

8 CORSO PRATICO COL COMPUTER

421602

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

NELLA LIBRERIA DI
SOFTWARE
IL GIOCO
DEL MASTER-MIND

A B U R

FABBRI
EDITORI

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

I tasti predefiniti con i comandi in Basic rendono ancora più facile la programmazione dando al neofita un piccolo grande aiuto.



SHARP PC-1245 costa solo L. 189.000 + IVA

Facile inserimento dei più usati comandi in BASIC.

Il sistema d'impostazione istantanea dei comandi in BASIC facilita notevolmente sia l'operatività del Computer che la programmazione dello stesso.

I 18 tasti alfabetici sono preprogrammati con i più comuni comandi BASIC; ad esempio al tasto A è abbinato il comando INPUT, al tasto F il comando GOTO ed al tasto Z il comando PRINT e così via. Questo, durante la programmazione, evita di dover continuamente scrivere i vari comandi per intero, garantendovi meno errori, una maggior velocità nella programmazione e facilitandovi l'apprendimento dei termini in BASIC.

18 tasti definibili dall'utilizzatore per etichettare i programmi

Questa è una interessante possibilità che vi permette di accedere immediatamente ai programmi più usati. Potete etichettare fino a 18 programmi assegnando loro un tasto. Per richiamare un programma basta premere il tasto assegnato.

Potenza portatile a vostra disposizione.

Potrete avere la potenza del Computer ovunque vi serve.

Il PC-1245 ha una capacità di 24 KBytes di ROM per governare l'intero sistema e 2,2 KBytes di RAM per programmazione.

C-MOS CPU a 8-bit

Il PC-1245 usa la stessa CPU dei Personal da tavolo. Questo, oltre ad una alta velocità d'elaborazione, vi garantisce una grande efficienza.

Tastiera tipo macchina da scrivere

Grazie alla disposizione dei tasti come sulle macchine da scrivere vi sarà facile impostare velocemente i vostri programmi.

Memoria protetta

Uno speciale sistema d'alimentazione protegge la memoria del PC-1245 anche a macchina spenta. Questo vi consente d'interrompere, in qualsiasi momento, un programma od un calcolo. Potete ricominciare quando volete dall'ultimo inserimento - anche dopo giorni - senza correre il rischio d'aver perso un dato o una istruzione.

Funzione PASS

Potete assegnare un codice segreto al programma in memoria

ottenendo così una completa protezione dello stesso. Non sarà possibile listarlo, modificarlo o vederlo. Si potrà solo elaborare.

Visore a 16 caratteri con matrice a punti 5x7

Il visore è in grado di visualizzare contemporaneamente fino a 16 caratteri. Ogni carattere appare chiaramente leggibile grazie alla matrice a punti di 5 per 7. Potete anche regolare la luminosità del visore per avere la miglior lettura.

Selezione tra RUN e programma

Un interruttore consente l'immediata selezione tra il modo Run e PROGRAMMA.

Un optional importante

Per dare al vostro programma una chance in più potete integrare il PC-1245 con la CE-125, stampante e microregistratore opzionale, rendendo il sistema ancora più completo. La possibilità di stampare e registrare su nastro i vostri programmi e dati vi sarà utile per conservare sia i risultati della elaborazione che i programmi realizzati. La CE-125 contiene armoniosamente il PC-1245 mantenendo le dimensioni di un libro.

Direttore dell'opera
GIANNI DEGLI ANTONI

Comitato Scientifico
GIANNI DEGLI ANTONI
Docente di Teoria dell'Informazione, Direttore dell'Istituto di Ciber 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 Calc 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 del Calcolatori all'Università Autonoma Metropolitana di Città del Messico), GOFFREDO HAUS, MARCO MAIOCCHI, DANIELE MARINI, GIANCARLO MAURI, CLAUDIO PARRELLI, ENNIO PROVERA

Testi
Eidos (TIZIANO BRUGNETTI), CLAUDIO PARRELLI, ENNIO PROVERA, VIRGINIO SALA, Etnoteam (ADRIANA BICEGO)

Tavole
Logical Studio Communication
Il Corso di Programmazione a 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 - Milano

Direttore Editoriale
ORSOLA FENGLI

Coordinatore settore scientifico
UGO SCAIONI

Redazione
MARINA GIORGETTI
LOGICAL STUDIO COMMUNICATION

Art Director
CESARE BARONI

Impaginazione
BRUNO DE CECCHI
PAOLA ROZZA

Programmazione Editoriale
ROSANNA ZERBARINI
GIOVANNA BREGGÈ

Segretaria di Redazione
RENATA FRIGOLI
LUCIA MONTANARI

**NEL PROSSIMO NUMERO
IN OMAGGIO
IL TERZO POSTER
"LA STORIA
DELL'INFORMATICA"**

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, S.p.A., Milano - Prima Edizione 1984 - Direttore responsabile GIOVANNINI - Registrazione presso il Tribunale di Milano n. 135 del marzo 1984 - Iscrizione al Registro Nazionale della Stampa n. 00282, 3, Foglio 489 del 20.9.1982 - Stampato presso lo Stabilimento Grafico Gruppo Editoriale Fabbri S.p.A., Milano - Diffusione Gruppo Editoriale Fabbri S.p.A. via Mecenate, 91 - tel. 50951 - Milano - Distribuzione per l'Italia: A. & G. Marco s.a.s., via Fortezza 27 - tel. 2526 - Milano - Pubblica periodica settimanale - Anno I - n. 8 - esce il giovedì - Spedizione in abb. postale - Gruppo II/70 L'Editore si riserva la facoltà di modificare il prezzo nel corso della pubblicazione, se costretto da mutate condizioni di mercato.

concessionaria
per l'Italia

MELCHIONI

**TUTTA LA POTENZA DI UN COMPUTER
NEL PALMO DELLA TUA MANO**

Per ulteriori informazioni, scrivete a
MELCHIONI - Divisione Pocket Computer - 20135 MILANO - Via P. Colletta, 37

SHARP

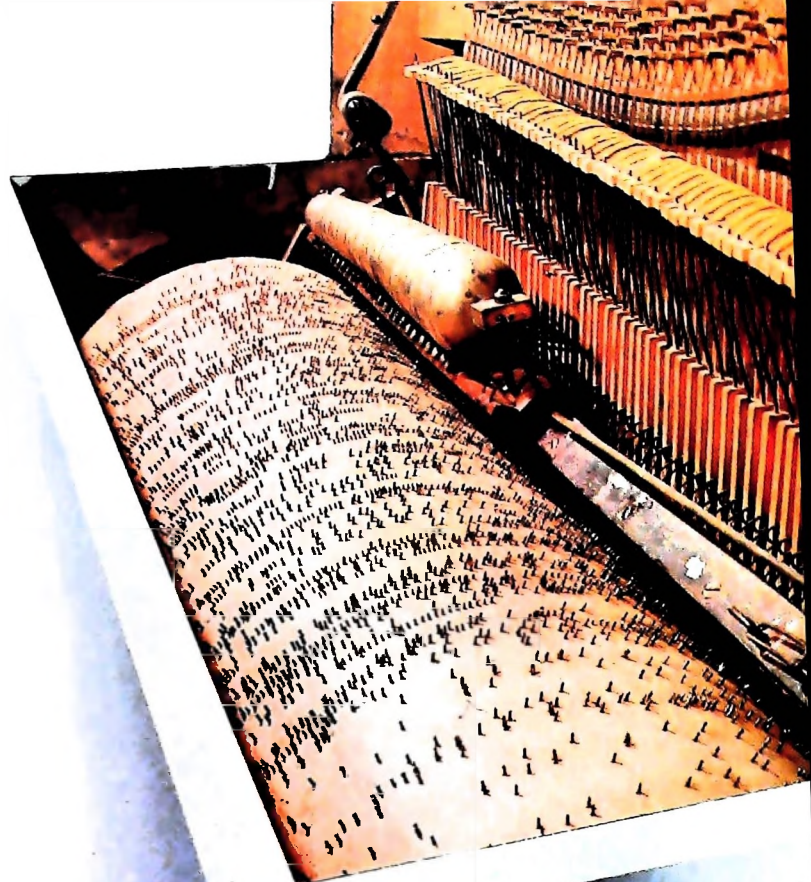
LE CONVERSIONI

I metodi per passare dalla rappresentazione di un numero in un sistema alla sua rappresentazione in un altro.

Qualunque informazione può essere rappresentata da un numero: il modo più semplice per ottenere tale risultato può essere quello di rappresentare ogni lettera dell'alfabeto e ogni simbolo che si presenti nella comunicazione con un numero, in modo che a numeri diversi corrispondano caratteri diversi (e viceversa). Possiamo assegnare 1 alla lettera A, 2 alla B, 3 alla C, e via di seguito: una sequenza di numeri a quel punto corrisponderà a un termine, a una espressione o a una frase. D'altra parte, qualunque numero decimale può essere espresso in forma binaria: quindi, qualunque informazione può essere rappresentata da un numero binario.

È questo il principio su cui si basano gli elaboratori elettronici attuali: bastano componenti fondamentalmente molto semplici, in grado di assumere due soli stati, per rappresentare qualunque informazione e per eseguire qualunque operazione su di essa. Questo significa però che alla macchina ogni informazione e ogni comando devono essere forniti sotto forma di una sequenza di 0 e 1, cioè sotto forma di un numero binario.

0	0							
1	1							
2	10							
3	11							
4	100							
5	101							
6	110							
7	111							
8	1000							
9	1001							



La rappresentazione delle informazioni in forma binaria è molto più antica del computer. Già nel secolo scorso esistevano strumenti musicali in grado di leggere ed eseguire automaticamente un brano scritto con una serie di fori su una striscia di carta. I vecchi organetti meccanici che fino a pochi anni fa capitava di incontrare per le strade funzionavano proprio in questo modo. Un foglio di cartone robusto veniva esaminato da un dispositivo che premava il tasto corrispondente ad una certa nota quando incontrava un foro in una certa posizione. In questo particolare codice binario i due stati opposti sono "buco", a cui corrisponde il suono, e "non buco" a cui corrisponde il silenzio.

Conversione da binario a decimale

1 0 1 1 0 1₂

$$\frac{1 \times 2^5}{32} + \frac{0 \times 2^4}{+} + \frac{1 \times 2^3}{8} + \frac{1 \times 2^2}{4} + \frac{0 \times 2^1}{+} + \frac{1 \times 2^0}{1}$$

= 45₁₀

Conversione da ottale a decimale

5 3 6₈

$$\frac{5 \times 8^2}{320} + \frac{3 \times 8^1}{24} + \frac{6}{6}$$

= 350₁₀

Conversione da esadecimale a decimale

2 B 7₁₆

$$\frac{2 \times 16^2}{512} + \frac{11 \times 16}{176} + \frac{7 \times 16^0}{7}$$

= 695₁₀

Conversione da binario a ottale

1 0 1 1 1 0 1 0 1 1 0₂

2 7 2 6₈

Conversione da binario a esadecimale

1 0 1 1 1 0 1 0 1 1 1₂

5 D 7₁₆

Conversione da ottale a binario

3 5 1 4₈

011 101 001 100

= 0 1 1 1 0 1 0 0 1 1 0 0₂

Conversione da esadecimale a binario

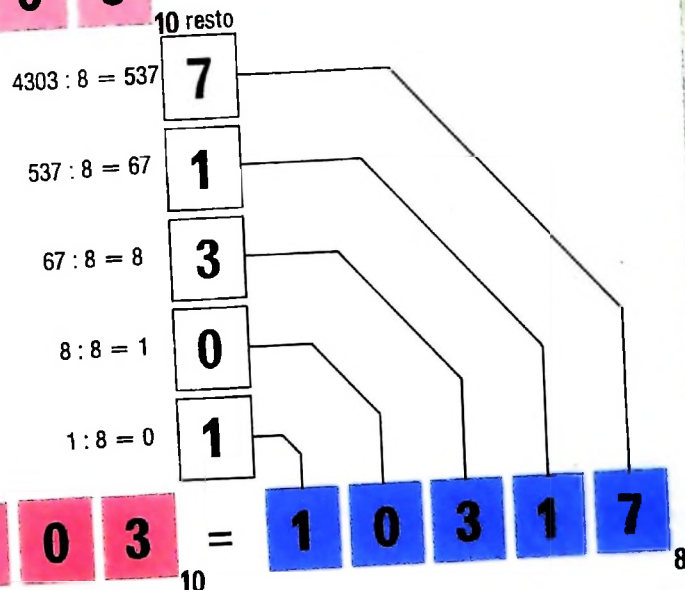
6 A F 3₁₆

0110 1010 1111 0010

= 0 1 1 0 1 0 1 0 1 1 1 1 0 0 1 0₂

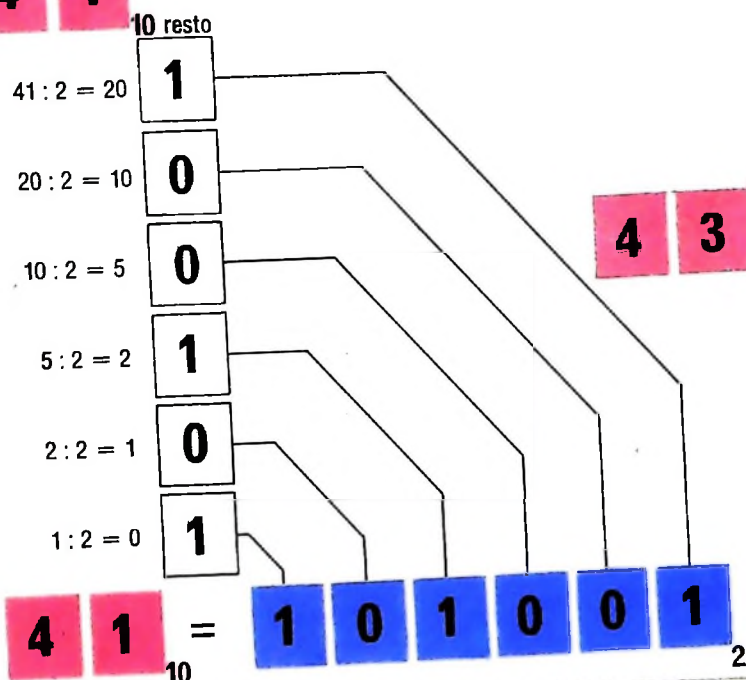
Conversione da decimale a ottale

4 3 0 3



Conversione da decimale a binario

4 1



Ma il sistema binario, per quanto semplice nella sua concezione, è poco maneggevole per un uomo: per questo in genere preferiamo preparare le informazioni o i comandi in una forma che ci sia più immediatamente comprensibile e trasformarli poi solo all'ultimo momento nella forma adatta per la macchina. Così, per esempio, è più semplice eseguire un calcolo nel sistema decimale e trasformare poi il risultato nella sua rappresentazione binaria, piuttosto che eseguire quello stesso calcolo, nel sistema binario, sulla rappresentazione binaria dei numeri di partenza. Il sistema esadecimale, è più complesso da usare, ma facile da convertire in binario: a ogni cifra esadecimale corrisponde biunivocamente un numero di quattro cifre nel sistema binario. Per introdurre dati in "codice di macchina" il sistema esadecimale rappresenta una eccellente scorciatoia: per questo sugli elaboratori programmabili esclusivamente in codice di macchina si trovano spesso tastierini esadecimali (è facile realizzare circuiti che a ogni tasto "esadecimale" facciano corrispondere la corretta sequenza di impulsi 0 e 1 in binario).

Per potersi muovere agevolmente fra un sistema di rappresentazione e l'altro, tuttavia, è necessario conoscere i metodi per passare dalla rappresentazione di un numero in un sistema alla sua rappresentazione in un altro sistema: nel prossimo paragrafo passeremo in rassegna questi metodi, esemplificandoli nelle illustrazioni.

Conversioni fra sistemi di numerazione

Il passaggio dal sistema binario al decimale si effettua sfruttando le potenze di due: per trasformare 101 si calcola semplicemente e direttamente $1 \times 2^2 + 1 \times 2^0$.

La conversione inversa, da decimale a binario, è meno immediata. Il modo più semplice per realizzarla è quello di effettuare una serie di divisioni per due del numero di partenza: la successione dei resti (che possono essere solo 0 e 1), presa in ordine inverso, ci dà le cifre del corrispondente numero binario. Per esempio, se il numero da convertire è 1473, abbiamo $1473 : 2 = 736$ (con resto 1); $736 : 2 = 368$ (resto 0); $368 : 2 = 184$ (resto 0); $184 : 2 = 92$ (resto 0); $92 : 2 = 46$ (resto 0); $46 : 2 = 23$ (resto 0); $23 : 2 = 11$ (resto 1); $11 : 2 = 5$ (resto 1); $5 : 2 = 2$ (resto 1); $2 : 2 = 1$ (resto 0); $1 : 2 = 0$ (resto 1). Il numero binario corrispondente è dunque 1011100001.

Le conversioni da ottale e da esadecimale in decimale seguono lo stesso procedimento usato per la conversione da binario a decimale: si sviluppa il numero di partenza sfruttando le potenze di 8 e di 16, rispettivamente. Analogamente, si risolve anche la conversione da decimale a ottale e a esadecimale: si divide il numero decimale successivamente per 8 o per 16, e si considerano i resti delle divisioni successive, in ordine inverso, come per la conversione da decimale a bina-

Sottrazione e complemento

Nei sistemi binario, ottale ed esadecimale, la sottrazione può essere effettuata in modo diretto, come siamo abituati a fare nel sistema decimale, ma spesso si ricorre per comodità al metodo del "complemento" (possibile anche in decimale, ma raramente usato). In binario, con il metodo diretto, le differenze fondamentali sono: $0 - 0 = 0$; $1 - 0 = 1$; $0 - 1 = \&$ (con il "prestito" di 1 dalla cifra immediatamente a sinistra). Un esempio:

$$\begin{array}{r} 10001 \\ - 01101 \\ \hline 00100 \end{array}$$

(Lo si può verificare trasformando tutto nel sistema decimale.)

Il metodo del complemento trasforma la differenza in una somma: $A - B$ diventa la somma di A e del complemento di B . (I due numeri devono avere lo stesso numero di cifre: basta aggiungere a sinistra tanti zeri quanti sono necessari, al numero con minor numero di cifre, eventualmente.) Il complemento di un numero in una data base b si ottiene calcolando per ogni cifra a del numero la differenza fra $b - 1$ e a , fatta eccezione per l'ultima cifra per cui si calcola la differenza fra b e a . Un esempio nel sistema decimale: il complemen-

to di 134 è 866 ($9 - 1 = 8$, $9 - 3 = 6$, $10 - 4 = 6$). Nel sistema binario, il calcolo del complemento consiste semplicemente nel trasformare ogni 1 in 0 e viceversa, fatta eccezione per l'ultima cifra, che resta inalterata. Quando poi si somma A al complemento di B , si deve trascurare l'ultimo riporto a sinistra. Calcoliamo, per esempio, in binario, $100101 - 011001$. Trasformiamo innanzitutto 011001 nel suo complemento, che è 100111 . Poi sommiamo:

$$\begin{array}{r} 100101 \\ + 100111 \\ \hline 001100 \end{array}$$

e trascuriamo l'ultimo riporto a sinistra.

Il procedimento è analogo per i sistemi ottale ed esadecimale (con il complemento alla relativa base).

Perché questo metodo funziona? Consideriamo il caso del sistema decimale. Data una sottrazione $A - B$, il complemento del numero B a n cifre è come calcolare $10^{n+1} - B$; quello che si calcola poi è $A + 10^{n+1} - B$, ovvero $A - B + 10^{n+1}$. Questo giustifica l'operazione e spiega perché si debba trascurare l'ultimo riporto a sinistra (che corrisponde a 10^{n+1}).

rio. Le conversioni fra i sistemi binario, ottale ed esadecimale sono più semplici: si sfrutta il fatto che 8 e 16 sono potenze di 2. Le otto cifre del sistema ottale possono essere rappresentate con otto numeri di tre cifre nel sistema binario; le sedici cifre del sistema esadecimale sono rappresentate da sedici numeri di quattro cifre nel sistema binario, come possiamo vedere negli esempi delle pagine precedenti. Per passare da una rappresentazione ottale a una rappresentazione nel

sistema binario, basta sostituire ciascuna cifra ottale con la corrispondente terna di cifre binarie; e analogamente a ogni cifra di una rappresentazione esadecimale si può sostituire la corrispondente quadrupla di cifre binarie per ottenere l'equivalente rappresentazione binaria.

Le conversioni inverse si ottengono suddividendo la rappresentazione binaria del numero in terne o in quadruple e sostituendo ad esse le corrispondenti cifre ottali o esadecimali.

Glossario

Bug - nel gergo degli informatici, un errore, in genere con riferimento a un programma; in inglese il termine è usato a volte anche per i guasti dell'hardware. La parola significa letteralmente "pulce".

Carattere - qualunque tipo di simbolo (cifre, lettere dell'alfabeto, segni di interpunzione, lo spazio vuoto, speciali simboli grafici ecc.) accettato in ingresso dal calcolatore.

Debugging - nel gergo, togliere i "bug", cioè gli errori da un programma. È una delle fasi più importanti e faticose nella realizzazione del software. L'espressione italiana più usata con lo stesso significato è "messa a punto".

Diagramma di flusso (flowchart) - un particolare tipo di diagramma, utilizzato per visualizzare una procedura, un algoritmo, i cui elementi fondamentali sono caselle di varia forma collegate da frecce. La forma della casella indica il tipo dell'operazione che si svolge a quel passo della procedura: un'operazione di ingresso, una decisione, e via di seguito. Le frecce indicano i collegamenti tra i vari passi della procedura. I diagrammi di flusso sono utilizzati nelle prime fasi della programmazione per evidenziare i singoli passi del procedimento di soluzione di un problema, prima di codificarli in un particolare linguaggio di programmazione.

Hardware - letteralmente significa "ferramenta" e indica in genera-

le i componenti fisici di un calcolatore: circuiti, cavi, tasti, manopole, e via dicendo. È usato in contrapposizione a "software".

Nuclei, memoria a - un tipo di memoria principale largamente usato nei calcolatori fino a pochi anni fa. I nuclei erano anelli di ferrite, del diametro di circa 1 millimetro, e sono stati soppiantati dai dispositivi microelettronici. In inglese il termine "core memory", "memoria a nuclei" o "memoria nucleare" è usato ancora spesso per indicare genericamente la memoria principale di una macchina, anche quando è realizzata con tecnologie diverse.

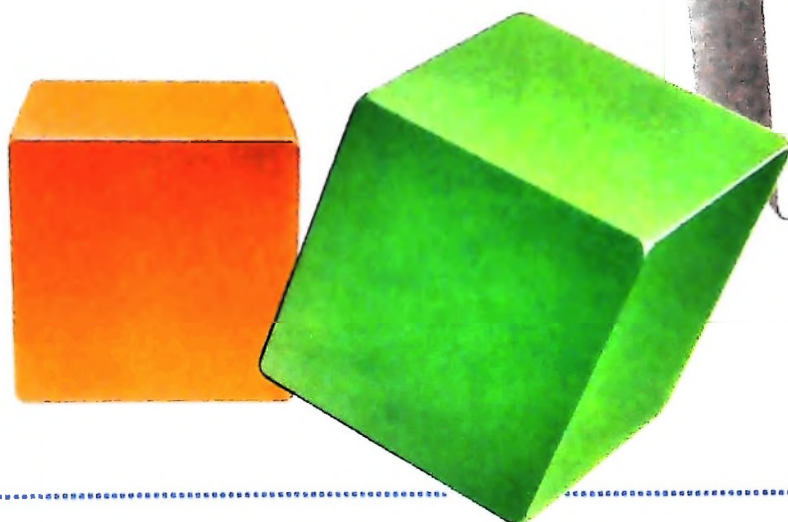
Scheda perforata - una scheda speciale in cartoncino in cui vengono praticati dei fori, per il controllo automatico di apparecchiature. Le schede perforate furono introdotte da Jacquard nell'Ottocento per il controllo dei telai da tessitura (i fori consentivano l'abbassamento degli aghi), poi furono adottate da Hollerith nelle sue macchine elaboratrici (progettate per l'elaborazione dei dati del censimento della popolazione degli Stati Uniti del 1900). A lungo le schede perforate hanno rappresentato il mezzo più economico per l'inserimento di dati nei calcolatori.

Software - termine generico, coniato sul calco di "hardware" ("hard" significa "duro", "soft" significa "molle, morbido") per indicare i programmi di un calcolatore (indipendentemente dal supporto fisico).

MASTER-MIND

Un gioco tradizionale, in una versione con il computer come avversario.

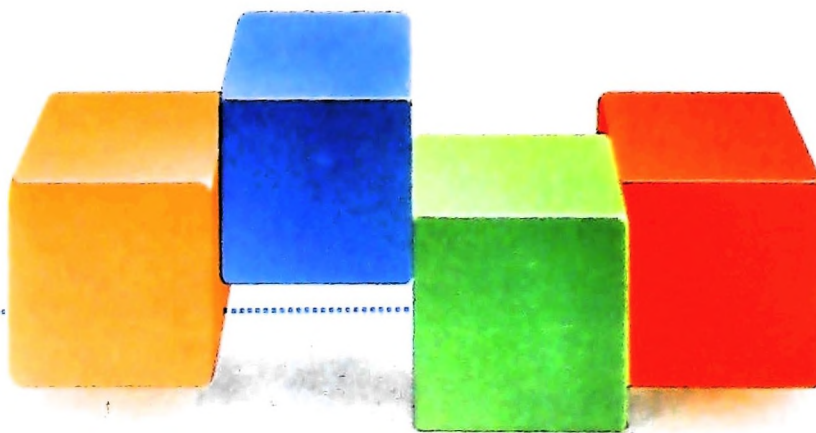
Il MASTER-MIND è un gioco tradizionale, che sicuramente vi sarà noto in almeno una delle sue tante forme. Ridotto all'osso, è un gioco a due, in cui ogni giocatore imposta una propria "parola segreta" traendo gli elementi componenti da un repertorio prefissato e noto. L'avversario deve indovinare tale parola segreta, compiendo una serie di tentativi e sfruttando le informazioni che il giocatore deve fornirgli di volta in volta, e che riguardano il numero degli elementi componenti indovinati e il numero di quelli indovinati che sono sta-



Il MASTER-MIND è un passatempo famosissimo di cui esistono infinite versioni con regole spesso diverse. In questo caso si usa un codice di 4 colori scelti in un repertorio di 8. Sono ammesse le ripetizioni, quindi ci sono in tutto $8^4 = 4096$ possibilità.

Codice

A B U R



ti individuati nella posizione corretta. Queste informazioni non comprendono tuttavia l'indicazione di quali siano gli elementi effettivamente indovinati, ma solo il loro numero. Il gioco può essere svolto usando come repertorio da cui "pescare" una serie di colori, delle lettere, dei numeri; talune versioni ammettono la ripetizione degli elementi nella parola segreta, altre la bandiscono. Il gioco, comunque, nelle sue linee essenziali, è molto semplice: non è facile però escogitare condotte di gioco molto efficienti, in particolare se la parola segreta è di una certa lunghezza e se sono ammesse le ripetizioni.

Il programma che presentiamo trasforma il computer in un tranquillo avversario, che si limita al ruolo del codificatore: M10, cioè, imposta il proprio codice segreto e attende poi con serafica pazienza che il giocatore umano (entro un limitato numero di tentativi) riesca a svelare il suo... segreto.

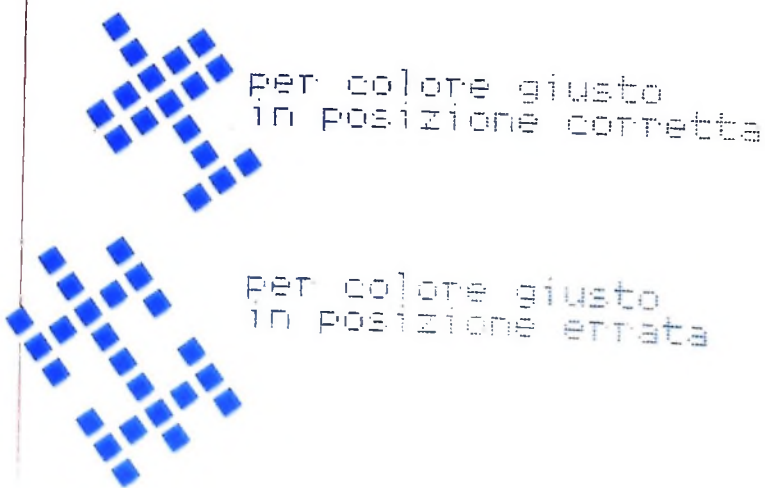
Il gioco

Si presenta in due versioni, MASTER-MIND (relativa alla scelta 1 del menù) che viene esposta in queste pagine e quella denominata SUPER MASTER-MIND (relativa alla scelta 2) che verrà presentata in seguito, e che è sviluppata in modo tale da poter essere inserita nel primo programma senza alcun problema.

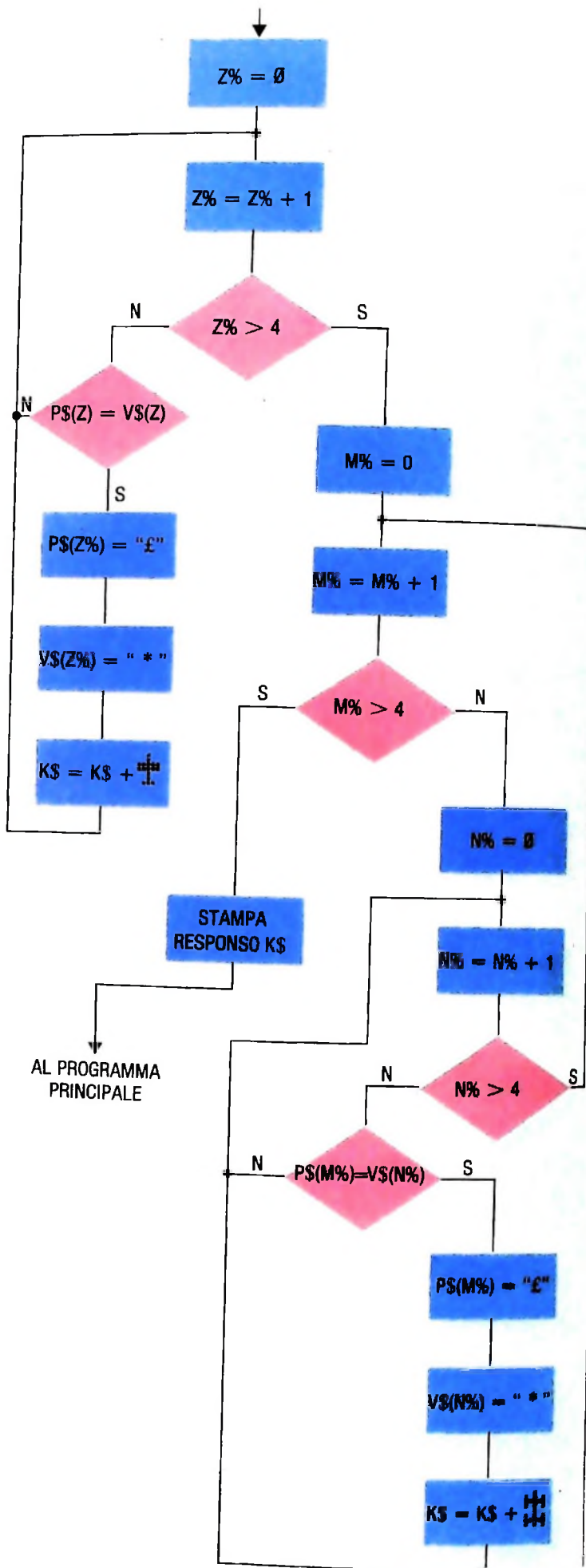
Nella versione MASTER-MIND, M10 dispone di 8 colori e precisamente: Arancione, Blu, Giallo, Verde, Rosso, Marrone, Nero, White (si è utilizzato il termine inglese per "bianco" evitando così che l'iniziale si potesse confondere con il Blu); deve sceglierne quattro, non necessariamente diversi, per formare il suo codice segreto. Avrà quindi 4096 combinazioni a sua disposizione.

Il decodificatore può contare su 10 tentativi per cercare di scoprire la sequenza preorganizzata; tali tentativi verranno inseriti digitando una di seguito all'altra le quattro iniziali dei colori scelti nel rettangolo CODICE.

Supponiamo ad esempio di aver scelto Rosso, Blu, Verde, Giallo: dovremo quindi digitare RBVG (e poi premere ENTER). M10 analizzerà la "proposta" e fornirà il responso per mezzo di due simboli grafici, e precisamente:



DAL PROGRAMMA PRINCIPALE



per poi riposizionare il cursore nel rettangolo CODICE e attendere la scelta successiva.

Al termine dei 10 tentativi, se non si fosse riusciti a svelare l'enigma, la pressione di un qualsiasi tasto permetterà la visualizzazione del codice segreto.

Il programma

L'elaborazione comincia con il caricamento in memoria (linee 10-30) degli otto colori registrati nella linea 60 e quindi visualizzerà (linea 40) il menù delle opzioni richiamabili. La scelta dell'opzione 1 ci invierà alla linea 6000 e quindi, dopo aver deciso se prendere visione delle istruzioni, alle linee 6030-6100 ove risiede la funzione di selezione del codice segreto, che verrà collocato nella variabile stringa C\$. (L'analisi dell'algoritmo di questa funzione verrà svolta quando introdurremo il SUPER MASTER-MIND.)

Le linee 6140-6380 contengono la routine di inserimento e controllo, che per prima cosa provvede ad assegnare ogni singolo carattere della stringa C\$ (il codice) al vettore V\$ di indice Z%. Si può notare come il % venga utilizzato per defi-

nire una variabile intera: si ottiene così una minore occupazione della memoria.

Le linee 6160-6180 permettono la visualizzazione del rettangolo con la scritta CODICE: a questo punto M10 rimane in attesa del nostro input, che verrà successivamente memorizzato nella stringa T\$ a sua volta elaborata e suddivisa, in ogni singolo elemento, nel vettore P\$ (S%) e visualizzato, in forma più ordinata, sul display grazie alla linea 6220. A questo punto il controllo; nelle linee 6260-6340 la funzione si evolve secondo l'algoritmo riportato nell'illustrazione in forma di diagramma di flusso.

La sostituzione degli elementi riscontrati uguali con i caratteri * e £ si rende necessaria per evitare i doppi controlli e quindi un responso errato.

Dopo aver visualizzato il responso K\$, la linea 6360 ne controllerà l'uguaglianza con la sequenza di quattro simboli di aeroplano, nel qual caso la linea 8010 segnalerà la riuscita del gioco, in caso contrario si proseguirà con il successivo tentativo. Nell'ipotesi si siano esaurite tutte le dieci possibilità a nostra disposizione, la linea 6410 esprimerà il "rammarico" del computer visualizzando, oltre alla soluzione, la statistica relativa alla media dei tentativi.

```

' *****
' *                                     MASTER - MIND                                     *
' *                                     *                                               *
' *   release 1.1           del 10/03/84                                           by C.V.P.   *
' *****

```

```

10 FOR J%=1 TO 8
20   READ A$(J%)
30 NEXT J%
35 FOR J% = 1 TO 6
36   READ AA$(J%),BB$(J%)
37 NEXT J%
40 GOSUB 5000
50 ON I% GOTO 6000,7000
60 DATA "A","R","V","B","M","G","N","W"
70 DATA "A","1","R","2","V","3","B","4"
80 DATA "M","5","G","6"
4990 ' *****
4992 '*      ROUTINE MENU' *
4995 ' *****
5000 CLS
      : F=0
      : AV=0
      : AVE=0
5010 LINE(5,5)-(234,58) ,1,B F
5020 PRINT$92,"1) MASTER-MIND"
5030 PRINT$172,"2) SUPER MASTER-MIND"
5040 PRINT$264,"Seleziona:";
      :INPUT I%
5050 IF I%>2 OR I%<1 THEN 5040
5060 RETURN
6000 CLS
6010 PRINT$8,"Vuoi le istruzioni? (s/n);
6015 B$ = INKEY$
6017 IF B$="" THEN 6015
6020 IF B$="s" THEN GOSUB 15000
6025 CLS
      :C$=""
      : K$=""
6030 FOR X%=1 TO 4
6040   Y% = VAL(RIGHT$(TIME$,2))
6050   IF Y%=0 THEN 6040
6060 FOR K%=1 TO Y%
6070   D% = (INT(RND(1)*8))+1
6080 NEXT K%
6090   C$ = C$ + A$(D%)
6100 NEXT X%
6110 ' *****
6120 '*      ROUTINE INSERIMENTO *
6130 ' *****
6140 L = 0
6150 FOR I=1 TO 10
6151 FOR Z% = 1 TO 4
6152 V$(Z%) = MID$(C$,Z%,1)
6153 NEXT Z%
6155   IF I=8 THEN L=20

```

```

6157 IF I=10 THEN L=99
6160 LINE(166,60)-(238,30) ,1,B
6170 PRINT$191,"Codice"
6180 PRINT$270," ";
      :INPUT T$
6190 FOR S%=1 TO 4
6200 P$(S%) = MID$(T$,S%,1)
6210 NEXT S%
6220 PRINT$L,I;" ";P$(1);" ";P$(2)
      );" ";P$(3);" ";P$(4); ●
6230 '*****
6240 '* CONTROLLO *
6250 '*****
6260 FOR Z%=1 TO 4
6270 IF P$(Z%) = V$(Z%) THEN K$ = K$ + C
      HR$(200) ●
      : V$(Z%) = "*"
      : P$(Z%) = "£"
6280 NEXT Z%
6290 FOR M%=1 TO 4
6300 FOR N%=1 TO 4
6310 IF M%=N% THEN 6330
6320 IF P$(M%) = V$(N%) THEN K$ = K
      $ + CHR$(201) ●
      : V$(N%) = "*"
      : P$(M%) = "£"
6330 NEXT N%
6340 NEXT M%
6350 PRINT" ";K$
6360 IF K$ =CHR$(200)+CHR$(200)+CHR$(20
      0)+CHR$(200) THEN GOSUB 8000 ●
      : GOTO 6025
6370 L = L + 40
      : K$ = ""
6380 NEXT I
6390 AB$ = INKEY$
      : IF AB$ = "" THEN 6390
6400 CLS
6410 PRINT "Mi dispiace, ma l'esatta so
      luzione è:" ●
6420 PRINT$90,C$
6430 GOSUB 9000
6440 GOTO 6025
8000 FOR H = 1 TO 600
      : NEXT H
      :CLS
8010 PRINT" COMPLIMENTI !! ci sei rius
      cito" ●
8020 GOSUB 9000
8030 RETURN
9000 F = F + 1
9010 LINE(67,31)-(197,60) ,1,B
9020 PRINT$173,"Tentativi Media"
9030 AV = AV + I
9040 AVE = AV/F
9050 PRINT$256,I;" " ;AVE

```

```

9060 BB$ = INKEY$
      : IF BB$ = "" THEN 9060
9070 CLS
9080 PRINT$122,"UN'ALTRA PARTITA?"
9090 AA$=INKEY$
      : IF AA$ = "" THEN 9090
9100 IF AA$="s" OR AA$="S" THEN RETURN
9120 GOTO 5000
15000 CLS
      : PRINT$10,"** MASTER-MIND **"
15010 PRINT"SALVE! Avete 10 tentat
      ivi"; ●
15011 PRINT" per scoprire il CODICE che h
      o scelto per voi." ●
15015 PRINTCHR$(10)
      : PRINT"Ci sono 8 colori a vostra di
      sposizione:"
15020 GOSUB 18020
15040 CLS
      : PRINT"R-osso V-verde B-l
      u G-giallo" ●
15050 PRINT"A-rancione M-arrone N-e
      ro W-hite" ●
15060 PRINT
      : PRINT"Per impostare la scelta è s
      ufficiente" ●
15065 PRINT" scrivere le 4 iniziali nel"
15070 PRINT"rettangolo CODICE e preme
      re |ENTER ●
15080 GOSUB 18020
15100 CLS
      : PRINT" M10 risponderà con ";
      :PRINTCHR$(200);
      :PRINT" se il colore è esatto e ne
      lla giusta posizione," ●
15110 PRINT"mentre risponderà con ";
      :PRINTCHR$(201);
      :PRINT" se il colore è esatto ma ne
      lla posizione errata." ●
15120 PRINT
      : PRINT"PREMI UN TASTO e.... BUON
      A FORTUNA !!" ●
15130 C9$ = INKEY$
      : IF C9$ = "" THEN 15130 ELSE RETURN
18020 PRINT" PREMI UN TAST
      O" ●
18030 C9$ = INKEY$
      : IF C9$ = "" THEN 18030
18040 RETURN

```

N.B. Il ● sta a significare che la linea va a capo per esigenze editoriali, quindi nell'eseguirlo non interrompere la digitazione

*Lezione 7***Le iterazioni enumerative**

Le strutture di iterazione fin qui mostrate sono molto "potenti" dal punto di vista della capacità di calcolo: infatti permettono di risolvere problemi in cui il numero di ripetizioni necessarie non è noto a priori, ma dipende da specifici eventi legati a ciò che avviene durante l'esecuzione.

Tuttavia, abbiamo usato tali strutture anche per effettuare iterazioni di porzioni di programma per un numero di volte noto a priori, che abbiamo scandito con una variabile usata come contatore che assumesse tutti i valori da 1 al numero totale di iterazioni.

Per soddisfare questa esigenza, molto comune nella pratica di programmazione, la maggior parte dei linguaggi di programmazione mette a disposizione strutture di controllo specifiche per questo tipo di iterazione, dette ITERAZIONI ENUMERATIVE.

L'esempio di visualizzazione dei primi 10 numeri già presentato può essere riscritto nel modo seguente:

- Facendo variare I da 1 a 10 esegui:
visualizza I

Come al solito, useremo la notazione del linguaggio Pascal:

```
FOR I: = 1 TO 10 DO
  visualizza I
```

La struttura FOR indicata effettua le seguenti operazioni:

- assegna 1 alla variabile I (si noti il simbolo "==" che in Pascal è il simbolo di assegnamento)
- controlla che il valore del contatore I non sia maggiore di 10; se I è maggiore di 10 termina l'iterazione proseguendo con la parte di programma che eventualmente segue
- visualizza I
- incrementa I di 1
- torna a verificare il valore di I.

La struttura For in Basic

In BASIC è disponibile una struttura FOR che ha un comportamento leggermente diverso.

Essa, infatti, effettua il controllo sul valore del contatore dell'iterazione solo alla

Completata questa settima lezione del Corso di Programmazione e BASIC, siete in grado di eseguire gli esercizi TAVOT.DO TAVOP.BA PITAT.DO PITAT.BA contenuti nella cassetta "11 Esercizi di Programmazione". I titoli seguiti dal suffisso DO corrispondono a testi, quelli seguiti da BA a programmi in BASIC. Caricateli secondo le modalità che avete appreso.

fine dell'iterazione stessa, secondo il modello seguente:

```
I:= 1
REPEAT
  visualizza I
  I:I+1
UNTIL I>10
```

Vediamo adesso la struttura FOR messa a disposizione dal BASIC:

```
10 FOR I=1 TO 10
20 PRINT I
30 NEXT I
```

che "alla lettera" può essere spiegata così:

```
variando I da 1 a 10
visualizza I
tratta il prossimo I
```

Vediamo un altro esempio ancora:

```
10 FOR I=10 TO 1 STEP -1
20 PRINT I
30 NEXT I
```

In questo caso la parola chiave STEP specifica il "passo" di cui incrementare il valore iniziale della variabile I per raggiungere il valore finale.

In questo caso, avendo il passo il valore negativo -1, la variabile I assumerà i valori 10, 9, 8... 3, 2, 1.

Si osservi ancora il seguente esempio:

N.B. Il ● sta a significare che la linea va a capo per esigenze editoriali, quindi nell'eseguirlo non interrompere la digitazione.

```
10 REM Tabella di conversione di misure di angoli ●
20 REM da radianti a gradi
30 REM Nota: 360 gradi=2*PI GRECO radianti
40 LET P=3.14159
45 PRINT "RADIANTI", "GRADI"
50 FOR I=0 TO 2*P STEP .5
60 PRINT I, I/(2*P)*360
70 NEXT I
```

Notiamo che il valore iniziale, il valore finale e il passo della variabile I possono assumere valori qualunque, indicati anche come espressioni.

La variabile che scandisce l'iterazione (nell'esempio la variabile I) prende il nome di **INDICE DELL'ITERAZIONE**.

Studiamo meglio il comportamento dell'iterazione quando il passo è diverso da 1.
Il programma:

```
10 INPUT S
20 FOR K=1 TO 10 STEP S
30 PRINT K
40 NEXT K
```

equivale al primo mostrato qualora si fornisca 1 come valore del passo. Se invece si fornisce il valore 2 verranno visualizzati i seguenti valori: 1, 3, 5, 7, 9. La variabile I, incrementata all'ultima passata fino ad assumere il valore 11, interromperà l'iterazione, poiché il valore è superiore a 10: se pertanto visualizziamo il valore di I alla fine dell'iterazione, ci apparirà il valore 11.

Ancora, se forniamo il valore del passo uguale a -1 , l'iterazione verrà eseguita una sola volta e il valore finale sarà uguale a 0. Infatti, esaminando la realizzazione di un FOR...NEXT con il modello REPEAT...UNTIL, osserviamo che:

- l'iterazione viene comunque eseguita una volta e quindi il valore iniziale di I viene visualizzato;
- quindi il valore dell'indice viene incrementato dal passo e l'iterazione viene interrotta in quanto il valore finale è chiaramente non raggiungibile a partire dal valore iniziale con il passo indicato.

Un esempio di iterazione enumerativa

Immaginiamo di creare un programma per la costruzione della classifica di una gara di atletica a cui partecipano gli alunni di una scuola: nella gara si fronteggiano le differenti classi della scuola, in modo che il punteggio che viene ottenuto da ogni alunno contribuisca a determinare il punteggio della rispettiva classe; quest'ultimo viene calcolato come media dei punteggi conseguiti dagli alunni appartenenti a quella classe.

Il programma che fornisce la classifica sarà fatto così:

chiede quante sono le classi
variando I da 1 al numero di classi
chiede quanti allievi ha la classe I-esima
azzera la somma dei punteggi della classe I-esima
variando k da 1 al numero di allievi della classe I-esima
chiede il punteggio dell'allievo K-esimo
somma tale punteggio al punteggio totale della classe
calcola il punteggio medio della classe I-esima e lo visualizza

Possiamo dunque effettuare (si veda alla pagina successiva) la realizzazione in BASIC di tale programma:

```

10 PRINT "CLASSIFICA DELLA GARA DI COR
SA AGLI OSTACOLI"
20 INPUT "Quante classi partecipano";C
30 FOR I=1 TO C
40 REM Chiede in A il numero di allievi
50 REM Calcola in S la somma dei punteggi
100 PRINT "PUNTEGGIO DELLA CLASSE: ";S/A
110 NEXT I

```

A questo punto, possiamo occuparci della somma dei punteggi degli allievi di ogni classe.

```

40 LET S=0
50 PRINT "Quanti allievi ha la classe ";I;
55 INPUT A
60 FOR K=1 TO A
70 PRINT "Punteggio allievo ";K;
75 INPUT P
80 LET S=S+P
90 NEXT K

```

Si noti che il programma effettua due iterazioni: una sulle classi della scuola e una sugli allievi della stessa classe.

Si noti in particolare che per realizzare l'iterazione di richieste di punteggio di ciascuno degli alunni della stessa classe è stato necessario inserire l'iterazione sugli alunni all'interno di quella sulle classi: abbiamo cioè innestato le due strutture iterative.

Cosa abbiamo imparato

In questa lezione abbiamo appreso:

- Il concetto di ITERAZIONE ENUMERATIVA
- La realizzazione Pascal di iterazioni enumerative
- La struttura FOR...NEXT del BASIC
- L'innestamento di strutture iterative.

STRUMENTI PER INSEGNARE

Oltre ad avere un valore formativo in se stessa, l'informatica si offre come potente strumento per la didattica.

L'elaboratore ha superato ormai da molti anni la fase di semplice strumento di calcolo: le sue prestazioni sono varie e complesse e quindi si adattano bene anche ai diversi settori della didattica.

Cominciando dalle apparecchiature (ma ritorneremo più avanti su questo argomento), il video può essere utilizzato per grafici di qualunque tipo: per rappresentare funzioni matematiche, per costruzioni geometriche, per visualizzare simulazioni di sistemi fisici o chimici. Attualmente esistono anche schermi sensibili al tocco del dito, il che semplifica notevolmente il rapporto uomo-macchina.

I plotter possono rappresentare le immagini dello schermo televisivo e quindi fornire grafici di grande raffinatezza a più colori di qualunque fenomeno studiato dal sistema. E, anco-

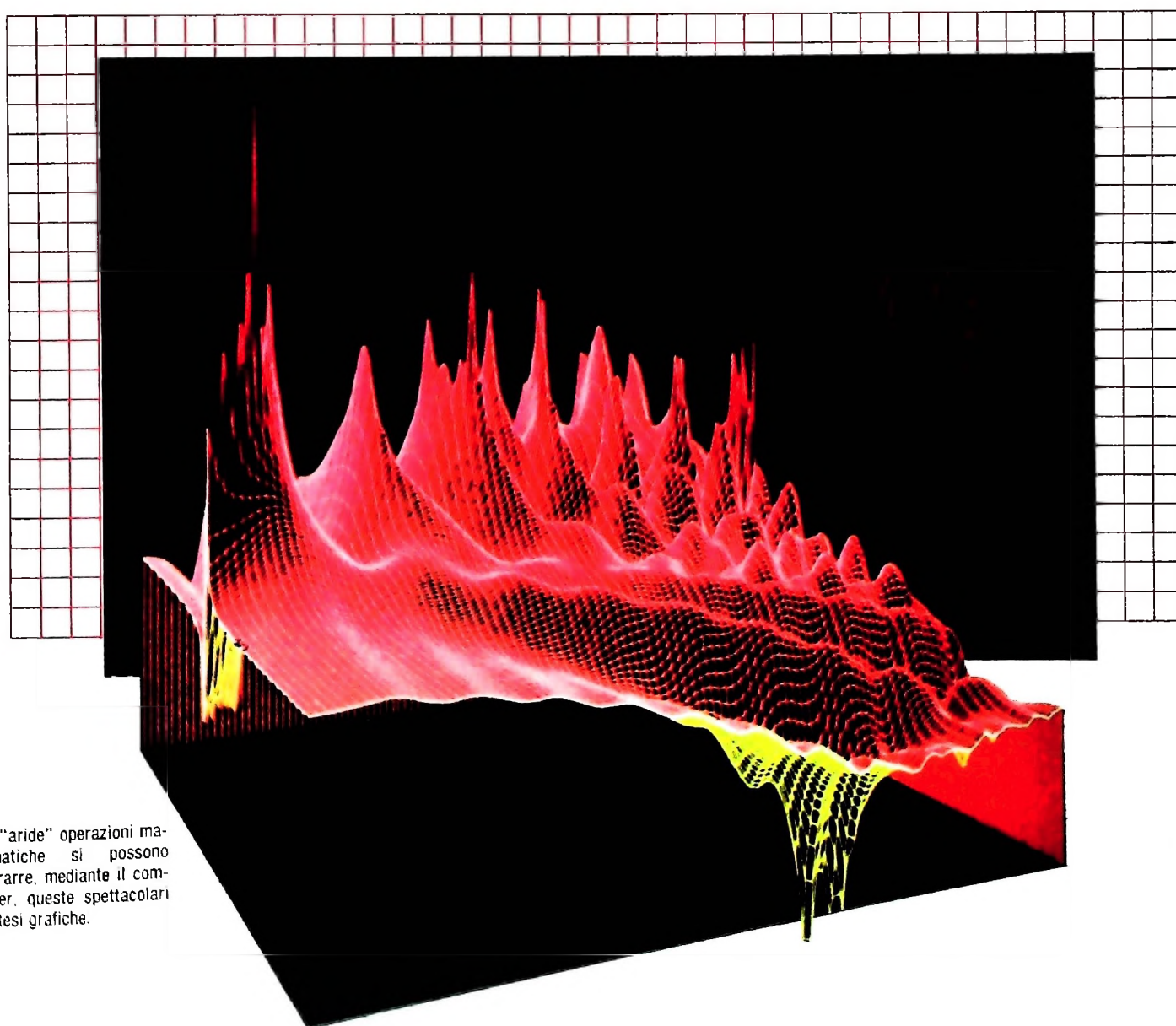
ra, le tavolette grafiche memorizzano qualunque disegno tracciato su di esse e lo rappresentano sul video, pronto per essere elaborato.

Infine la grande flessibilità dei programmi permette i più diversi impieghi in campo didattico.

Ormai l'elaboratore è entrato in modo permanente nell'attività didattica. Dopo un'esperienza pluriennale, in cui sono state create e sperimentate metodologie diverse, è tempo di riflettere e di controllare i risultati.

Avremo modo di tornare anche sulle metodologie con un discorso a parte.

Per ora vogliamo considerare due grandi campi riguardanti l'uso dell'elaboratore nella didattica: e precisamente il CAI al docente e il CAI al discente.



Da "aride" operazioni matematiche si possono estrarre, mediante il computer, queste spettacolari sintesi grafiche.

Computer Aided Instruction

La sigla CAI è una delle tante coniate per designare l'uso dell'elaboratore nella didattica; significa *Computer Aided Instruction*, cioè "didattica assistita da elaboratore". Si tratta di un'espressione molto generale che sottende tipi diversi di impiego. Noi vogliamo sottolineare che esiste una differenza fondamentale tra l'uso dell'elaboratore da parte del docente durante la lezione alla classe, e l'uso dell'elaboratore da parte del discente per un apprendimento più personalizzato. Soffermiamoci sul primo aspetto.

A meno che non si tratti di classi assolutamente ridotte, sarà ovviamente necessario un video a grande schermo, in modo che ogni allievo, stando al proprio posto, possa controllare gli ingressi e le uscite dell'elaboratore.

In questo caso si tratta di uno strumento che interviene nella lezione solo quando si rende necessario e ne integra perfettamente lo svolgimento. L'insegnante si serve dell'elaboratore così come si serve di una lavagna luminosa, a prescindere naturalmente dalle enormi differenze tra i due strumenti. Riportiamo due esempi, regolarmente sperimentati in una scuola media superiore, per l'elaboratore M20 della Olivetti.

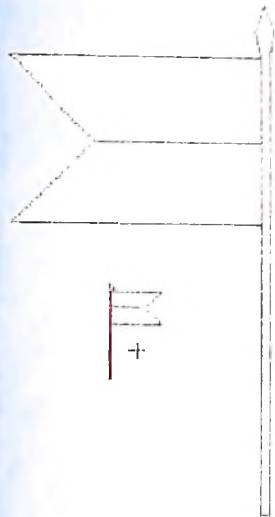
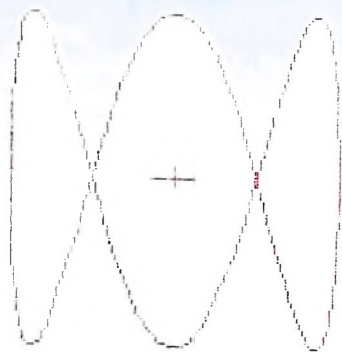
Programma tabella moltiplicazione per Z_n (per M20)

```
10 'Programma TABELLA MOLTIPLICAZIONE PER  $Z_n$ 
20 DIM T(43,43)
30 CLS: PRINT "TABELLA MOLTIPLICAZIONE PER  $Z_n$ " : PRINT
40 PRINT "Valore del modulo? (max=43)"
50 INPUT M
60 IF M<1 OR M>43 THEN 30
70 LPRINT "TABELLA MOLTIPLICAZIONE PER  $Z$ ": STR$(M): LPRINT
80 FOR C=0 TO M
90 T(0,C)=C-1
100 NEXT C
110 FOR R=1 TO M
120 T(R,0)=R-1
130 NEXT R
140 FOR R=0 TO M-1
150 FOR C=0 TO M-1
160 T(R+1,C+1)=(R*C) MOD M
170 NEXT C
180 NEXT R
190 A$=" X "
200 LPRINT A$:
210 FOR C=1 TO M
220 LPRINT TAB(C*3-LEN(STR$(T(0,C)))+3); STR$(T(0,C));
230 NEXT C
240 LPRINT: LPRINT
250 FOR R=1 TO M
260 FOR C=0 TO M
270 LPRINT TAB(C*3-LEN(STR$(T(R,C)))+3); STR$(T(R,C));
280 NEXT C
290 LPRINT
300 NEXT R
```

TABELLA MOLTIPLICAZIONE PER Z_6

X	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

La curva a destra, una figura di Lissajous, subisce una trasformazione isometrica, equivalente alla somma di una rotazione e una traslazione. La bandiera invece è sottoposta a una trasformazione simile (un ingrandimento o una riduzione) in questo caso inversa, perché l'immagine viene anche ribaltata.

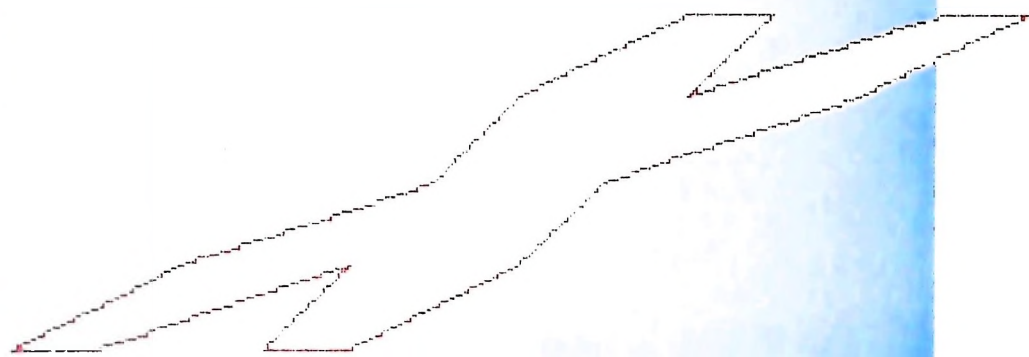
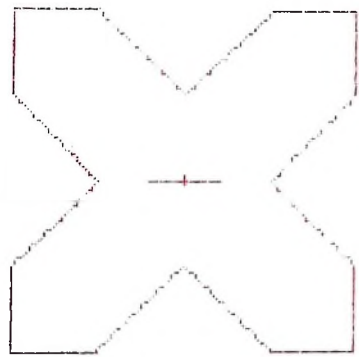


$A1 = .5$ $B1 = -.866025$ $C1 = 3$
 $A2 = .866025$ $B2 = .5$ $C2 = 2$
 $A3 = 0$ $B3 = 0$ $C3 = 1$

PARAMETRI CHE DEFINISCONO UNA TRASFORMAZIONE ISOMETRICA

$A1 = -5$ $B1 = 0$ $C1 = 0$
 $A2 = 0$ $B2 = 5$ $C2 = 0$
 $A3 = 0$ $B3 = 0$ $C3 = 1$

PARAMETRI CHE DEFINISCONO UNA TRASFORMAZIONE SIMILE



$A1 = 1$ $B1 = 2$ $C1 = 8.5$
 $A2 = 0$ $B2 = 1$ $C2 = 0$
 $A3 = 0$ $B3 = 0$ $C3 = 1$

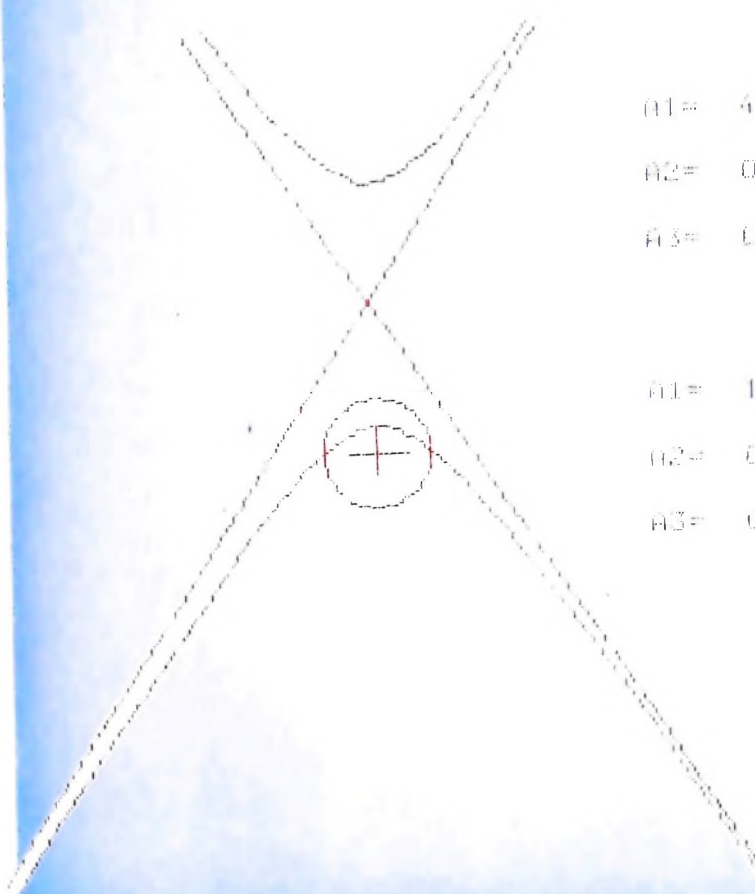
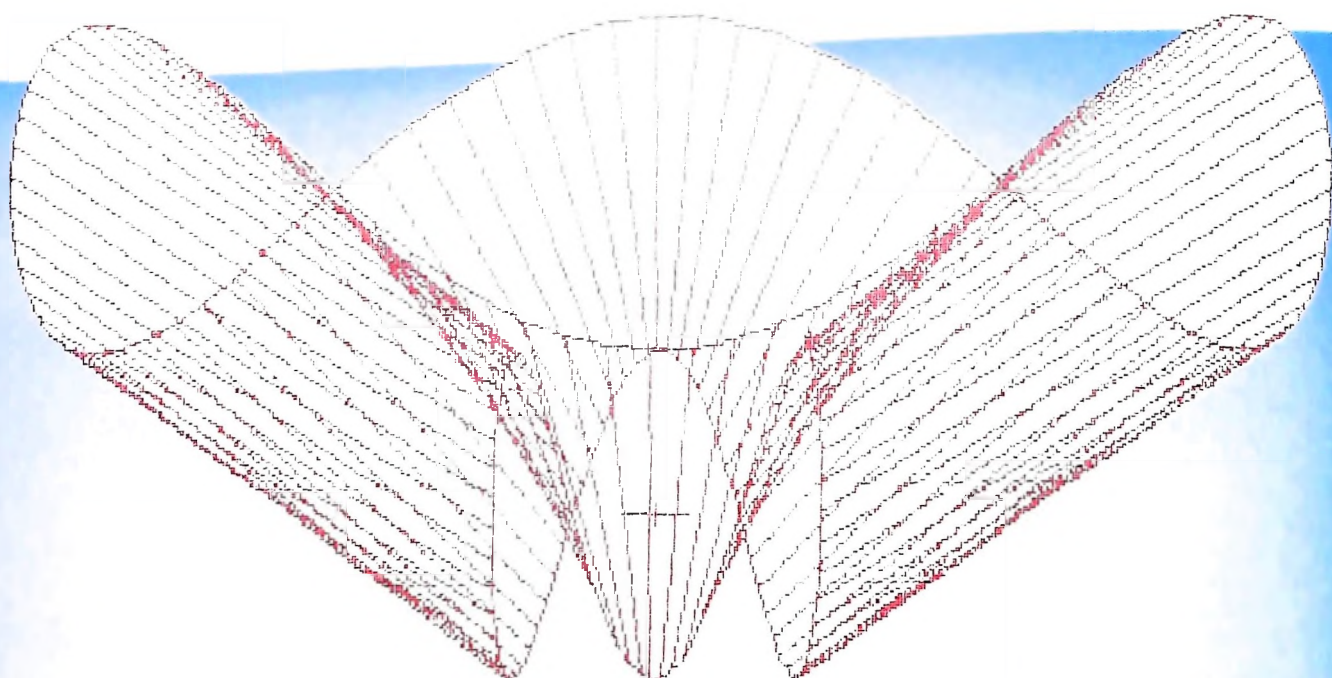
PARAMETRI CHE DEFINISCONO UNA TRASFORMAZIONE EQUIAFFINE

Questo è un esempio di trasformazione equiaffine. La figura di arrivo ha la stessa area della croce di partenza, e le linee parallele nell'originale rimangono tali anche dopo la trasformazione.

Due esempi

Il primo è un programma che calcola la "tabella di moltiplicazione" per Z_n . Z_n è l'insieme delle classi di resti, modulo n . Ciascuna classe è l'insieme costituito da tutti i numeri interi che, divisi per n , danno un certo resto. In altre parole, prefissato n (ad esempio $n = 6$), tutti i numeri interi (indicati con Z) vengono suddivisi in classi a seconda del resto della divisione per n di ciascuno di essi. Le classi sono appunto 6.

dal momento che i resti possono essere rispettivamente 0, 1, 2, 3, 4, 5. Ogni classe è rappresentata dai numeri più bassi, appunto i resti. Ora l'insieme di queste classi ha una struttura algebrica complessa, cioè di anello con unità o addirittura di corpo, se n è un numero primo. Esistono infatti due operazioni tra gli elementi: somma e prodotto. Indipendentemente dagli elementi sui quali le operazioni sono eseguite, i risultati sono identici. La tabella della moltiplicazione, come mostra l'illustrazione qui a fronte,



$$A1= 4 \quad B1= 0 \quad C1= 0$$

$$A2= 0 \quad B2= 1 \quad C2= 2$$

$$A3= 0 \quad B3= 0 \quad C3= 1$$

PARAMETRI CHE
DEFINISCONO UNA
TRASFORMAZIONE
AFFINE

$$A1= 1 \quad B1= 0 \quad C1= 0$$

$$A2= 0 \quad B2= 1 \quad C2= 0$$

$$A3= 0 \quad B3= 1,2 \quad C3= 1$$

PARAMETRI CHE
DEFINISCONO UNA
TRASFORMAZIONE
PROIETTIVA

In alto una trasformazione affine della figura di Lissajous rappresentata nella pagina precedente: una dilatazione lungo l'asse X (orizzontale) sommata a una traslazione lungo l'asse Y (verticale). Ci sono anche le "tracce", i collegamenti

tra i punti dell'originale e i punti corrispondenti del trasformato. Qui a lato, un esempio di trasformazione proiettiva. Con una trasformazione di questo tipo la sfera si trasforma in una iperbole, o in generale in una conica.

fornisce un esempio di come possono operare queste classi. Il secondo esempio è costituito da un programma molto più complesso del precedente. (e che non viene qui rappresentato), un programma che permette di studiare e illustrare le trasformazioni geometriche del piano.

Secondo la proposta del matematico tedesco Felix Klein (1872), la geometria è vista come lo studio delle proprietà invarianti delle figure che subiscono trasformazioni, più esatta-

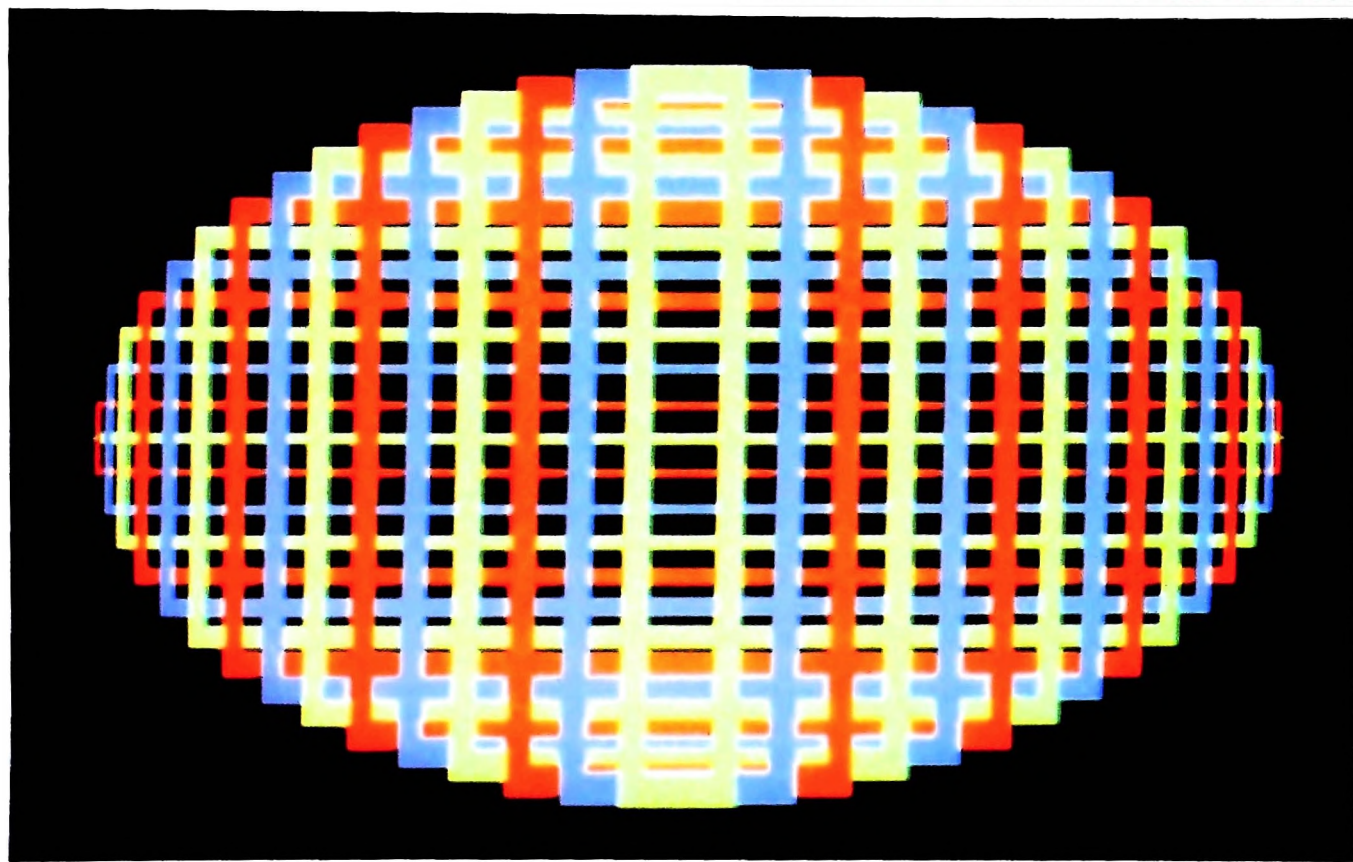
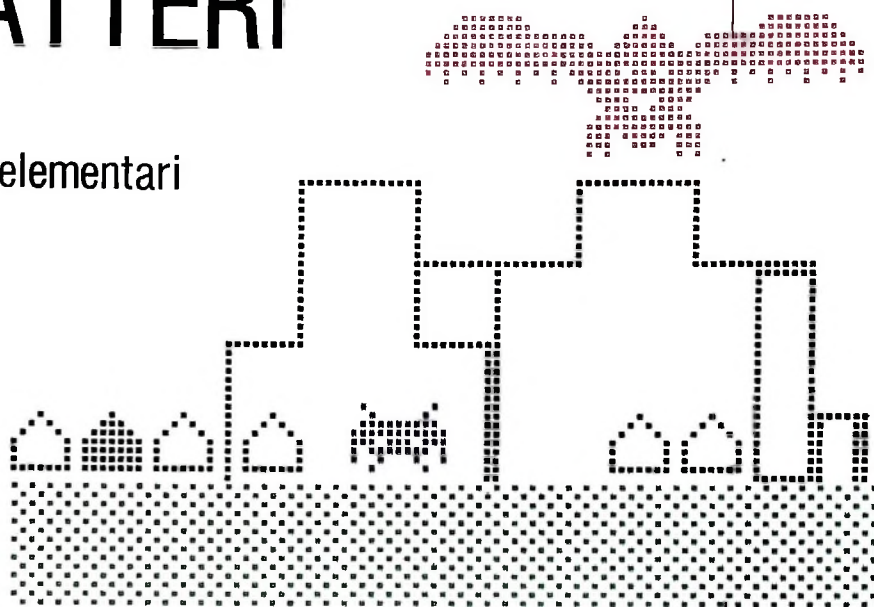
mente gruppi, in senso algebrico, di trasformazioni. Il programma permette di definire figure diverse; di definire il tipo di trasformazione attraverso i parametri della matrice che la esegue, e infine di visualizzare la figura di partenza e la figura trasformata.

I gruppi di trasformazioni presenti nel programma sono: isometrie, similitudini, equiaffinità, affinità e proiettività. Ne vediamo qualche esempio nelle illustrazioni di queste pagine.

GRAFICA A CARATTERI

Impariamo a simulare le istruzioni grafiche elementari con i caratteri grafici speciali: potremo impostare persino dei semplici videogiochi.

C'è una grande differenza fra la grafica al tratto o pittorica e la grafica a caratteri. M10 dispone di un set di caratteri alternativi utilizzabile per eseguire semplici disegni. In questa lezione impareremo a usare il tasto grafico GRPH. Per farci una prima idea, posizioniamo il cursore sul comando TEXT nel menù di M10 e battiamo il tasto ENTER. Ci



ACM ASSOCIATION - ARCHIVIO EIDOS

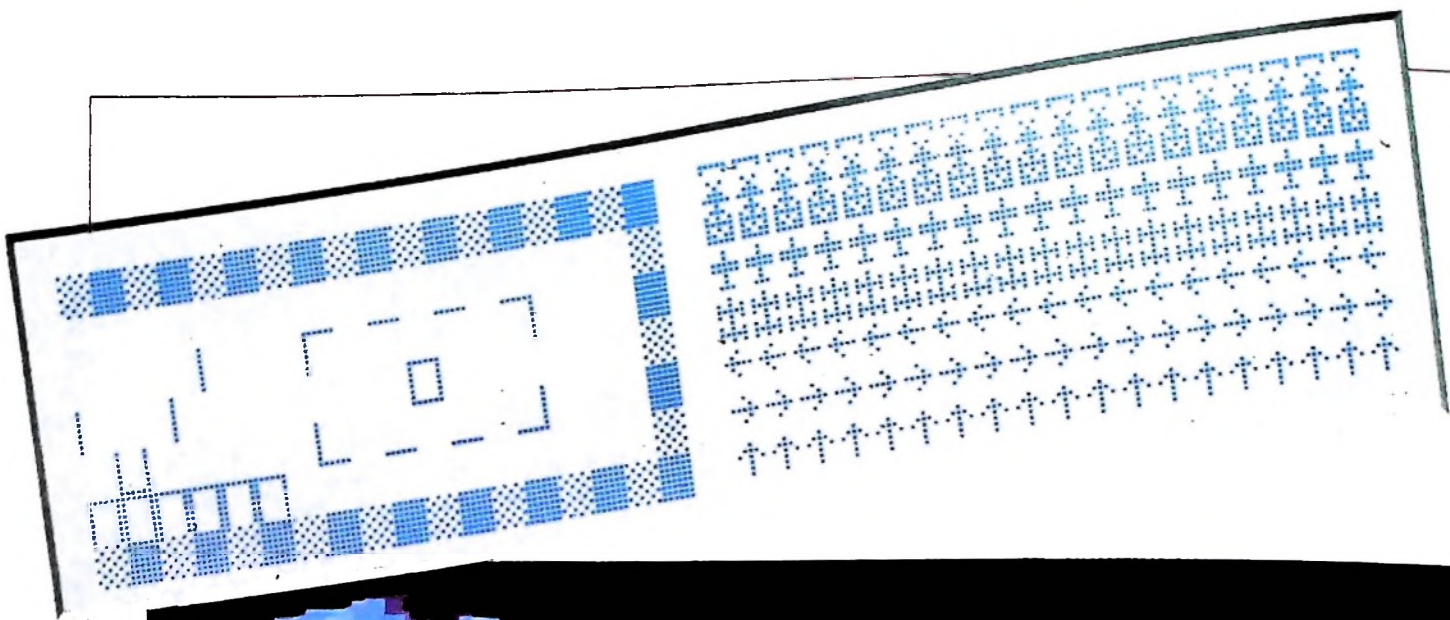
viene ora richiesto sullo schermo di dare un nome al file che stiamo creando. Chiamiamolo GRAPH. Ora, tenendo premuto GRPH, premiamo tutti gli altri tasti alfanumerici, partendo dal primo in alto a sinistra e dando uno spazio fra una battuta e l'altra e un ENTER alla fine di ogni riga di tasti.

Ripetiamo ora la stessa operazione tenendo premuti contemporaneamente i tasti GRPH e SHIFT.

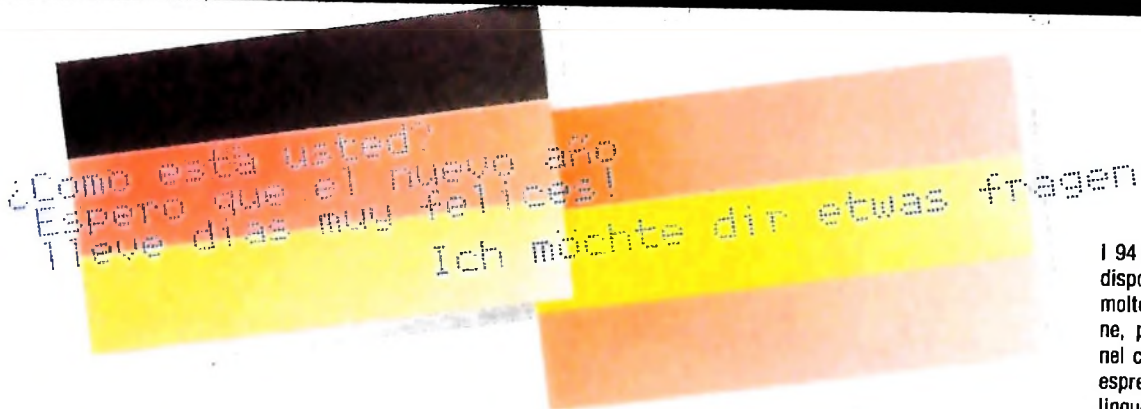
Otterremo in tal modo sullo schermo tutto il set dei 47 + 47

caratteri grafici che ci può mettere a disposizione M10 mediante l'utilizzazione del tasto GRPH.

Proviamo ora a premere il tasto funzione F8: siamo tornati al menù principale. Possiamo osservare che il file GRAPH è stato memorizzato automaticamente: se vogliamo rivedere la lista dei caratteri grafici basterà posizionare il cursore sul file GRAPH. DO (ricordiamo che il suffisso .DO indica che GRAPH è un file di testo) e premere ENTER.



IBM



I 94 simboli e caratteri grafici speciali disponibili sulla tastiera di M10 offrono molteplici possibilità di rappresentazione, particolarmente utili, per esempio, nel caso in cui si abbia a che fare con espressioni matematiche o con testi in lingue straniere.

Qualche uso dei caratteri grafici

Abbiamo quindi a disposizione una ricca scelta di caratteri, che vanno dai simboli matematici di integrale, sommatoria e radice sino alle lettere con tilde e dieresi tipiche di alcune lingue straniere.

Per esempio, possiamo formulare la famosa equazione di Einstein, oppure l'espressione di calcolo di un limite matematico, o anche una nota proprietà del calcolo integrale.

Possiamo anche scrivere ai nostri amici spagnoli o tedeschi nella loro lingua in modo più corretto e di gran lunga più gradevole per loro.

I caratteri più tipicamente grafici sono però quelli che si ottengono premendo contemporaneamente GRPH e tutti quelli dell'ultima riga in basso della tastiera. Combinando fra loro questi caratteri è possibile ottenere semplici disegni geometrici, istogrammi o prospetti per impaginare dei dati.

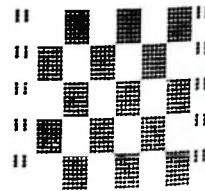
Caratteri grafici e istruzione LINE

Ognuno di questi caratteri, va notato, può essere ottenuto anche utilizzando l'istruzione LINE. Dipenderà dal tipo di problema che vogliamo risolvere utilizzare i caratteri grafici o l'istruzione LINE. Per esempio, consideriamo il seguente programma che realizza una piccola scacchiera:

```

10 CLS
20 PRINT
30 PRINT TAB(16)
40 PRINT TAB(16)
50 PRINT TAB(16)
60 PRINT TAB(16)
70 PRINT TAB(16)
80 END

```



Cognome	nome	indirizzo	tel.

```

10 CLS
20 LINE (0,0)-(239,63),1,B
30 PRINT
40 PRINT "   Cognome       Nome       Indirizzo   Tel."
50 PRINT
60 LINE (0,23)-(239,23)
70 LINE (0,33)-(239,33)
80 LINE (0,43)-(239,43)
90 LINE (0,53)-(239,53)
110 LINE (59,0)-(59,63)
120 LINE (113,0)-(113,63)
130 LINE (185,0)-(185,63)
140 END

```

Questo programma è realizzato utilizzando una serie di caratteri grafici e richiede cinque semplici istruzioni PRINT TAB (16), dove TAB è una opzione che serve a ottenere che il primo carattere sia stampato a partire dalla diciassettesima posizione della riga. Se volessimo disegnare la stessa scacchiera utilizzando l'istruzione LINE, il programma diventerebbe molto più lungo e di più lenta esecuzione. Se invece dovessimo realizzare una tabella come quella dis-

gnata qui sopra, ci converrebbe sicuramente usare l'istruzione LINE.

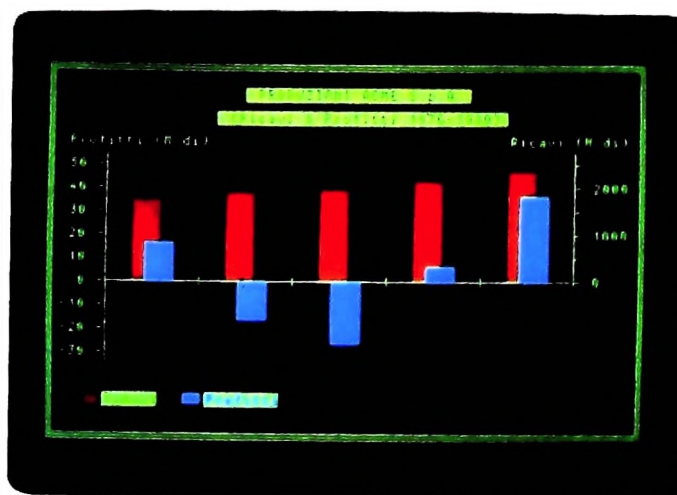
Ricordiamo che l'istruzione 20, per effetto dell'opzione B, disegna un rettangolo (BOX) avente per vertici i pixel di coordinate (0, 0) e (239, 63), ossia disegna il perimetro dello schermo.

Potete provare, per esercizio, a riprodurre la stessa tabella utilizzando i tasti grafici.



Le possibilità grafiche degli elaboratori (in questo caso un M20 Olivetti) rivestono notevole importanza nel campo della

gestione aziendale, per esempio nella fatturazione o nella visualizzazione dell'andamento economico.



Autovetture in corsa

Nel set dei tasti grafici di M10 sono presenti anche alcuni caratteri atti ad essere utilizzati per la realizzazione di semplici videogiochi. Per esempio, è possibile utilizzare il carattere grafico che rappresenta una vettura da corsa stilizzata e che si ottiene premendo contemporaneamente i tasti GRPH, SHIFT e V, per impostare un programma che simuli una competizione automobilistica. Di seguito dia-

mo il listato di tale programma. Si noti che questo, ovviamente, non è ancora un vero e proprio videogioco, mancando l'aspetto interattivo fra computer e giocatore. Questo semplice programma costituisce però un istruttivo avvicinamento alla realizzazione di un semplice effetto dinamico. Vedremo più avanti come ottenere invece un videogioco vero e proprio.

```
10 CLS
20 PRINT " "
30 PRINT " "
40 PRINT TAB(10) " "
50 PRINT TAB(10) " "
60 PRINT TAB(10) " "
70 PRINT TAB(10) " "
80 PRINT TAB(10) " "
90 INPUT
100 FOR I=1 TO 5
110 PRINT TAB(10) " "
120 PRINT TAB(10) " "
130 PRINT TAB(10) " "
140 PRINT TAB(10) " "
150 PRINT TAB(10) " "
160 PRINT TAB(10) " "
170 NEXT I
180 END
```

Commento al programma

Il programma è pressoché interamente costituito da istruzioni di PRINT. Con le istruzioni iniziali si prepara il terreno di gioco: vengono cioè disegnati i bordi della pista, la striscia di partenza e le vetture nelle varie posizioni. Il motore del programma è però il ciclo

di FOR all'istruzione 100. Esso forza a stampare per 5 volte (o per un numero a piacere di volte) le restanti istruzioni di PRINT, simulando così l'effetto dinamico, anche se necessariamente ripetitivo, della corsa.

Banca Aperta

LE NUOVE RISPOSTE DEL BANCO DI ROMA.



*Vorrei avere
un rapporto più diretto
con la mia banca...*

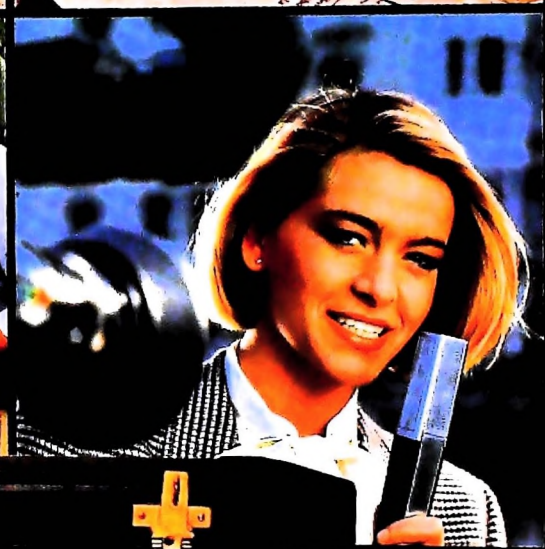
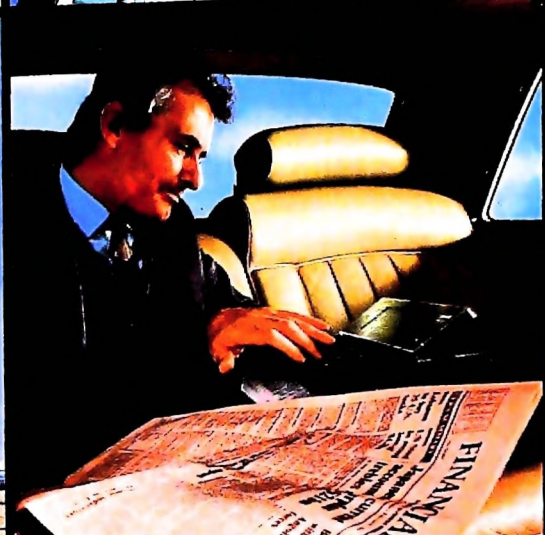
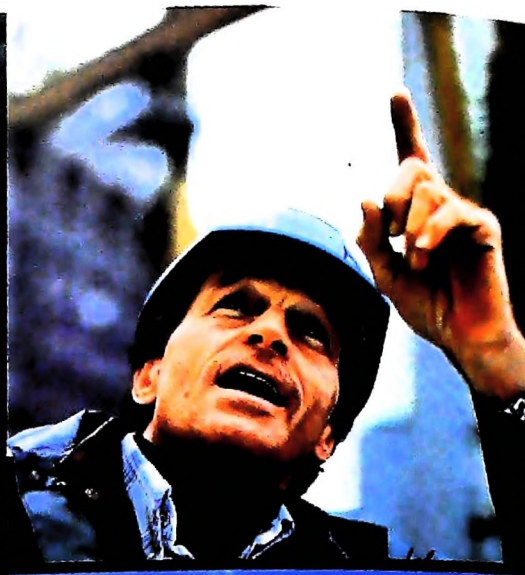
Anche le strutture bancarie si evolvono. Il Banco di Roma, primo in Italia, sta introducendo la struttura a "banca aperta", già attuata da molte sue filiali italiane. "Banca aperta": non il solito bancone, le lunghe file, ma un

nuovo modo di essere banca, un rapporto più personalizzato, un clima più agevole, più professionale e una maggiore rapidità in ogni operazione. Un ulteriore passo avanti verso la completa consulenza finanziaria che il Banco di Roma intende mettere a disposizione dei propri clienti. Tra i numerosi servizi offerti ricordiamo: Prestito Personale. Prestito Casa, gestione dei patrimoni. Leasing, assistenza all'import-export, attraverso ben 60 sedi

estere in 30 Paesi dei 5 continenti. Tutto questo perché il Gruppo Banco di Roma è in grado di gestire ogni servizio specifico con grande professionalità, fornendo anche informazioni dirette a domicilio attraverso i sistemi Videotel e Voxintesi.



 **BANCO DI ROMA**
CONOSCIAMOCI MEGLIO.



PERSONAL COMPUTER OLIVETTI M10 L'UFFICIO DA VIAGGIO

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 collegarsi via telefono per spedire o ricevere informazioni.

Qualunque professione sia la vostra, M10 è in grado, dovunque vi troviate, di offrirvi delle capacità di soluzione davvero molto grandi. M10: il più piccolo di una grande famiglia di personal.

Per informazioni rivolgersi ai negozi contrassegnati da "Olivetti M10 Punto di Vendita" o inviare coupon a Olivetti, Divisione Personal Computer, Via Meravigli 12, 20123 Milano.

NOME/COGNOME

VIA/N.

CAPACITÀ

TELEFONO

olivetti