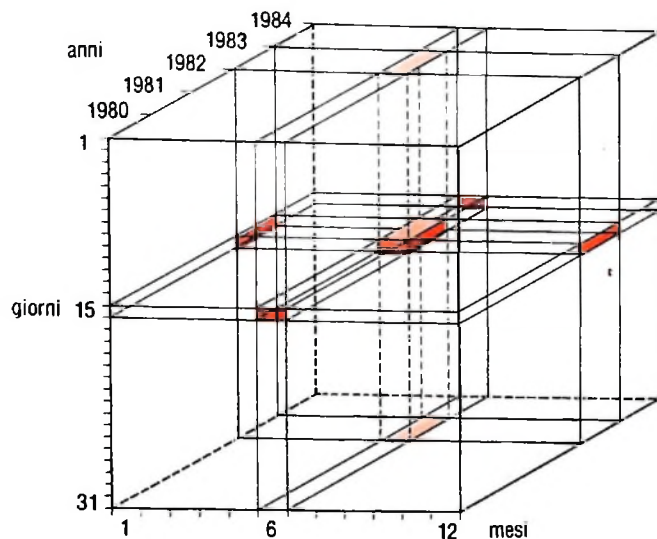
Una tabella per un solo anno, per esempio il 1984, sarebbe la seguente:

TEMPERATURE MEDIE 1984												
	1	2	3	4	5	6	7	8	9	10	11	12
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												
29												
30												
31												
32												
33												
34												
35												
36												
37												
38												
39												
40												
41												
42												
43												
44												
45												
46												
47												
48												
49												
50												
51												
52												
53												
54												
55												
56												
57												
58												
59												
60												
61												
62												
63												
64												
65												
66												
67												
68												
69												
70												
71												
72												
73												
74												
75												
76												
77												
78												
79												
80												
81												
82												
83												
84												
85												
86												
87												
88												
89												
90												
91												
92												
93												
94												
95												
96												
97												
98												
99												
100												

Se ora allineiamo 5 tabelle una dietro l'altra otterremo il seguente risultato:

che possiamo pensare come un array a tre indici, rispettivamente per individuare l'anno, il mese e il giorno. Possiamo rappresentare tale array come una scatola a

forma di parallelepipedo, le cui dimensioni corrispondono agli anni, ai mesi e ai giorni:



Tale scatola può essere idealmente scomposta in scatole più piccole ciascuna individuata da una terna di valori: ciascuna di tali scatole è una variabile, che nel nostro caso contiene il valore della temperatura media del giorno, mese e anno indicati (ovviamente per i mesi con meno di 31 giorni i giorni finali non vengono usati). Possiamo allora costruire il programma che introduce i dati nel casellario e che ricerca la temperatura media del giorno, del mese e dell'anno indicati:

```

10 DIM C(5,12,31)
20 FOR A=1 TO 5
30 REM Costruisce le informazioni per l'anno A-
esimo
40 FOR M=1 TO 12
50 FOR G=1 TO 31
60 REM Costruisce l'informazione per ogni gior-
no del mese m-esimo dell'anno A-esimo
65 PRINT "Inserire temperatura media in data:
":A+80-1;"/":M;"/":G
70 INPUT C(A,M,G)
80 NEXT G
90 NEXT M
100 NEXT A
110 REM Il casellario e' completo
120 REM Funzione di ricerca
130 CLS
140 PRINT "    INFORMATORE TEMPERATURE MEDIE"
150 PRINT
160 INPUT "Inserire anno,mese,giorno";A,M,G
170 REM Converte valore anno nel corrispondente
180 REM valore dell'indice dell'array
190 LET A=A-80+1
200 REM Controlla correttezza valori richiesti
210 IF A<=5 AND A>=1 GOTO 350
220 PRINT "VALORE DELL'ANNO SCORRETTO"
230 GOTO 160
240 IF M<=12 AND M>=1 GOTO 350
250 PRINT "VALORE DEL MESE SCORRETTO"

```

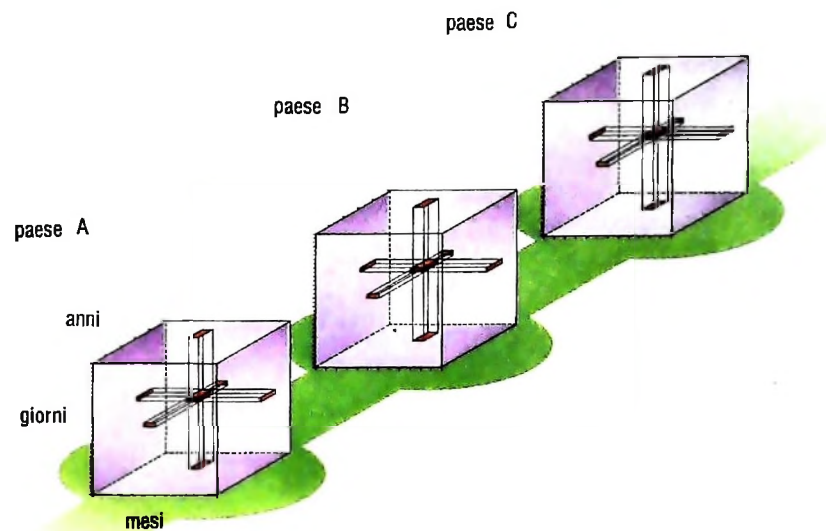
```

260 GOTO 160
270 IF G>=1 AND G<=31 GOTO 350
280 PRINT "VALORE ERRATO DEL GIORNO"
290 GOTO 160
350 PRINT "TEMPERATURA DEL ";G;"/";M;"/";80+A-1
;" ; ";C(A,M,G)
360 REM Fine programma

```

Poiché non è possibile usare indici di array che partano da valori arbitrari, come sarebbe utile nel nostro caso, abbiamo rappresentato i valori degli anni dal 1980 al 1984 con i valori da 1 a 5.

Come abbiamo visto siamo arrivati al concetto di array a tre dimensioni immaginando di allineare un certo numero di array a due dimensioni. Allo stesso modo, se allineiamo per esempio tre array a tre dimensioni otteniamo il seguente array a quattro dimensioni:



che potrebbe essere utile per rappresentare la temperatura media di un dato giorno, mese e anno in uno tra tre differenti paesi, che ancora una volta dovremo rappresentare con i valori da 1 a 3.

In generale potremo estendere questo criterio per costruire array a N dimensioni senza vincoli di sorta a eccezione dell'occupazione di memoria e anche della difficoltà di rappresentare mentalmente tabelle a più di tre o al massimo quattro dimensioni, con la conseguente difficoltà di risolvere problemi con tale tipo di soluzione.

Cosa abbiamo imparato

In questa lezione abbiamo visto:

- Il concetto di array a N dimensioni;
- La realizzazione BASIC di array a N dimensioni;
- L'uso di virgola e punto e virgola nell'istruzione PRINT per ottenere effetti di tabulazione.

LA SIMULAZIONE

Una metodologia di C.A.I. che permette di studiare sul video esperienze difficili o impossibili da realizzare nella realtà.

Con questo articolo concludiamo la presentazione e l'analisi delle principali metodologie impiegate nelle applicazioni dell'informatica alla didattica.

La simulazione è una strategia usata in molti campi, tipicamente in fisica, chimica, scienze naturali, economia.

L'elaboratore, attraverso un opportuno programma, diventa una macchina per simulare il comportamento di sistemi di qualunque tipo, cioè una specie di macchina per esperimenti. Le modalità con cui si svolgono questi esperimenti sono diverse. A volte la struttura è in larga misura prefabbricata, nel senso che l'allievo può variare soltanto pochi parametri che la caratterizzano. Altre volte invece l'allievo ha maggiore libertà, può per così dire costruire "pezzo per pezzo" lo schema e la struttura dell'esperimento e verificarne poi l'andamento dell'esecuzione.

Un esempio: la simulazione di circuiti elettronici

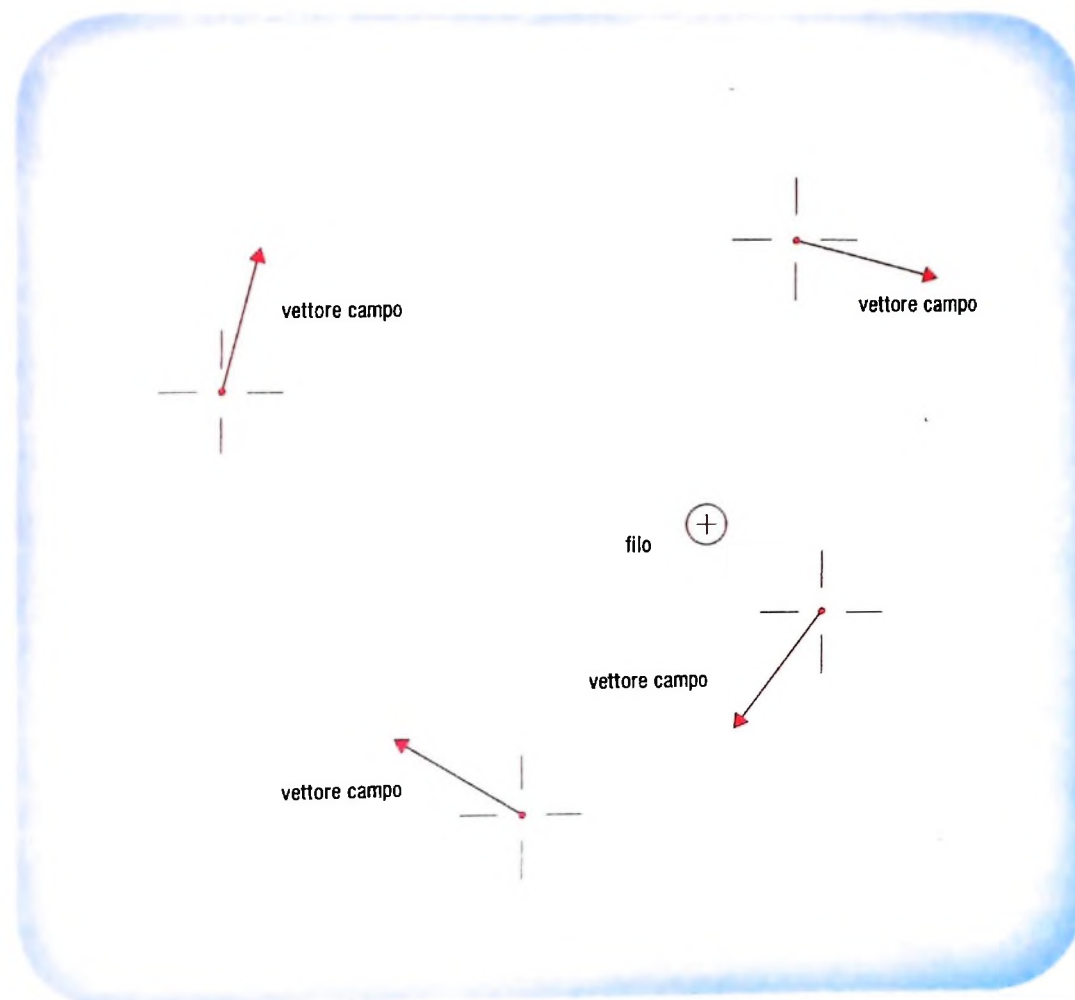
Un esempio di questo secondo tipo è il montaggio di un circuito elettronico da parte dell'allievo, attraverso l'indicazione sul video di posizione e collegamenti dei vari componenti (resistenze, capacità, induttanze, transistori ecc.).

Quando il circuito è pronto il programma permette di analizzarne il funzionamento.

È naturalmente l'allievo che stabilisce di volta in volta il valore della tensione o la forma degli impulsi in ingresso.

I risultati possono comparire sul video in modi diversi, sia grafici sia alfanumerici.

È immediatamente evidente che un programma di questo tipo consente di concentrare il lavoro degli allievi sui punti più interessanti del problema, cioè la rapida costruzione del cir-



Il vettore campo magnetico visualizzato in diversi punti nelle vicinanze di un filo percorso da corrente perpendicolare al piano del foglio o, nella simulazione, al piano dello schermo dell'elaboratore.

cuito e soprattutto il controllo del suo funzionamento, variando a piacere le condizioni esterne e anche modificandone in parte i componenti, se lo si desidera. È altrettanto chiaro che questo programma permette la sostituzione, almeno parziale, di attrezzature di laboratorio piuttosto costose.

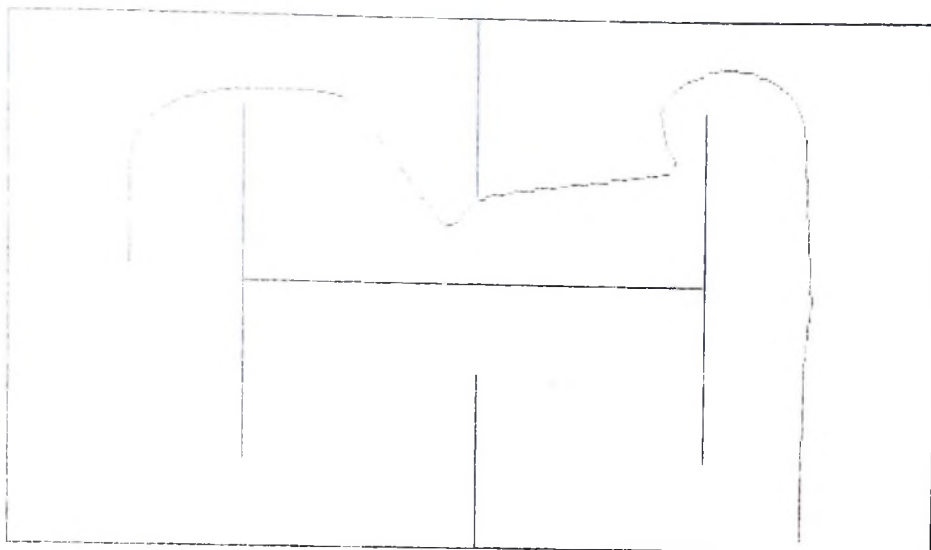
L'esplorazione di un campo magnetico

Vediamo un altro esempio, tratto dalla fisica: la simulazione del comportamento di un ago magnetico nei dintorni di un filo percorso da corrente ortogonale al piano dello schermo. L'allievo sposta un cursore grafico a croce, fino a una posi-

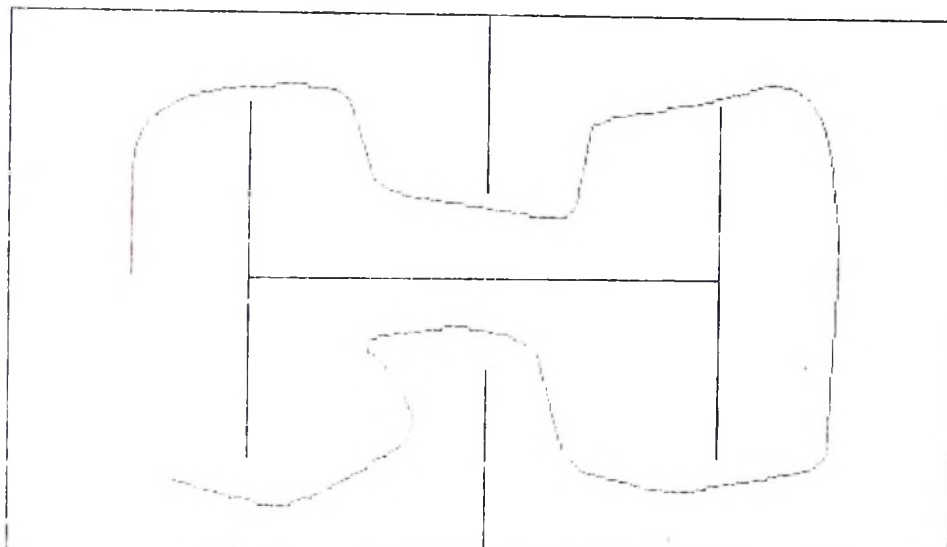
```

10 'Programma NEWTON. Simulazione moti piani con attrito
20 CLEAR
30 AZ=WINDOW (0,35)
40 '-----
50 WINDOW%1
60 SCALE 0,4,0,6
70 LINE (0,0)-(4,6),,B
80 LINE (1,1)-(1,5) : LINE (1,3)-(3,3)
90 LINE (3,1)-(3,5) : LINE (2,0)-(2,2)
100 LINE (2,4)-(2,6)
110 '-----
120 WINDOW%2
130 CLS
140 INPUT"Introduzione valore costante di attrito ":K
150 IF K>0 THEN 130
160 CLS
170 INPUT"Introduzione valore accelerazione di gravita' ":G
180 CLS
190 INPUT"Introduzione valore massimo propulsione ":PM
200 IF PM<=0 THEN 180
210 CLS
220 X=.5 : Y=3 : VX=0 : VY=0 : V=0 : VTH=0
230 '-----
240 CLS
250 INPUT"Introduzione valore propulsione (modulo) => ":P
260 IF P<0 OR P>PM THEN 240
270 INPUT"Introduzione valore propulsione (angolo) => ":TH
280 '-----
290 PX=P*COS(TH*3.14159/180) : PY=P*SIN(TH*3.14159/180)
300 WINDOW%1
310 FOR A=1 TO 10
320 RX=K*V^2*COS(VTH) : RY=K*V^2*SIN(VTH)
330 AX=RX+PX : AY=RY+PY+G
340 X1=X+AX*.1+.5*AX*.1^2 : Y1=Y+VY*.1+.5*AY*.1^2
350 VX=VX+AX*.1 : VY=VY+AY*.1
360 LINE (X,Y)-(X1,Y1) : X=X1 : Y=Y1
370 V=SQR(VX^2+VY^2) : VTH=ATN(VY/VX)
380 IF VX<0 AND VY>0 THEN VTH=VTH+3.14159
390 IF VX<0 AND VY<0 THEN VTH=VTH-3.14159
400 NEXT A
410 '-----
420 WINDOW%2
430 VTH1=INT(VTH*180/3.14159)
440 PRINT"X = "X,"Y = "Y
450 PRINT"V = "V,"V(Th) = "VTH1
460 A=INPUT"(1) : GOTO 240
470 END

```

$x = 3.40054$ $y = -1.23304$
 $u = .978193$ $u(Th) = -91$



$x = .558328$ $y = .788817$
 $u = .305732$ $u(Th) = 140$

Due percorsi sulla pista a inclinazione variabile tracciata sul video di M 20 dal programma NEWTON per la simulazione di moti piani.

Si noti che nel primo caso lo studente alla fine ha accelerato senza curvare, mentre nel secondo sta impostando la curva.

zione voluta; poi dà un segnale e sullo schermo compare una freccia, parallela al campo nel centro del cursore. Dopo aver visualizzato l'andamento del campo in diversi punti del piano, il sistema può intervenire con una strategia tutoriale chiedendo: "Hai osservato qualche regolarità nella direzione assunta dall'ago magnetico nei vari punti?". Se l'allievo risponde negativamente, il sistema lo può condurre a scoprire la forma delle linee di campo, in questo caso cerchi concentrici aventi per centro il filo percorso dalla corrente e ad esso perpendicolari.

La simulazione di moti piani

Vediamo infine un esempio sviluppato in modo completo, la simulazione di moti piani: si tratta di un'applicazione nel campo della fisica e al tempo stesso di un gioco interessante. Il moto simulato è quello di un corpo puntiforme su una superficie piana, con o senza la presenza di un campo gravitazionale uniforme, e con la presenza di resistenze passive (attriti vari e resistenza del mezzo).

Questo consente di immaginare il piano come orizzontale

(veicolo su strada, slitta su ghiaccio, nave), inclinato (discesa su pendio nevoso), oppure verticale (problemi balistici, missili teleguidati).

Il moto in un piano è descritto cinematicamente dalle equazioni classiche che determinano posizione e velocità.

$$x = x_0 + V_{ox} t + 0,5a_x t^2$$

asse x

$$V_x = V_{ox} + a_x t$$

asse y

$$y = y_0 + V_{oy} t + 0,5a_y t^2$$

$$V_y = V_{oy} + a_y t$$

Il moto viene scomposto in intervalli regolari della durata di un secondo. Chi guida la simulazione all'inizio di ciascun intervallo, in base alla posizione e alla velocità del corpo in moto ed alla traiettoria da seguire, effettua una manovra, fissando il valore della spinta in coordinate polari, cioè p (ρ , ϑ). Per determinare le componenti (a_x , a_y) della accelerazione è necessario anzitutto scomporre la spinta in coordinate rettangolari (p_x , p_y), quindi calcolare il valore delle resistenze passive, che si assumono complessivamente proporzionali al quadrato della velocità, attraverso una costante K negativa.

$$R_x = KV^2 \cos V\vartheta$$

$$R_y = KV^2 \sin V\vartheta$$

Notiamo che V è la velocità angolare in coordinate polari. Quindi il valore dell'accelerazione risulta dalla somma della propulsione, delle resistenze passive e, per l'asse y, della accelerazione di gravità (che può anche essere uguale a zero, nel caso di simulazione di moti su piani orizzontali). Vediamo in formule:

$$a_x = p_x + R_x$$

$$a_y = p_y + R_y + g$$

Entro ciascun intervallo di un secondo la spinta (più eventualmente la gravità) viene mantenuta costante. Il programma calcola, in intervalli di tempo di un decimo di secondo, la nuova posizione e la nuova velocità, tenendo conto anche della variazione della accelerazione dovuta alla variazione delle resistenze passive, legate alla velocità.

Al termine dell'intervallo di un secondo viene visualizzato il valore della posizione, in coordinate rettangolari, e quello della velocità, in coordinate polari, mentre viene tracciata sullo schermo una linea che corrisponde alla traiettoria.

Il programma

Abbiamo visto rappresentato nelle pagine precedenti il programma completo, sviluppato per l'elaboratore M20 della Olivetti, della simulazione di moti piani con attrito.

Le righe dalla 50 alla 90 si riferiscono al tracciamento della "pista" sullo schermo. Si tratta di una pista piuttosto semplice, come si può vedere anche dalle figure che mostrano l'andamento di due simulazioni. Le righe dalla 110 alla 210 effettuano la inizializzazione del programma, introducendo il valore della costante di attrito, il valore della gravità (che può anche essere zero), il valore massimo del modulo della propulsione e fissano la posizione e la velocità iniziali. Le righe dalla 230 alla 260 corrispondono alla introduzione del valore della propulsione in coordinate polari, cioè in modulo e angolo. Le righe dalla 280 alla 390 si riferiscono al calcolo della nuova posizione e della nuova velocità e al tracciamento sul grafico della traiettoria corrispondente. Infine le righe dalla 410 alla 440 visualizzano la nuova posizione in coordinate rettangolari e la nuova velocità in coordinate polari.

Analizziamo i due esempi di grafici. Nel primo caso, ad un certo punto, il veicolo è uscito dalla pista mentre nel secondo ha compiuto un giro di pista quasi completo.

Facciamo notare che la pista è stata percorsa in senso orario e che le figure rappresentano una copia identica di quanto appare sul video: possiamo notare, rappresentati accanto alle illustrazioni, i valori relativi all'ultima posizione che è stata assunta dal veicolo e alla velocità corrispondente in coordinate polari.

Vantaggi e svantaggi

L'impiego di questa metodologia presenta vantaggi abbastanza evidenti: anzitutto permette la simulazione di situazioni difficilmente accessibili o riproducibili per ragioni di costo, pericolo o tempi troppo lunghi.

Inoltre la strategia offre la possibilità di effettuare una grande quantità di esperimenti in tempi brevi e con molta maggiore facilità di indagine rispetto a quella di una situazione reale.

Tra gli svantaggi segnaliamo soprattutto il pericolo di considerare il laboratorio simulato come un perfetto sostituto del laboratorio reale.

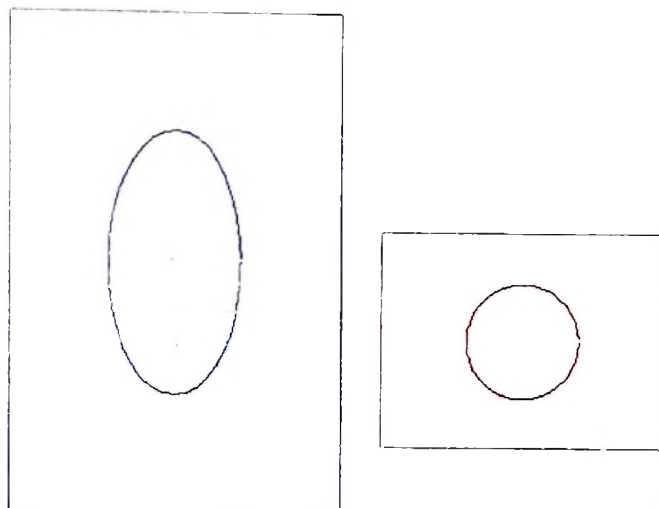
Non possiamo né dobbiamo dimenticare il valore insostituibile di una esperienza reale: l'informatica si pone perciò come uno strumento utilissimo per integrare quella esperienza quando tempi, pericoli o costi non ne consentirebbero o non ne consiglierebbero lo svolgimento diretto.

Concludiamo il panorama relativo alle diverse metodologie impiegate nella didattica mediante l'informatica, ricordando che diversi tipi di esse possono anche utilmente interagire in un singolo programma che sappia sfruttare al meglio le possibilità offerte da ciascuna.

Ripetiamo ancora una volta, però: si tratta di un campo sempre aperto alla ricerca e alla sperimentazione, anche se parecchio è già stato fatto.

UN ESERCIZIO SULLE TRASFORMAZIONI DI WINDOW

Esercitiamoci a disegnare un cerchio nella window e a riprodurlo in una viewport, su plotter o su schermo.



Esempio di un output del programma. Un cerchio può deformarsi in ellisse a seconda dei rapporti fra i lati delle finestre (window e viewport).

In questa lezione vogliamo realizzare un'applicazione sulle trasformazioni di window che abbiamo introdotto per via teorica precedentemente. A tal fine inventiamoci un esercizio, cioè disegnare un cerchio nella window e riprodurre tale figura in una viewport, definita o sul plotter o sullo schermo. Facciamo subito un'osservazione: abbiamo visto come i rapporti fra i lati delle due finestre (window e viewport) debbano essere uguali affinché il disegno non risulti deformato; bene, quest'informazione ci può permettere di utilizzare un'apparente limitazione in maniera divertente. Ossia, vogliamo poter controllare i rapporti fra i lati di entrambe le finestre, al fine di poter disegnare nella viewport, modificando opportunamente tale rapporto, un cerchio deformato, ossia un'ellisse. Vediamo quali sono i punti concettualmente più importanti nella stesura di un programma che realizzi le nostre richieste. Bisogna disporre di un algoritmo per disegnare un cerchio, che già abbiamo visto; di equazioni di trasformazione di window per passare dalle coordinate del mondo alle coordinate dello schermo o del plotter che conosciamo; di una serie di istruzioni (subroutine) richiamabili all'interno del programma a seconda che si voglia uscire sul video o sul plotter, ossia in grado di gestire la periferica di output prescelta. Il listato del programma che realizza tutto ciò viene fornito qui di seguito.

```

10 CLS
20 PIGRECA=3.1415
30 REM DEFINISCO LA WINDOW
40 INPUT "QUALI SONO LE COORDINATE DEI LATI
DELLA WINDOW WXS,WXD,WYB,WYA";W1,W2,W3,W4
50 REM CALCOLO LE COORDINATE DEL CENTRO DELLA
WINDOW
60 XC=FIX((W2-W1)/2)+W1
70 YC=FIX((W4-W3)/2)+W3
80 IF (W2-W1)<(W4-W3) THEN Z=(XC-W1) ELSE
Z=(YC-W3)
90 PRINT "DAMMI IL RAGGIO DEL CERCHIO (<"Z")"
91 INPUT R
100 INPUT "VUOI USCIRE SU VIDEO (V) O SU
PLOTTER (P)";A$
110 IF A$<>"V" AND A$<>"P" THEN 1000
120 REM CALCOLO IL RAPPORTO FRA
121 REM LARGHEZZA ED ALTEZZA DELLA
122 REM WINDOW.
130 RW=(W2-W1)/(W4-W3)
132 CLS
135 PRINT "IL RAPPORTO FRA I LATI ";
136 PRINT "DELLA WINDOW E'";RW
140 REM DEFINISCO LA VIEWPORT
142 PRINT
145 INPUT "QUALI SONO LE COORDINATE DEI LATI
DELLA VIEWPORT VXS,VXD,VYB,VYA";V1,V2,V3,V4
150 IF (V2-V1)>480 THEN 800
155 IF (V2-V1)>239 AND A$="V" THEN 850
160 IF (V4-V3)>63 AND A$="V" THEN 900
170 REM CALCOLO IL RAPPORTO FRA
180 REM LARGHEZZA ED ALTEZZA DELLA
190 REM VIEWPORT.

```

Animazione al computer

Tra i tanti campi applicativi della grafica al computer ve ne è uno di straordinario interesse per gli sbalorditivi risultati che permette di ottenere: si tratta della animazione, ovvero della creazione di filmati in cui oggetti generati col computer si muovono sullo schermo come in un vero e proprio cartone animato o come se una astronave viaggiasse in uno spazio fantastico.

Uno dei più famosi esempi di animazione a computer è il film TRON della Walt Disney, presentato alla fine del 1981, in cui sono state messe alla prova le capacità di animazione a computer delle maggiori società americane del settore.

Già da alcuni anni tuttavia c'erano stati esempi di computer animation, come la chiamano in USA. Attualmente il settore più coinvolto è quello della pubblicità cinematografica o televisiva, in quanto gli effetti che si possono ottenere sono certamente i più adatti per comunicare messaggi con efficacia certa.

Come si fa la computer animation? Dipende dal livello al quale si vuole arrivare. Utilizzando infatti un personal computer è possibile fare muovere sullo schermo un piccolo disegno, che rappresenta ad esempio una rana che salta in uno stagno e afferra con la lingua una mosca. Le tecniche che vengono utilizzate in questo caso le troviamo applicate nei videogiochi che funzionano sui personal e home computer. I limiti di questa animazione, che peraltro può essere straordinariamente divertente, sono quelli della velocità, piuttosto limitata, e della qualità delle figure. La velocità è limitata dalla capacità dell'elaboratore stesso, il quale riesce a far muovere sullo schermo solo disegni piuttosto piccoli, lasciando fermo lo sfondo. La qualità è limitata dalla precisione del display, limitato nel numero di pixel e nel numero di colori.

Per poter fare muovere su un display un piccolo disegno in modo da dare l'illusione di movimento, occorre rigenerare almeno una dozzina di volte al secondo l'intera immagine, e ogni volta essa deve essere cambiata. Il cambiamento dell'immagine viene effettuato sui piccoli computer copiando una certa area di memoria di schermo, in cui è descritta la figura, in un'altra area un poco più spostata, e ciò viene ripetuto facendo percorrere alla figura la strada voluta.

Una operazione di copiatura da memoria a memoria, nei sistemi più veloci, richiede qualche microsecondo ed è quindi evidente che si possono eseguire solo poche centinaia di operazioni di copiatura le 12 volte al secondo necessarie.

Con trucchi ed acrobazie, gli esperti di videogiochi riescono ad ingannare l'elaboratore facendogli fare più operazioni insieme o semplificando la descrizione delle figure in movimento. Il risultato sono appunto i videogiochi che possiamo utilizzare a casa collegandoci col nostro televisore.

Del tutto diverso è ciò che si può fare con elaboratori più potenti, collegati a display di alta definizione. In questo caso solo i costi pongono limiti alla capacità di emulare il movimento reale nello spazio.

Un caso interessante è la simulazione del volo del PIONEER II studiata nei laboratori Lawrence Livermore da J. Blinn, il quale ha creato alcuni filmati in cui si vede ciò che si vedrebbe stando a bordo della sonda spaziale. Per far ciò Blinn ha risolto alcuni problemi, primo tra tutti il calcolo delle traiettorie dei pianeti del sistema solare e della stessa sonda spaziale. Ancor più interessante è però il secondo problema risolto da Blinn: la raffigurazione della superficie dei pianeti stessi. Partendo da fotografie scattate da altre sonde o da telescopi ottici in anni precedenti, con operazioni matematiche, ha eseguito il trasferimento dell'aspetto di una porzione della superficie ad esempio di Marte sull'intero pianeta, facendo in pratica il procedimento inverso che si esegue tracciando una carta del mondo a partire dalle coordinate latitudine e longitudine delle coste e dei continenti.

Il problema della computer animation è dunque un problema di hardware e di

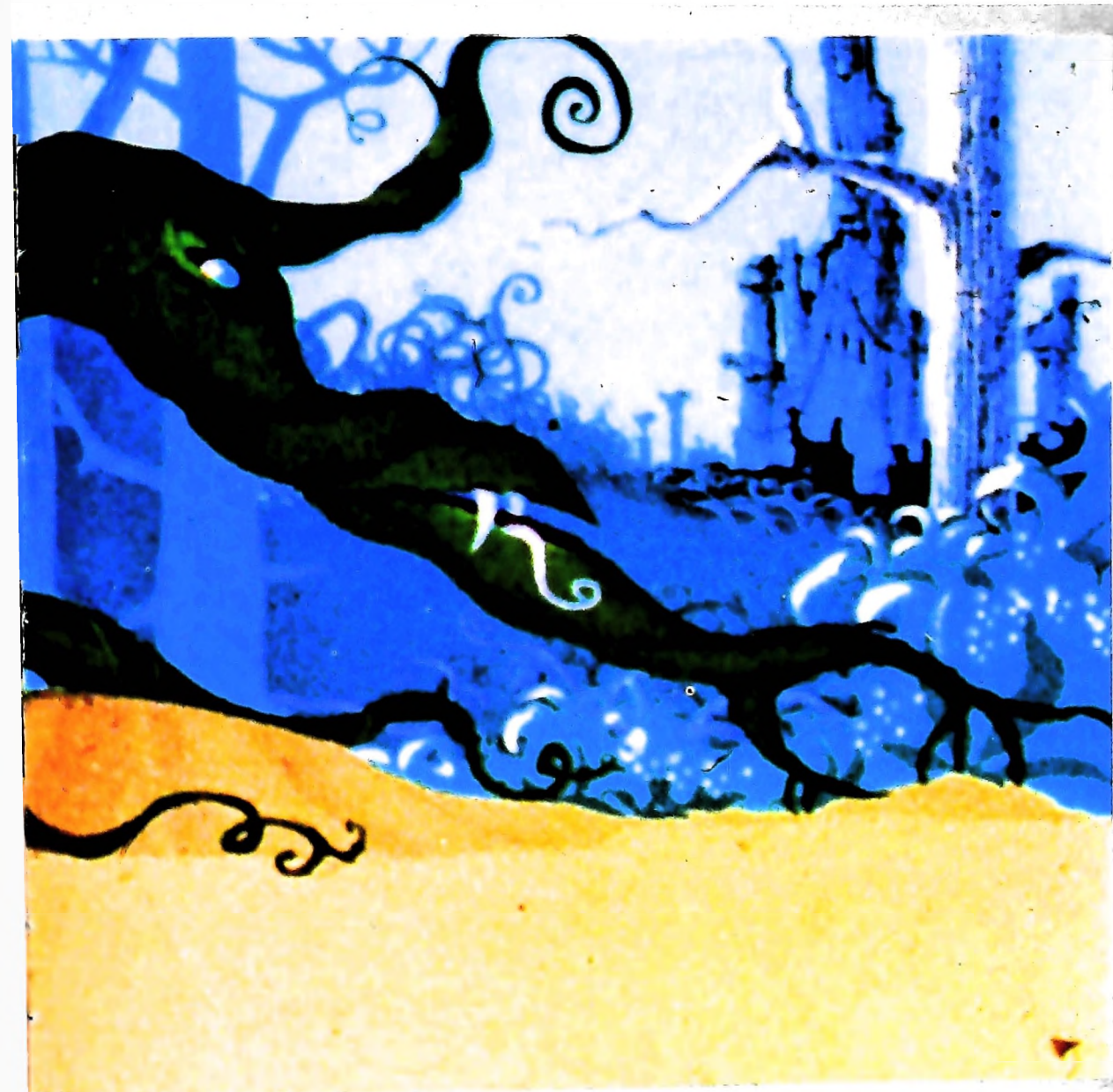
AMC ASSOCIATION - ARCHIVIO EIDOS - © WALT DISNEY PRODUCTIONS



software nello stesso tempo. L'aspetto hardware permette di raggiungere livelli qualitativi svariati, mentre quello software consente di simulare movimenti di pianeti o di personaggi e movimenti di una macchina da presa che si muove nello spazio.

L'aspetto più complesso della computer animation è tuttavia mettere insieme tutti questi vari aspetti, creando sistemi di facile uso per persone che con l'informatica non hanno nulla a che fare, come gli sceneggiatori o i registi. Le "bucce di banana" di cui è cosparsa la strada verso un'animazione completa e integrata sono tantissime, e forse la più grossa è quella della riproduzione finale delle immagini sulla pellicola cinematografica. C'è infatti spesso da domandarsi se valga la pena di realizzare col computer effetti che un abile animatore o un esperto in trucchi cinematografici riesce sovente a fare con risultati più che apprezzabili.

ACM ASSOCIATION - ARCHIVIO EIDOS - © WALT DISNEY PRODUCTIONS



Disegnare cartoni animati è un'attività estenuante, che impegna da decenni legioni di disegnatori costretti a riprodurre uno stesso oggetto o personaggio decine e decine di volte con modifiche minime per riprodurre nella maniera il più possibile gradevole all'occhio un movimento che dura pochi attimi. È comprensibile, quindi, che le case cinematografiche, desiderose di limitare le enormi spese legate alla realizzazione di un film a cartoni animati anche breve, si rivolgano all'elaboratore, uno strumento ormai famoso per la sua capacità di eseguire rapidissimamente lavori monotoni e ripetitivi. Fino ad oggi, però, i metodi di animazione al calcolatore non sono ancora economicamente competitivi con le tecniche tradizionali, tranne forse per la realizzazione di effetti particolari.

```

200 RV=(V2-V1)/(V4-V3)
210 PRINT "IL RAPPORTO FRA I LATI ";
211 PRINT "DELLA VIEWPORT E'";RV
220 IF RW<>RV THEN 1100
225 REM CALCOLO I FATTORI DI SCALA
226 REM DELLA TRASFORMAZIONE DI WINDOW.
230 SX=(V2-V1)/(W2-W1)
235 SY=(V4-V3)/(W4-W3)
240 CLS
250 N=36
252 I=2*PIGRECA/N
253 REM CALCOLO LE COORDINATE WINDOW
254 REM INIZIALI.
255 XVECCHIA=R*COS(0)+XC
256 YVECCHIA=R*SIN(0)+YC
257 XVECCHIA=SX*(XVECCHIA-W1)+V1
258 YVECCHIA=SY*(YVECCHIA-W3)+V3
260 REM DISEGNO LA VIEWPORT
263 IF A$="V" THEN GOSUB 1150 ELSE GOSUB 1170
268 FOR K=1 TO N
270 ANG=ANG+I
275 REM CALCOLO LE COORDINATE WINDOW
276 REM DEL CERCHIO.
280 XW=R*COS(ANG)+XC
290 YW=R*SIN(ANG)+YC
300 REM CALCOLO LE COORDINATE SCHERMO
310 XS=SX*(XW-W1)+V1
320 YS=SY*(YW-W3)+V3
330 IF A$="V" THEN GOSUB 400 ELSE GOSUB 600
340 XVECCHIA=XS
350 YVECCHIA=YS
360 NEXT K
365 IF A$="P" THEN 1200
370 GOTO 1220
400 REM SUBROUTINE PER USCIRE SU VIDEO.
410 LINE (XVECCHIA,YVECCHIA)-(XS,YS)
420 RETURN
600 REM SUBROUTINE PER USCIRE SU
601 REM PLOTTER.
620 LPRINT "D"+STR$(XVECCHIA)+",""+STR$(YVECCHIA)
+"", "+STR$(XS)+",""+STR$(Y
640 RETURN
800 PRINT "LA VIEWPORT E' TROPPO LARGA SIA PER
LO SCHERMO CHE PER IL PLOT
810 GOTO 145
850 PRINT "LA VIEWPORT E' TROPPO LARGA PER LO
SCHERMO. RIDEFINISCILA"
860 GOTO 145
900 PRINT "LA VIEWPORT E' TROPPO ALTA PER
LO SCHERMO. RIDEFINISCILA"
910 GOTO 145
1000 PRINT "ERRORE. RIPROVA"
1010 GOTO 100
1100 PRINT "LA WINDOW E LA VIEWPORT ";
1101 PRINT "NON SONO PROPORZIONALI.";
1102 PRINT "IL DISEGNO SARA' DISTORTO ";
1103 PRINT "RISPETTO ALL'ORIGINALE.";
1104 PRINT "VUOI MODIFICARE I DATI ";
1105 PRINT "DELLA VIEWPORT ?";
1106 INPUT " (S=SI;N=NO)";TS
1110 IF T$="N" THEN 225
1120 V2=RW*(V4-V3)+V1
1140 GOTO 225
1150 LINE (V1,V3)-(V2,V4),1,B
1160 RETURN
1170 LPRINT CHR$(18)
1175 LPRINT "C2"
1177 LPRINT "M"+STR$(V1)+",""+STR$(V3)
1180 LPRINT "D"+STR$(V1)+",""+STR$(V3)+",""+STR$
(V2)+",""+STR$(V3)+",""+STR$
1185 LPRINT "H"
1187 LPRINT "C3"
1189 LPRINT "M"+STR$(XVECCHIA)+",""+STR$
(YVECCHIA)
1190 RETURN
1200 LPRINT CHR$(17)
1220 END

```

Vediamo ora qualche semplice commento al programma, trascurando di soffermarci sulle istruzioni autoevidenti.

Dopo aver definito le coordinate della window, è indispensabile sapere quelle del suo punto centrale (istruzioni 60 e 70), in cui centeremo il cerchio. La funzione FIX, nelle istruzioni 60 e 70, tronca la parte decimale del risultato di un calcolo. Nell'istruzione 80 si effettua una serie di controlli per determinare la misura massima del raggio.

Dopo la scelta della periferica di uscita (video o plotter), si calcola il rapporto fra i lati della window e lo si evidenzia sullo schermo; vengono poi richieste le coordinate della viewport e si effettuano (istruzioni 150, 155, 160) una serie di controlli sulle dimensioni della viewport che deve essere compatibile con le dimensioni fisiche della periferica di uscita prescelta. Segue l'analogo calcolo del rapporto fra i lati della viewport e si confronta il risultato ottenuto (RV) con il corrispondente della window (RW). Se i due valori sono diversi (istr. 220) si segnala il fatto e viene chiesto se si desidera ottenere il disegno distorto oppure se si preferisce correggere automaticamente le proporzioni della viewport (istr.1120). Il passo successivo (istr. 230 e 235) è quello di calcolarsi i fattori di scala per le equazioni di trasformazione di window, che abbiamo illustrato precedentemente, e di memorizzarle in due costanti SX e SY.

Nell'istruzione 250 si definisce il numero di punti che costituiranno il cerchio; per ottenere una precisione maggiore nel disegno basterà aumentare il valore della costante N. Con le istr. 255 e 256 si calcolano le coordinate iniziali del cerchio nella window, mentre con le istr. 257 e 258 quelle nella viewport. Per chiarezza, si disegna anche la viewport utilizzando l'istruzione 263: se la periferica deve essere il video di M10, viene allora richiamata, mediante il comando GOSUB, la subroutine costituita dalle due istruzioni 1150 e 1160; se la periferica è il plotter viene invece richiamata la subroutine costituita dalle istruzioni dalla 1170 alla 1190. Entrambe si chiudono con il comando di RETURN, che permette al programma di ritornare automaticamente all'istruzione successiva a quella in cui è stata richiamata la subroutine stessa (nel nostro caso all'istr. 268). Il vantaggio di una tale soluzione è quello di poter richiamare un certo numero di istruzioni da più punti del programma, senza dover duplicare più volte gli stessi comandi. Nel nostro esempio le due subroutine vengono chiamate una sola volta per ogni esecuzione, e quindi si poteva utilizzare anche il semplice GOTO invece dei due comandi GOSUB e RETURN, ma ci è parso istruttivo dare un primo esempio di questo modo di procedere, che verrà sempre più utilizzato prossimamente.

Dall'istruzione 268 alla 360 vengono effettuati i calcoli delle coordinate del cerchio nella window e le corrispondenti coordinate nella viewport, utilizzando sempre le equazioni di trasformazione. La stampa avviene utilizzando due subroutine diverse a seconda della periferica scelta.

Abbiamo così visto come utilizzare in un esempio pratico i concetti di window e di viewport. Il passo successivo per arricchire il nostro pacchetto grafico sarà quello di affrontare il problema del clipping. Tratteremo questo argomento prossimamente.

Olivetti M10 vuol dire disporre del proprio ufficio in una ventiquattrore. Perché M10 non solo produce, elabora, stampa e memorizza dati, testi e disegni, ma è anche capace di comunicare via telefono per spedire e ricevere informazioni. In grado di funzionare a batteria oppure collegato all'impianto elettrico, M10 mette ovunque a disposizione la sua potenza di memoria, il suo display orientabile a cristalli liquidi capace anche di elaborazioni grafiche, la sua tastiera professionale arricchita da 16 tasti funzione.



Ma M10 può utilizzare piccole periferiche portatili che ne ampliano ancora le capacità, come il micro-plotter per scrivere e disegnare a 4 colori, o il registratore a cassette per registrare dati e testi, o il lettore di codici a barre. E in ufficio può essere collegato con macchine per scrivere elettroniche, con computer, con stampanti. Qualunque professione sia la vostra, M10 è in grado, dovunque vi trovate, di offrirvi delle capacità di soluzione che sono davvero molto grandi. M10: il più piccolo di una grande famiglia di personal.

PERSONAL COMPUTER OLIVETTI M10

L'UFFICIO DA VIAGGIO



Anche in leasing con Olivetti Leasing.

olivetti

Per informazioni rivolgetevi al negoziante contrassegnato da Olivetti M10 Punto di Vendita o inviate il coupon a Olivetti, Divisione Personal Computer, Via Meravigli 12, 20124 Milano.

NOVE COGNOME
VIA N.
CAPOCITTA
TELEFONO

UN NUOVO MODO DI USARE LA BANCA.

CONSALENZA

GLI INVESTIMENTI CON VOI E PER VOI DEL BANCO DI ROMA.

Il Banco di Roma non si limita a custodire i vostri risparmi. Vi aiuta anche a farli meglio fruttare. Come? Mettendovi a disposizione tecnici e analisti in grado di offrirvi una consulenza di prim'ordine e di consigliarvi le forme di investimento piú giuste. Dai certificati di deposito ai titoli di stato, dalle obbligazioni alle azioni, il Banco di Roma vi propone professionalmente le varie opportunità del mercato finanziario. E grazie ai suoi "borsini", vi permette anche di seguire, su speciali video, l'andamento della Borsa minuto per minuto.

Se desiderate avvalervi di una gestione qualificata per investire sui piú importanti mercati mobiliari del mondo, i fondi comuni del Banco di Roma, per titoli italiani ed esteri, vi garantiscono una ampia diversificazione.

Inoltre le nostre consociate Figeroma e Finroma forniscono consulenze per una gestione personalizzata del portafoglio e per ogni altra esigenza di carattere finanziario.

Veniteci a trovare, ci conosceremo meglio.

 **BANCO DI ROMA**
CONOSCIAMOCI MEGLIO.

