

6 CORSO PRATICO COL COMPUTER

421586

F4

F5

F6

F7

F8

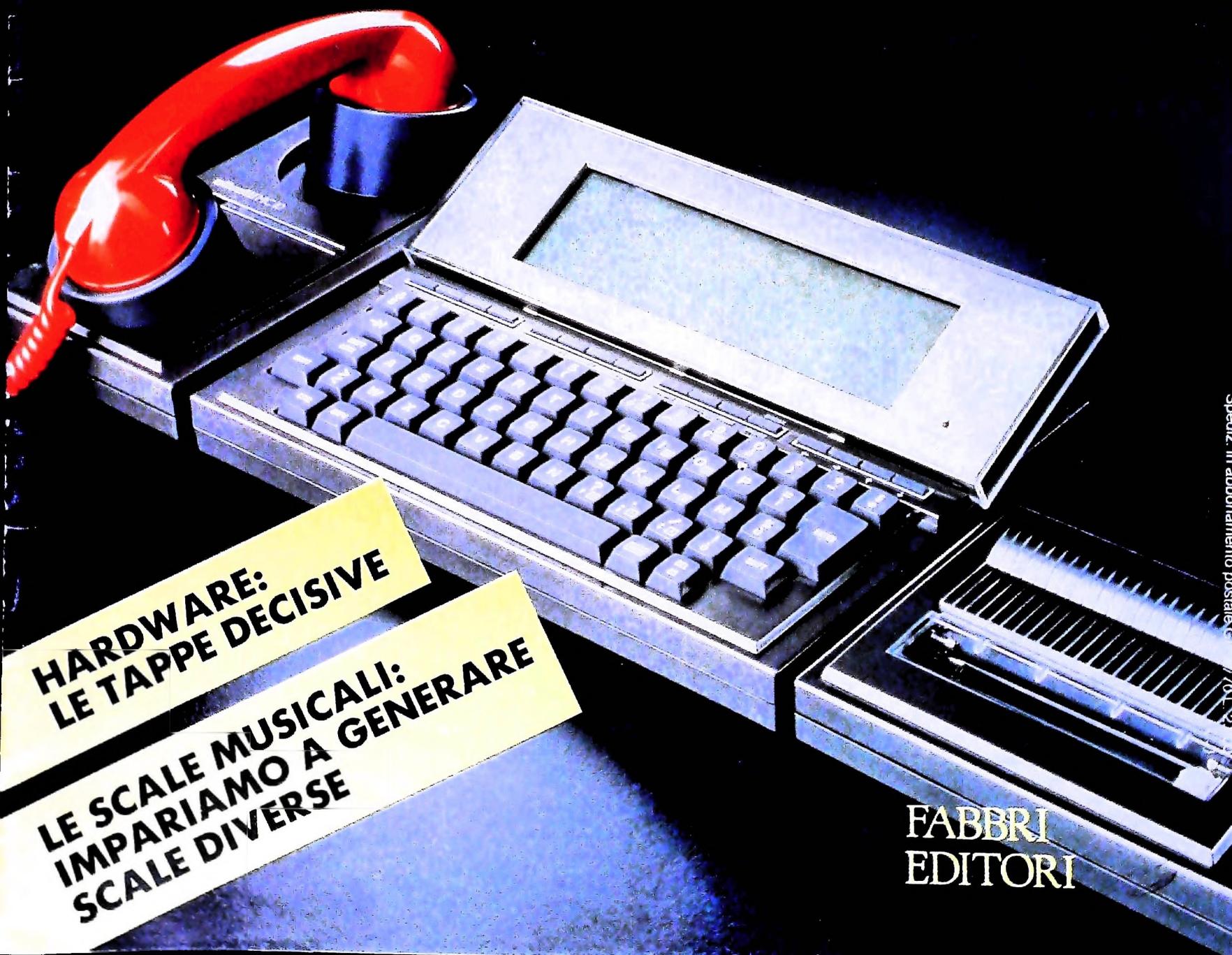
diretta da **GIANNI DEGLI ANTONI**

è una iniziativa
FABBRI EDITORI

in collaborazione con
BANCO DI ROMA

e **OLIVETTI**

BATTERY LOW.



**HARDWARE:
LE TAPPE DECISIVE**

**LE SCALE MUSICALI:
IMPARIAMO A GENERARE
SCALE DIVERSE**

**FABBRI
EDITORI**

Spedizione in abbonamento postale

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.

Potete 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 listararlo, 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 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 MAJOCCHI
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 MAJOCCHI, DANIELE MARINI, GIANCARLO MAURI, CLAUDIO PARMELLI, ENNIO PROVERA

Testi
ADRIANO DE LUCA, GOFFREDO HAUS, Eidos (DANIELE MARINI, TIZIANO BRUNETTI), 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.N.C. - Milano

Direttore Editoriale
ORSOLA FENGLI

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 BREGGÈ

Segretarie di Redazione
RENVATA FRIGOLI
LUCIA MONTANARI

**NEL PROSSIMO NUMERO
IN OMAGGIO
IL SECONDO 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, Etas S.p.A., Milano - Prima Edizione 1984 - Direttore responsabile GIOVANNI GIOVANNINI - Registrazione presso il Tribunale di Milano n. 136 del 10 marzo 1984 - Iscrizione al Registro Nazionale della Stampa n. 00262, vol. 3, Foglio 489 del 20.9.1982 - Stampato presso lo Stabilimento Grafico del 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 - Pubblicazione periodica settimanale - Anno I - n. 6 - 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 TAPPE DECISIVE

I componenti essenziali del calcolatore:
lo schema di Von Neumann.

Probabilmente il primo calcolatore di cui si abbia conoscenza è l'abaco, usato in Oriente 3000 anni fa. Questa macchina, ancor oggi largamente usata in molte parti del mondo, non ebbe rivali sino al 1600, quando John Napier usò i logaritmi come base per la creazione di un sistema (i "bastoncini di Nepero") capace di moltiplicare e dividere numeri. I progressi da allora sono stati giganteschi: attraverso la macchina di Pascal, le calcolatrici elettromeccaniche, le macchine a schede perforate, si è giunti al calcolatore elettronico, come l'ENIAC, che usava ben 18000 valvole (tubi a vuoto), con un consumo enorme di energia elettrica e, infine, ai moderni microprocessori con più di centomila transistori interni e un consumo di energia quasi irrisorio.

Da un'analisi di tutta questa rivoluzione tecnica si riescono a individuare tre importanti scoperte che hanno permesso la

rapida evoluzione che ha portato ai calcolatori di oggi. La prima è la proposta di John Von Neumann di memorizzare il programma oltre che i dati. Esso può essere prelevato dalla memoria ed eseguito, oltre che modificato. Questa proposta risolverà le grandi difficoltà incontrate nel programmare l'ENIAC, primo calcolatore basato su tecnologie elettroniche. La seconda fu la scoperta del transistor (1947), che riduce drasticamente sia il volume degli elementi attivi (o "porte") sia il consumo di energia elettrica.

transistore di potenza



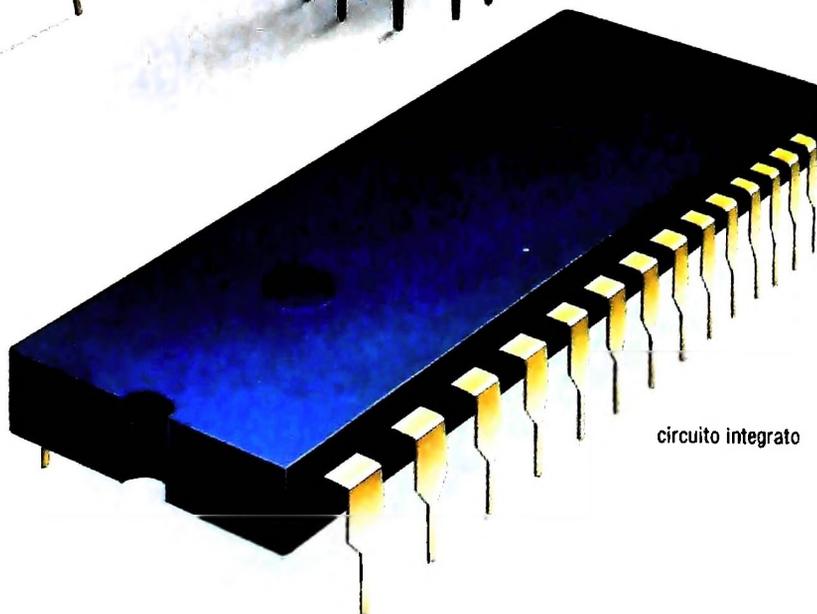
transistore



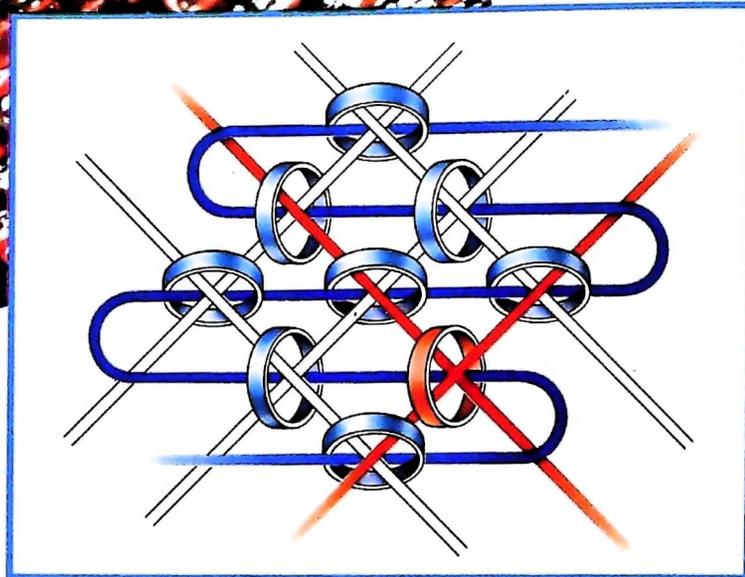
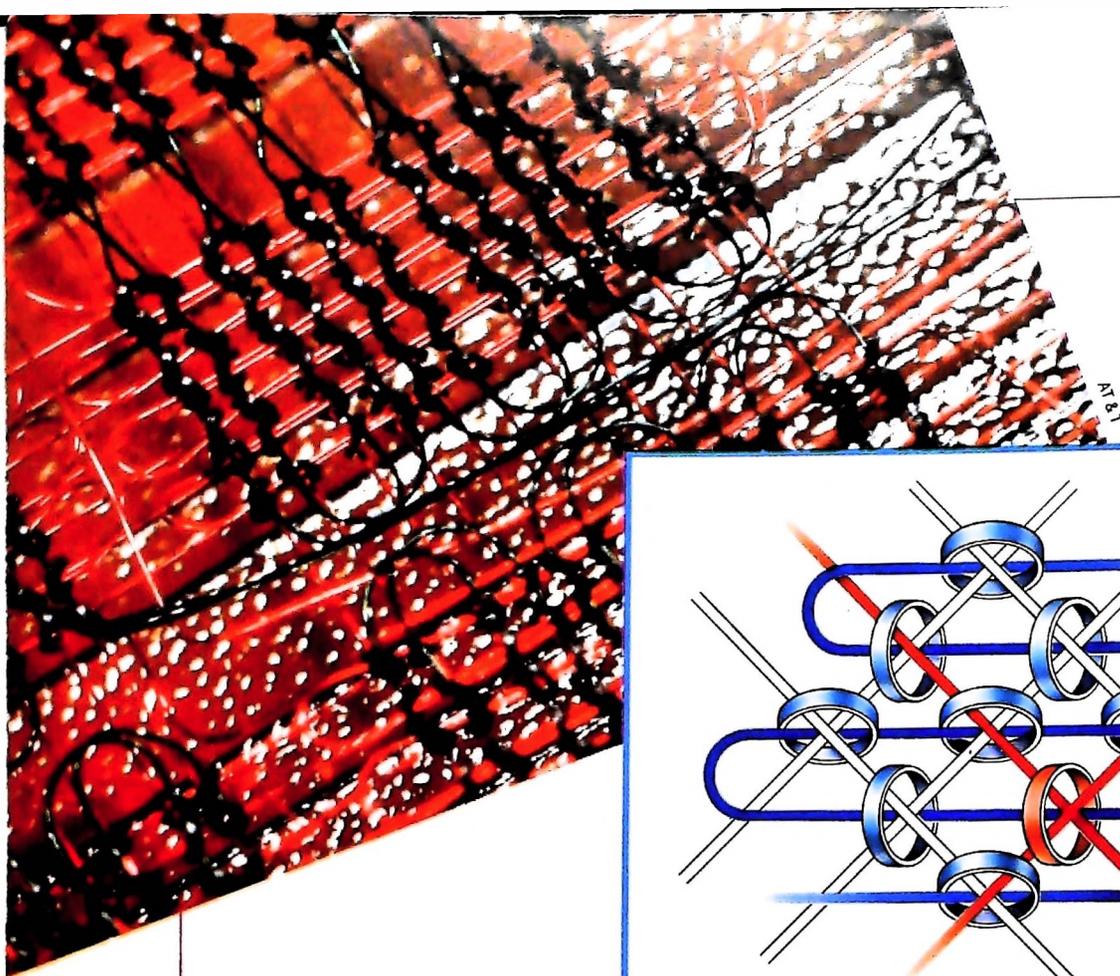
tubo a vuoto



Tre generazioni di componenti elettronici (tubi a vuoto, transistori discreti e circuiti integrati). I tubi a vuoto ("valvole") simboleggiano la prima generazione di calcolatori elettronici, che all'incirca corrisponde agli anni Cinquanta; i transistori discreti simboleggiano la seconda generazione (gli anni Sessanta), ma già si affacciava la terza, cioè quella basata sui circuiti integrati, che va dagli anni Settanta a oggi.



circuito integrato



La memoria ad anelli di ferrite, prima memoria veloce di un calcolatore dotata di una certa capacità. Ideata da J.W. Forrester, lo rese ricco e famoso. Segnò il primo impiego su larga scala di manodopera femminile e del Terzo Mondo nell'industria elettronica moderna, dando luogo al fenomeno della "parcellizzazione" della costruzione di un calcolatore elettronico, che oggi viene assemblato con componenti fabbricati in ogni parte del mondo, dovunque la manodopera costi meno o gli impianti di produzione siano più efficienti. Il tutto è poi spedito per via aerea sino ai paesi dove viene commercializzato, spesso sotto più marchi di fabbrica.

La terza fu l'invenzione delle memorie magnetiche a nuclei di ferrite (J.W. Forrester e collaboratori), che permise per la prima volta di immagazzinare un gran numero di dati.

Il calcolatore digitale

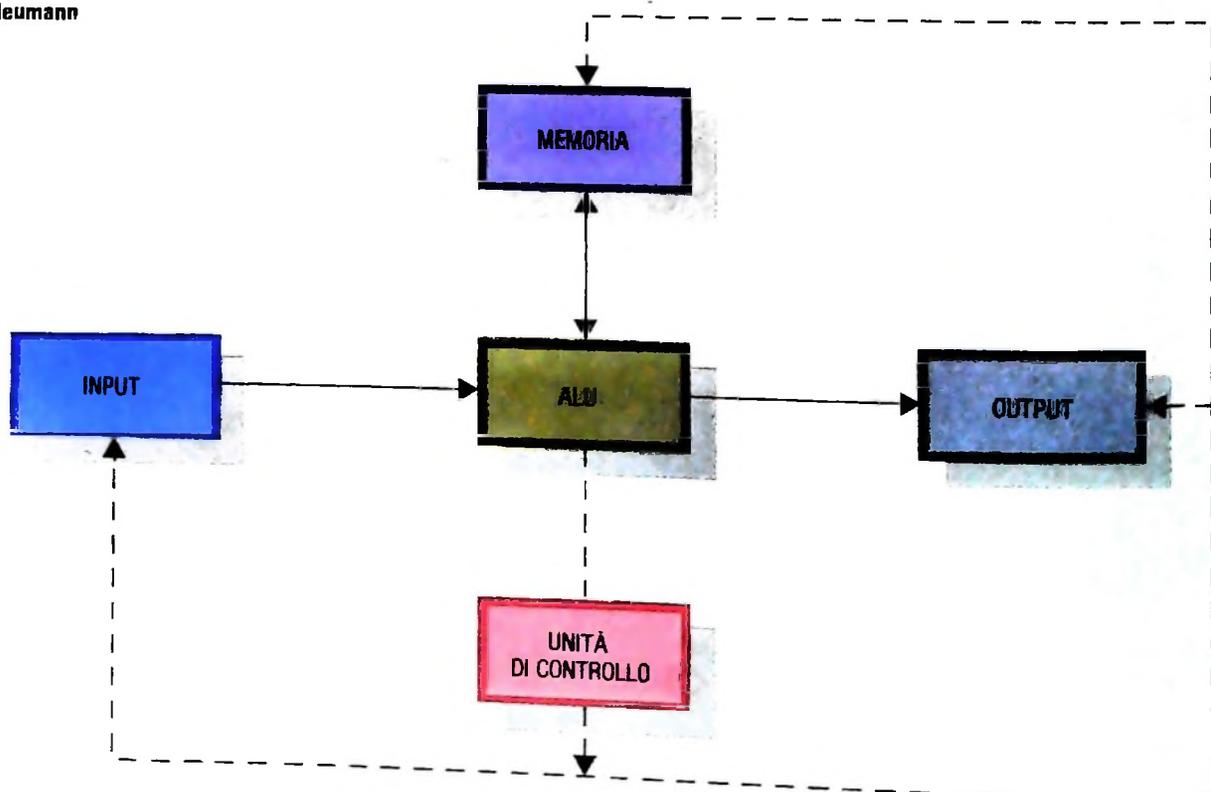
Che cos'è un calcolatore digitale? Esso è una macchina elettronica formata da vari gruppi di elementi capaci di eseguire

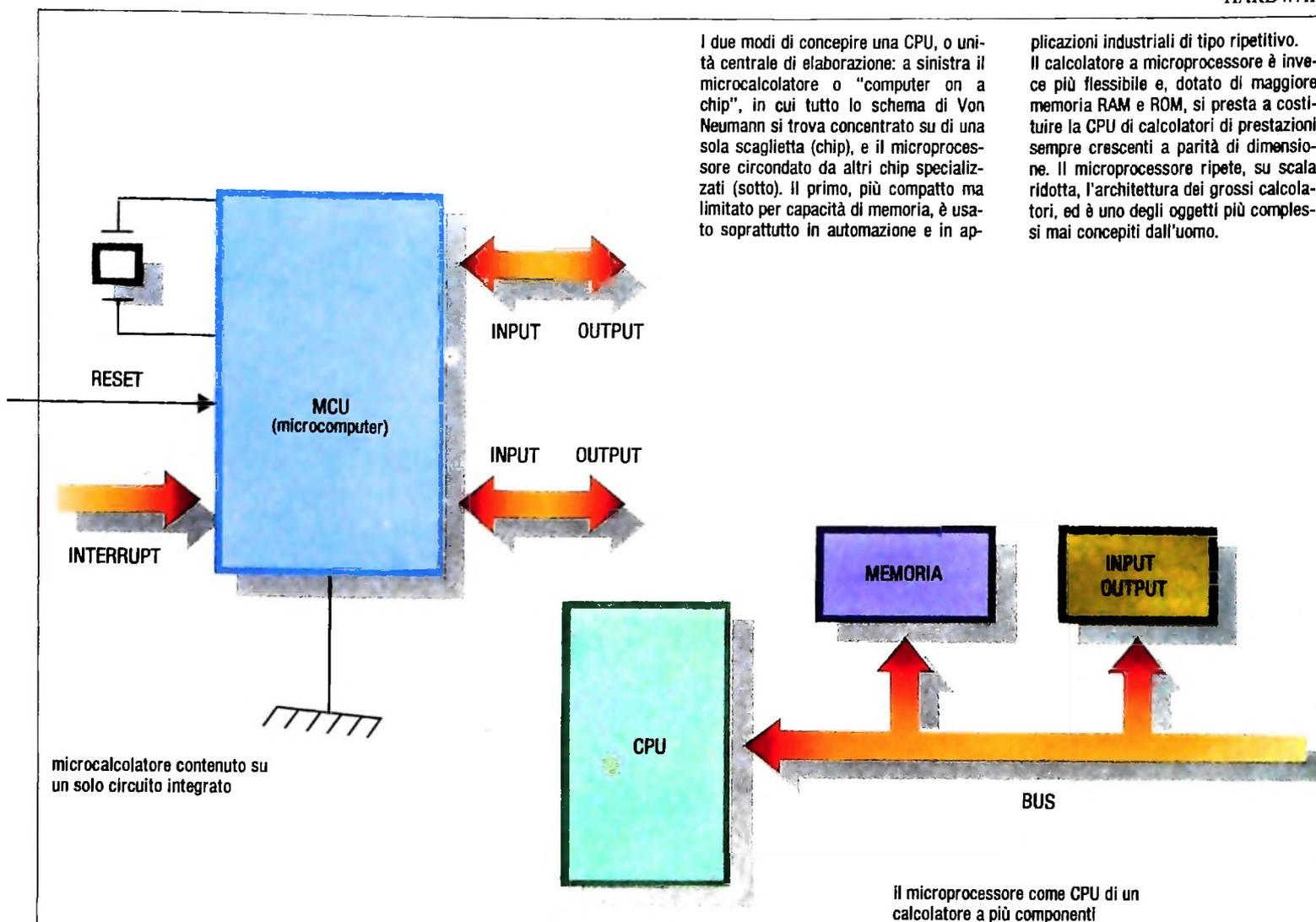
determinate operazioni. Si osservi la figura in basso: in essa individuiamo immediatamente lo schema classico di Von Neumann.

Analizziamo le funzioni di ogni blocco della figura, cominciando dall'UNITÀ DI CONTROLLO.

Come si vedrà, esso è il vero e proprio "cervello" della macchina in quanto è capace di andare a prendere ("fetch") le istruzioni e i dati dalla MEMORIA e, dopo aver analizzato le prime, agire sui secondi. La sua funzione è paragonabile a

Schema di Von Neumann





quella di un direttore d'orchestra che controlla sia nel tempo sia nell'esecuzione l'intervento dei vari strumenti, onde trarne l'armonia voluta.

Molte delle grandi capacità dei calcolatori moderni risiedono non soltanto nella potenza delle singole istruzioni quanto nell'abilità peculiare di uscire momentaneamente dall'esecuzione di una lista di istruzioni per andare ad un altro punto della memoria e da lì partire per eseguire un'altra lista, per poi tornare alla prima, nello stesso punto di partenza, e riprendere il processo lasciato in sospeso.

L'ALU (*Arithmetic Logic Unit*), comunemente considerato il "cuore" della macchina, è la parte del sistema capace di eseguire sui dati operazioni logiche (quelle dell'algebra di Boole) ed aritmetiche.

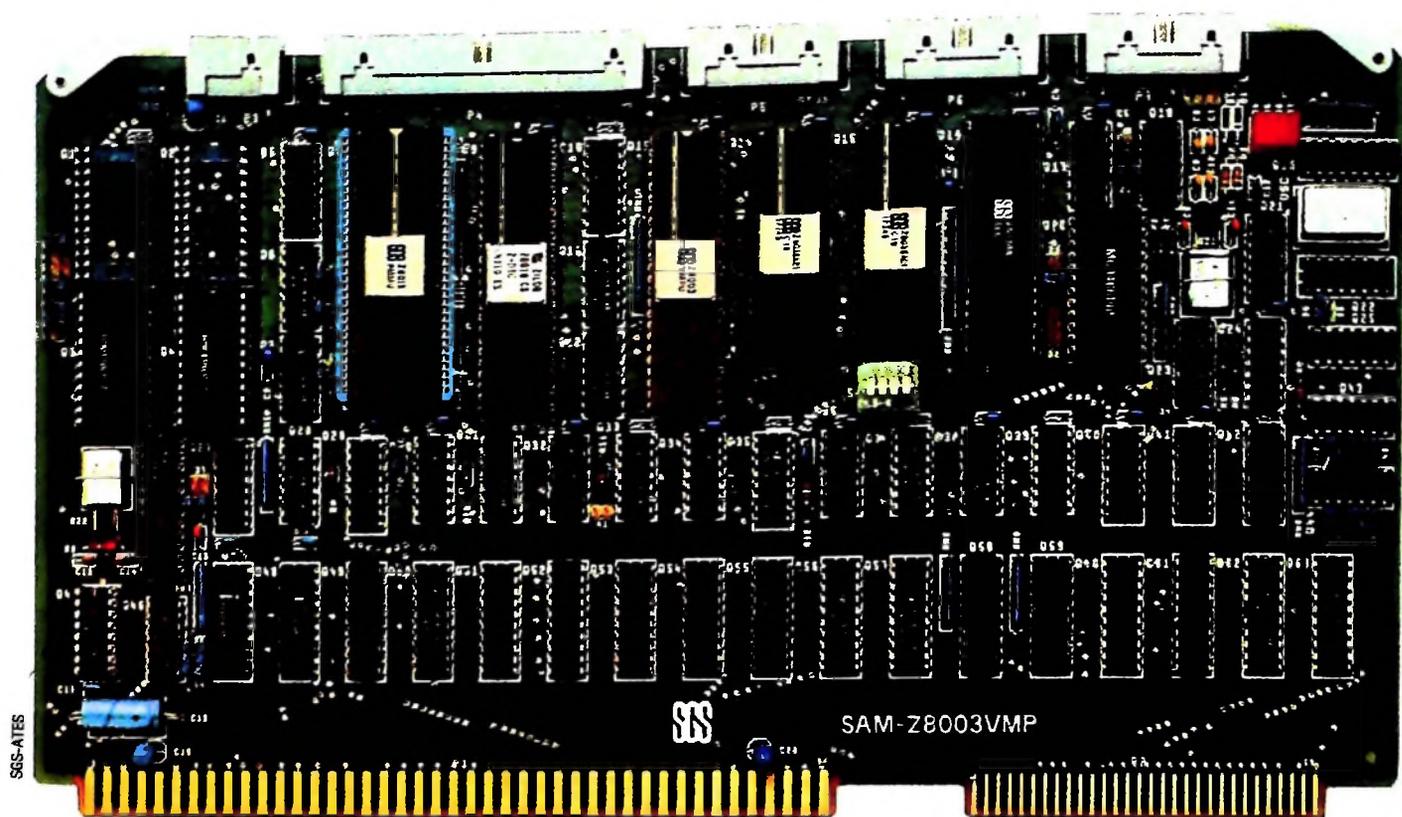
Per quanto riguarda i blocchi INPUT e OUTPUT (ingresso e uscita), essi sono circuiti capaci di mettere in comunicazione il mondo esterno al calcolatore con il suo mondo interno. Prima di continuare su questo tema, è opportuno chiarire alcuni concetti riguardanti l'uso corretto delle parole Microcalcolatore (*Microcomputer*) e Microprocessore (*Microprocessor*). Con la prima si indica un calcolatore digitale formato da un unico circuito integrato, che racchiude internamente

tutti gli elementi visti nello schema di Von Neumann ed è capace di funzionare da solo, salvo alcuni circuiti ausiliari esterni. A questa famiglia appartengono quasi tutti i calcolatori usati per il controllo dei processi industriali.

Se, invece, il calcolatore è composto da una serie di circuiti integrati separati, il più importante di essi è il Microprocessore, comunemente chiamato CPU (*Central Processing Unit*). Esso è collegato agli altri circuiti di memoria, INPUT/OUTPUT eccetera, attraverso un "canale" di comunicazione comunemente chiamato "bus".

Dopo questa breve parentesi volta a chiarire la terminologia, riprendiamo l'analisi degli elementi che compongono il calcolatore digitale dal punto di vista dei circuiti (detti anche hardware).

Sia l'ALU sia l'UNITÀ DI CONTROLLO sono di solito costruiti usando elementi elettronici attivi a semiconduttori disposti entro schemi determinati. I calcolatori detti della seconda generazione (dopo quelli a tubi elettronici della prima generazione) utilizzavano componenti come transistori, diodi, resistori montati su schede a circuiti stampati, con al massimo due facce di piste di collegamento fra gli elementi attivi (cioè le due facce visibili della scheda, realizzata in bachelite



Una scheda moderna di CPU. Essa contiene, entro uno spazio molto ridotto, tutti i componenti essenziali di un calcolatore elettronico basato su di un microprocessore (In questo caso lo Z8003

della famiglia Z8000 a 16 bit). Oltre allo Z8003 sono visibili numerosi altri componenti della stessa famiglia, che fanno di questa scheda una CPU completa e autosufficiente. Questa scheda corri-

sponde all'incirca a due dei blocchi già visti nello schema di Von Neumann, l'ALU e l'unità di controllo. Il collegamento con altre schede avviene tramite un "bus"; in pratica, le schede sono inseri-

te in appositi cestelli, che collegano elettricamente i connettori visibili nella parte inferiore della scheda. In questo caso, anzi, i bus sono 2 (uno a sinistra e uno a destra), con funzioni diverse.

in un primo tempo e fibra di vetro successivamente.

I modelli dalla terza generazione in poi usano non più elementi discreti ma circuiti integrati montati su schede; le piste di collegamento fra gli elementi non corrono più solo lungo le due facce visibili, ma anche entro altre facce "nascoste" all'interno della scheda, che è a più facce.

L'unità di memoria è quasi sempre costruita da elementi semiconduttori ad alta velocità; le memorie magnetiche sono tuttora usate come memoria di massa, per la grande capacità di immagazzinamento. Fra i sistemi di memoria di massa di uso corrente figurano dischi (flessibili o rigidi, di vari tipi) e nastri (anch'essi in vari tipi): alcuni di essi sono mostrati nella pagina opposta.

Ogni unità di memoria è divisa in "celle" di otto bit (unità elementari di informazione), byte, o multipli di otto chiamate parole. Ad ogni parola corrisponde un indirizzo.

Il concetto di indirizzo di memoria di una determinata parola è perfettamente equivalente a quello di un indirizzo postale, caratterizzato dal nome di una via e da un numero. Analogamente, l'indirizzo di una parola in memoria è un numero che, come si vedrà, viene diviso convenzionalmente in due parti, di cui una indica il blocco di memoria (la memoria è

talvolta organizzata in blocchi) e l'altra rappresenta la posizione del blocco.

Ogni tipo di memoria è caratterizzato da un tempo di ciclo, definito come il tempo necessario per leggere o scrivere una parola in memoria, dato questo molto importante e da tener presente nella progettazione.

Il calcolatore comunica con l'esterno, come si è visto, attraverso i «blocchi» di INPUT e OUTPUT e vari tipi di unità periferiche, tra cui la tastiera (con o senza terminale video) e la stampante. Il terminale video con tastiera è diventato oggi il mezzo più comune di interfaccia uomo-calcolatore: su di esso è possibile visualizzare sia i programmi sia i risultati dell'elaborazione dati in codice alfanumerico (cioè mediante lettere dell'alfabeto e numeri) oppure mediante rappresentazione grafica in bianco e nero o a colori. Altre periferiche sono i plotter (per disegnare), i lettori di nastri e schede perforate e i lettori di schede magnetiche, i lettori ottici, i lettori di codici a barre. Se il calcolatore è usato per il controllo di macchine utensili oppure di processi industriali, gli elementi più comuni di interfaccia sono i convertitori digitali/analogici o analogici/digitali, che trasformano un segnale analogico (continuo) in un segnale digitale (discreto) e viceversa.

Sommario

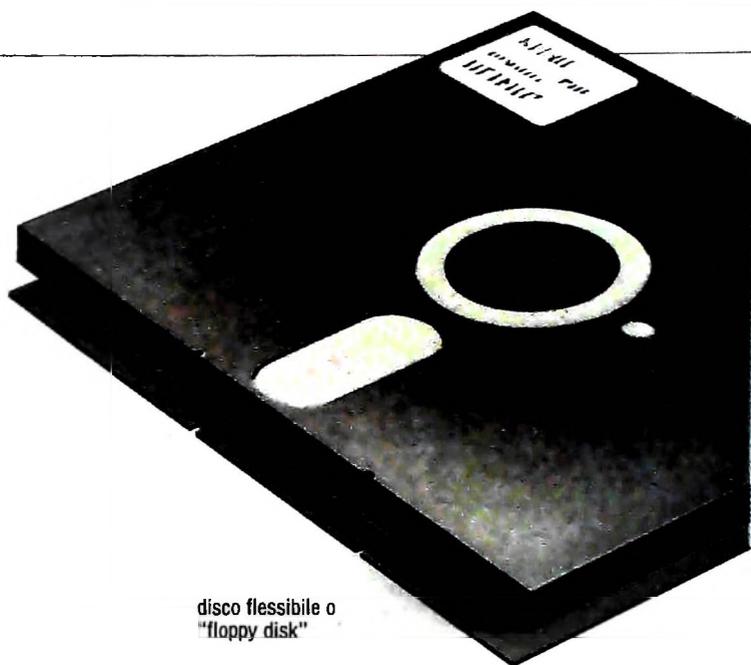
Concludendo questa prima parte del discorso, possiamo dire che l'unità di memoria contiene i dati e le istruzioni di un programma particolare. L'unità di controllo esegue la lista di istruzioni, dirige e attiva l'unità logico-aritmetica e quelle di ingresso e uscita fino alla fine del programma. Ogni unità esegue il suo compito sincronizzata e regolata dall'unità di controllo. I temi che verranno illustrati prossimamente saranno i seguenti: le porte (*gate*), gli elementi a tre stati logici (*tri-state*) e, importantissimo, il BUS.

nastro magnetico



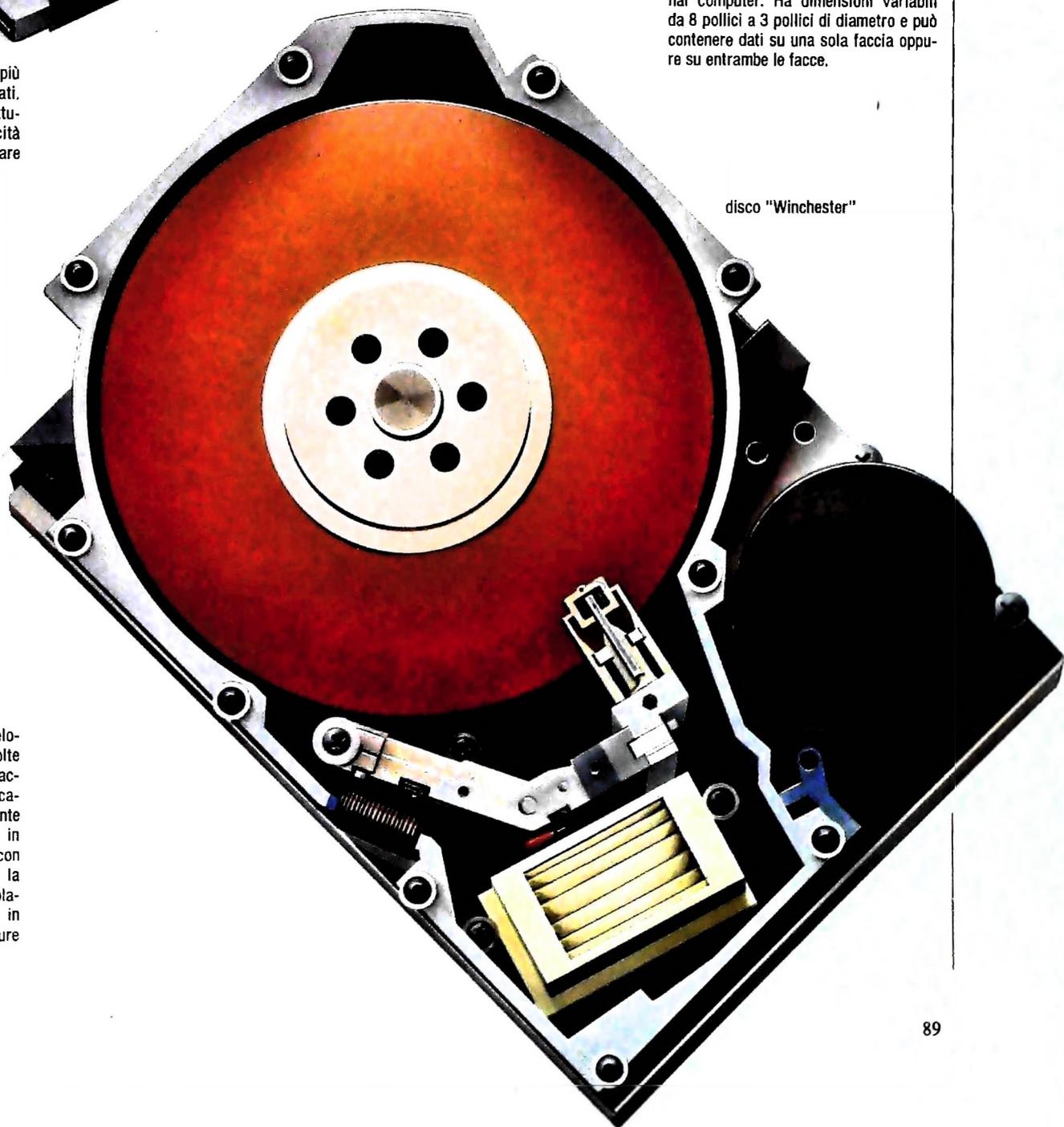
Il nastro magnetico è uno dei mezzi più comuni di immagazzinamento dei dati. Relativamente lento in lettura e scrittura, possiede però grandissima capacità ed è usato soprattutto per archiviare grandi quantità di dati.

disco flessibile o "floppy disk"



Il disco flessibile o "floppy disk" è la memoria di massa più usata nel personal computer. Ha dimensioni variabili da 8 pollici a 3 pollici di diametro e può contenere dati su una sola faccia oppure su entrambe le facce.

disco "Winchester"



Il disco "Winchester" gira a forte velocità (3600 giri/min), circa dieci volte più rapidamente del floppy, per cui l'accesso ai dati è molto più veloce. Più capace, ma più costoso e meccanicamente più delicato del floppy, è contenuto in scatole a tenuta stagna riempite con gas inerte. Si sta imponendo come la memoria di massa principe per calcolatori piccolo-medi. Spesso è posto in tandem con uno o più floppy, oppure con un registratore a nastro.



DIDATTICA E INFORMATICA

Un binomio che si va affermando nella scuola di oggi e si affermerà sempre di più nella scuola di domani.

L'elaboratore elettronico non è più un semplice strumento di calcolo ma fornisce prestazioni di grande complessità, utilizzabili anche in numerosi settori della didattica. È ovvia la comparsa dell'elaboratore negli istituti per ragionieri programmatori o per periti informatici, ma va regolarmente aumentando il numero delle scuole in cui si insegna informatica o si utilizza l'elaboratore nella didattica. Questo binomio

L'informatica ha qualcosa di nuovo da dire nel campo della didattica: il suo ruolo non si esaurisce nei semplici ausili, analoghi agli ausili tradizionali (anche se questo aspetto non è certo da trascurare), ma tocca la stessa struttura profonda, nel punto nodale, nella avvio di una disciplina, come nell'evoluzione della didattica, è il "problema", l'emergere di esigenze inedite, di lacune, di conflitti, di vere e proprie contraddizioni. I contenuti delle discipline vanno visti come risposte ai problemi, e l'informatica offre i metodi e gli strumenti più flessibili per trattare i problemi veri.

merita dunque particolare attenzione: ma come parlarne?

Il lettore non si stupisca se affrontiamo il tema partendo apparentemente da molto lontano, domandandoci quale sia il ruolo fondamentale della didattica nella società in generale. La risposta non è difficile. La didattica ha il compito di ottimizzare il processo di "allevamento dei cuccioli dell'uomo". Non sembri esagerata questa impostazione. In termini più scientifici possiamo dire che la sopravvivenza di qualunque sistema in evoluzione è strettamente legata alla capacità di conservare le informazioni legate al sistema stesso, di diffonderle a tutti i propri componenti e di trasmetterle ai componenti che vengono man mano a far parte del sistema stesso.

L'umanità stessa, nel suo complesso, non sfugge a questa regola: se vogliamo sopravvivere dobbiamo conservare, diffondere, trasmettere ai successori l'insieme delle conoscenze su ciò che siamo e su ciò che facciamo.

La didattica, nei suoi aspetti più generali, rappresenta la somma degli sforzi per rendere ottimale questo processo.

È chiaro a questo punto che si apre tutto il discorso sui vari aspetti della didattica, con i relativi problemi: ne accenniamo qualcuno.

Anzitutto il tema della storia: sia in generale come storia del-

l'umanità sia in particolare come storia della singola disciplina oggetto di studio. Come porsi in rapporto ad essa; come sentirla; come rapportarla alla vita di ogni giorno; come sentirsi inseriti in essa nel momento dell'insegnamento e come farvi sentire inseriti gli allievi.

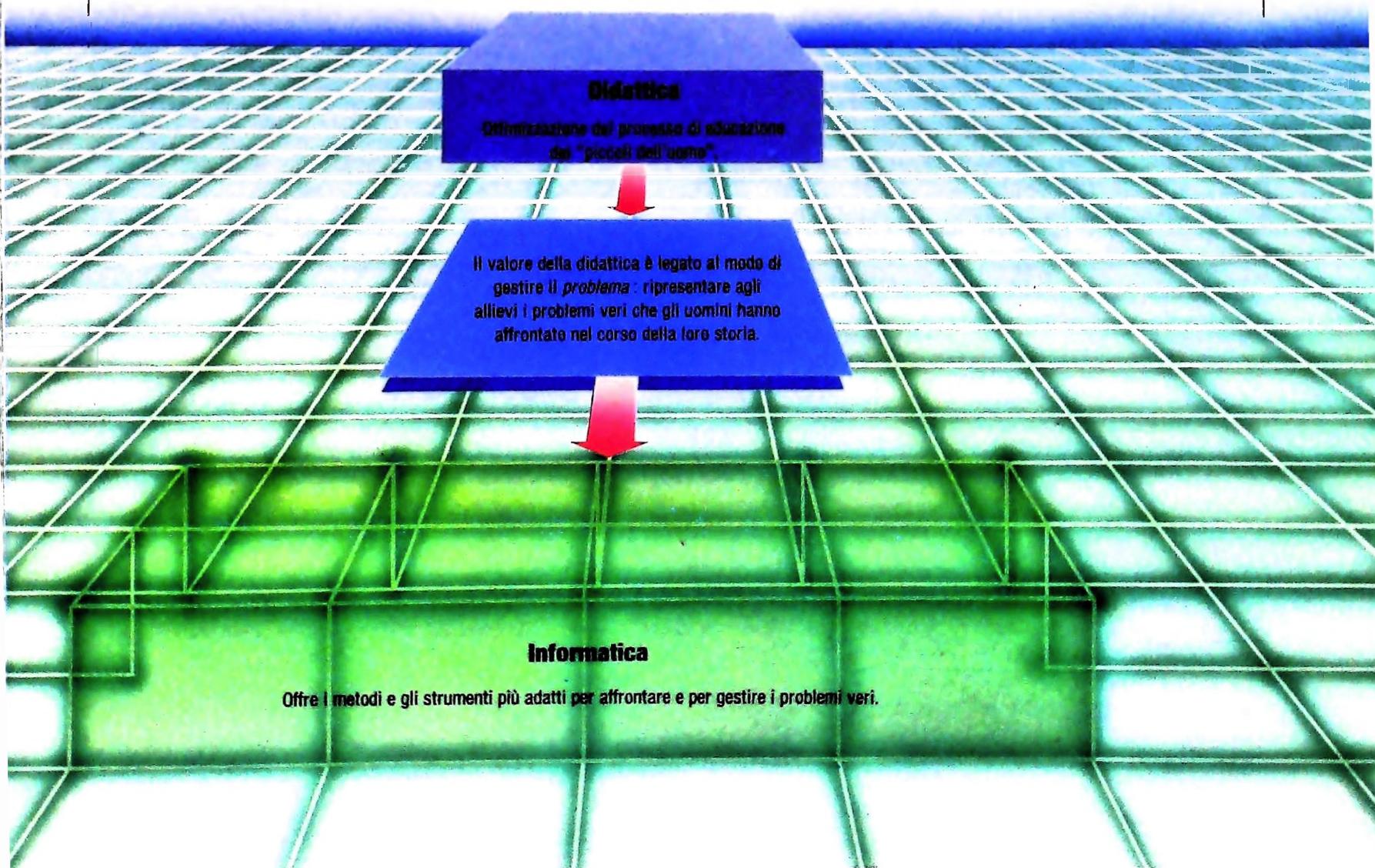
Poi c'è il tema dell'epistemologia, cioè, sempre in generale, il problema dei fondamenti teorici e storici di ogni disciplina. Come cogliere le ragioni profonde che hanno prodotto determinati contenuti, determinate forme di sapere, determinate organizzazioni del sapere stesso; come guardare indietro alle radici delle varie discipline e insieme come guardare ai loro sviluppi futuri.

Senza contare poi gli svariati problemi pratici di metodi e di strumenti.

Il ruolo del problema

Fermiamoci soltanto su un punto, piccolo in apparenza, e tuttavia molto significativo per gli sviluppi legati all'informatica: il ruolo che ha il "problema".

Schematizzando un poco possiamo dire che il valore di una





HEMETT PACKARD

L'impiego del calcolatore come strumento di insegnamento è una situazione che va facendosi sempre più comune, anche

al di fuori dei corsi di studio specificamente o anche parzialmente dedicati all'informatica.

certa didattica dipende essenzialmente dal ruolo e dalla funzione che in essa ha il problema. Il problema infatti nasce dall'emergere di esigenze non soddisfatte, di conflitti, magari di vere e proprie contraddizioni. La storia dell'umanità in generale e di ciascun settore in particolare è costellata di problemi, la cui soluzione ha scandito la storia stessa. Pensiamo al progresso tecnico dalle mille sfaccettature. Pensiamo al progresso scientifico sia in generale sia riferito alle singole discipline.

La didattica deve far ripercorrere agli allievi almeno le tappe principali di questi cammini, ponendo gli allievi di fronte agli stessi problemi che hanno dovuto affrontare gli uomini nelle diverse fasi della storia. I contenuti delle varie discipline verranno visti propriamente come risposte ai problemi stessi, risposte parziali e provvisorie magari, ma utili per risolvere i conflitti e le contraddizioni, per rispondere alle esigenze da cui sono nati.

Crediamo infatti che la vera didattica non consista nel fingere i problemi veri, ma nell'affrontarli creando un rapporto diretto tra la soluzione attuale e le situazioni attraverso cui sono passati coloro che hanno contribuito allo sviluppo dell'umanità. sviluppo che poi si traduce nei contenuti delle discipline che vengono affrontate nella scuola. E l'informatica? La vediamo collocata proprio in questa prospettiva.

Non è stato superfluo essere partiti così da lontano: crediamo infatti che l'informatica per l'insegnamento abbia radici molto profonde, le radici stesse generali della didattica. L'informatica ha veramente qualcosa di nuovo da dire in questo campo. Qualcosa che non si esaurisce in semplici sussidi,

analoghi agli audiovisivi ormai convenzionali, ma che penetra nella struttura delle discipline e le trasforma profondamente. Perché ormai è chiaro che non esiste apparecchiatura più flessibile e più adattabile dell'elaboratore elettronico: con il suo impiego nella didattica possiamo affrontare i problemi più diversi, possiamo simulare situazioni differenti, possiamo inventare metodologie nuove, possiamo gestire anche tutti gli altri sussidi tradizionali.

Diciamo chiaramente che, volendo, attraverso l'informatica, si possono affrontare seriamente i problemi veri, senza fingerli; si possono porre i problemi reali, affrontati dall'umanità nella sua storia. Attenzione però: qui vengono presentate delle prospettive, delle potenzialità. Non è detto che tutto questo sia stato realizzato, né che ciò che si è fatto sia stato fatto bene.

Che cosa faremo

Comunque il calcolatore e il suo impiego costituiscono certamente un importante passo avanti per affrontare i problemi fondamentali della didattica ed è proprio questo che noi vogliamo affermare.

Cercheremo soprattutto di fare il punto sulla situazione, evitando atteggiamenti preconcepiuti pro o contro l'informatica, ma guardando da una posizione serena e possibilmente obiettiva quello che è stato fatto e quello che resta da fare.

Vediamo ora brevemente il panorama del nostro lavoro. Passeremo in rassegna le varie apparecchiature disponibili ed i relativi tipi d'impiego. Approfondiremo poi il discorso sui linguaggi e i prodotti software attualmente usati.

Particolare attenzione sarà prestata alle varie metodologie usate nell'applicazione dell'informatica alla didattica: esercitazioni, simulazioni, istruzione (impostazione *tutoriale*), soluzione di problemi ecc., con i riferimenti opportuni alle varie discipline scolastiche: matematica, fisica, scienze naturali, linguistica e materie umanistiche in genere.

Infine presenteremo l'impiego dell'informatica nel problema della valutazione e più in generale della gestione della classe, attraverso i concetti fondamentali della cibernetica.

Un compito impegnativo il nostro, certo non facile, ma che cercheremo di svolgere avendo particolare attenzione ai più diretti interessati, cioè agli insegnanti. Il nostro discorso infatti va principalmente a loro, affinché acquistino le conoscenze fondamentali di informatica e del relativo impiego nella didattica e sappiano fare le opportune scelte nel campo di apparecchiature e metodi di lavoro.

Parecchio è stato fatto, ma c'è ancora molto da fare. È necessario provare, sperimentare, riflettere sui risultati, rafforzare le esperienze positive; è necessario soprattutto che queste esperienze siano fatte da molti insegnanti, che i confronti siano nutriti, approfonditi, che i risultati positivi vengano diffusi e riprodotti. Tutto sommato crediamo di essere ancora all'inizio dell'esplorazione delle enormi possibilità dell'informatica nel campo didattico: un'esplorazione che presenta certo dei problemi, ma che lascia anche intuire delle grandi possibilità: sta a noi sfruttarle.

Lezione 5

Le iterazioni

Un altro modello ricorrente nella definizione di algoritmi è l'ITERAZIONE: si tratta di ripetere più volte l'esecuzione di una o più istruzioni, come nell'esempio seguente, che visualizza i primi dieci numeri interi:

- predisponi a una variabile che conti le visualizzazioni
- RIPETI
 - visualizza il valore della variabile
 - incrementa la variabile

FINO AL MOMENTO IN CUI la variabile ha superato il valore 10.

Questo nuovo modello di STRUTTURA DI CONTROLLO cambia radicalmente la potenza dei programmi: infatti, mentre con la selezione riusciamo a far sì che un solo programma contenga due sequenze d'esecuzione, con l'iterazione riusciamo a far sì che un programma ne possa contenere infinite! Ad esempio, il programma

- acquisisci un valore
- RIPETI
 - visualizza il valore
 - acquisisci un nuovo valore

FINO AL MOMENTO IN CUI il valore è zero

può causare l'iterazione delle istruzioni controllate dalla struttura RIPETI... FINO A... 1, 2, o un numero qualsivoglia di volte che non è noto a priori e che dipende dalla sequenza dei valori forniti.

Ancora una volta, il modello introdotto è estremamente generale e possiamo ritrovarlo nei più moderni linguaggi di programmazione. In PASCAL diremo:

acquisisci un valore nella variabile X

REPEAT

- visualizza X
- acquisisci un nuovo X

UNTIL X=0

Come al solito, come schema di riferimento ci atterremo alla forma inglese usata dal PASCAL.

Come realizzare iterazioni in Basic

Il linguaggio BASIC non mette a disposizione una struttura di controllo iterativa come il REPEAT... UNTIL..., e noi saremo costretti pertanto a "sintetizzarla" con le istruzioni che già conosciamo.

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

*MEDIT.DO
MEDIP.BA
FATTT.DO
FATTP.BA
SFATT.DO*

contenuti nella cassetta "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.

Così

REPEAT..... UNTIL X=0

può essere realizzato come segue:

```
10 REM Inizio parte da iterare
20 REM .....
30 IF X<>0 THEN 10
```

ovvero, se dopo l'esecuzione di 10 e di 20 la variabile X è uguale a zero, l'istruzione IF non ha effetto e si prosegue con il resto del programma (cioè si interrompe l'iterazione); se invece X è diverso da zero (la coppia "<>" ha proprio il significato di "diverso"), allora si torna all'istruzione 10 e si itera l'esecuzione.

Così, i due programmi precedenti possono essere realizzati in BASIC come segue:

```
10 REM Valore iniziale del contatore
20 LET I=1
30 REM Inizia l'iterazione
40 PRINT I
50 LET I=I+1
60 REM Controllo di fine iterazione
70 IF I<=10 THEN 30
```

Si noti che, se vogliamo terminare l'iterazione al valore di una certa condizione, dobbiamo tornare a iterare al valore della contraria.

E il secondo:

```
10 REM Acquisisci un valore
20 INPUT "X":X
30 REM Inizia l'iterazione
40 PRINT "Valore:":X
50 INPUT "X":X
60 REM Controllo fine iterazione
70 IF X<>0 THEN 30
```

Attenzione: X è usato come comando di fine iterazione e come tale non comparirà tra i valori visualizzati.

L'esecuzione fornirà:

- nel primo caso, la visualizzazione dei numeri da 1 a 10
- nel secondo caso, la visualizzazione di tutti i valori forniti, a eccezione del valore zero che ha l'effetto di interrompere l'iterazione.

Come memorizzare un programma

Finora abbiamo usato il calcolatore per inserirvi un programma per volta, senza mai avere la possibilità di inserirne uno nuovo senza perdere il precedente. In realtà il nostro M10 ha la possibilità di memorizzare programmi, associando a ciascuno di essi un nome, in modo da poterne avere a disposizione più di uno allo stesso tempo.

L'ultimo programma che abbiamo costruito visualizza i valori forniti fino allo zero, e, se non avete dato il comando NEW, si trova ancora a disposizione memorizzato

nell'M 10, come potrete facilmente rendervi conto selezionando il BASIC e chiedendo la lista.

Se non l'avete più, inseritelo nuovamente e fornite poi il seguente comando:

```
SAVE "VISUA"
```

Il programma verrà memorizzato in una zona della memoria e ad esso verrà associato il nome "VISUA". Tornando infatti al menù principale (con il tasto F8), troviamo tra gli elementi selezionabili anche

```
VISUA.BA
```

che corrisponde al nome del nostro programma, a cui l'M 10 ha aggiunto il suffisso

```
.BA
```

per ricordarci che si tratta di un programma BASIC.

Torniamo ora al BASIC e, con il comando NEW, cancelliamo il programma (potremo accertarci che sia stato effettivamente cancellato chiedendone la lista, che non sarà evidenziata).

Se ora ritorniamo al menù principale e selezioniamo VISUA.BA (come al solito, posizionando il cursore e premendo ENTER), il nostro programma partirà subito con l'esecuzione; ad esempio:

```
run
X? 4
Valore: 4
X? 5
Valore: 5
X? 83
Valore: 83
X? 0

Ok
```

Alla fine dell'esecuzione, potremo considerarlo a tutti gli effetti un programma BASIC, chiedendone la lista, o eseguendolo, o modificandolo: questa volta però le modifiche risulteranno già memorizzate in VISUA.BA.

Ad esempio, aggiungendo il commento

```
1 REM Testo di VISUA.BA
```

spegnendo il calcolatore, poi riaccendendolo e selezionando VISUA.BA, possiamo ottenere la lista del programma, che ci mostra di avere mantenuto l'alterazione:

```
1 REM Testo di VISUA.BA
10 REM Acquisisci un valore
20 INPUT "X":X
30 REM Inizia l'iterazione
40 PRINT "Valore:":X
```

```
50 INPUT "X":X
60 REM Controllo fine iterazione
70 IF X<>0 THEN 30
```

La cancellazione di programmi

Supponiamo ora di voler cancellare programmi che abbiamo precedentemente memorizzati e che non sono più di nostro interesse: il comando BASIC che permette di fare questo è

```
KILL
```

(che in inglese significa, appunto "uccidi").

Così, torniamo al menù principale e selezioniamo il BASIC.

Diamo il comando

```
FILES
```

che permette di evidenziare sullo schermo dell'M 10 tutti i programmi o i testi memorizzati (FILE in inglese indica una "sfilza" di informazioni, che possono essere pensate come programmi o in generale come testi, e il comando ce ne evidenzia i nomi): sullo schermo compare il nostro VISUA.BA.

Con il comando

```
KILL "VISUA.BA"
```

otteniamo la risposta "Ok" che indica l'avvenuta distruzione del programma, come potremo vedere tornando al menù principale, in cui non appare più.

Cosa abbiamo imparato

Nella presente lezione abbiamo visto:

- l'ITERAZIONE come classe di strutture di controllo
- la struttura REPEAT... UNTIL...
- la realizzazione di REPEAT... UNTIL... in BASIC
- il comando SAVE per salvare in memoria programmi
- il comando FILES per evidenziare, mentre abbiamo selezionato il BASIC, quali "files" abbiamo in memoria
- il comando KILL per cancellare un programma o un testo.

SCALE MUSICALI

Impariamo a generare scale diverse per caratterizzare meglio la nostra musica.

Abbiamo già accennato, parlando della notazione delle altezze, ai rapporti tra le frequenze corrispondenti alle altezze *do, re* ecc. Questi rapporti sono detti *intervalli*; quando sentiamo una sequenza di altezze (una *melodia*) o alcune altezze sovrapposte (un *accordo*), oltre al valore *assoluto* in frequenza delle singole altezze, abbiamo anche la facoltà di percepire i rapporti tra le differenti altezze, cioè gli intervalli; a seconda del rapporto, infatti, una certa sequenza (o una certa sovrapposizione) assume la caratteristica di *consonanza* o *dissonanza*; in altri termini, ci appare gradevole o sgradevole.

Gli intervalli

In prima approssimazione, diciamo che un intervallo è tanto più *consonante* quanto più sono piccoli i numeri del rapporto che definiscono l'intervallo stesso. Ad esempio, il rapporto tra due *do* di due ottave successive è 1:2 ed è quindi il più consonante; il rapporto tra il *do* ed il *si* della medesima ottava è 8:15, quindi è più dissonante. Sulla base di questo crite-

rio possiamo creare una gerarchia dei rapporti intervallari tra una nota e le altre: l'intervallo *do-do* abbiamo detto che è il più consonante (1:2, viene detto intervallo d'*ottava*), *do-sol* (2:3, intervallo di *quinta*), *do-fa* (3:4, intervallo di *quarta*) ecc. L'uso del concetto di intervallo ci permette di definire strutture musicali che si riferiscono a un insieme di rapporti intervallari prefissati; queste strutture vengono chiamate *scale*; se gli intervalli di una scala sono ascendenti, cioè fanno passare da una altezza più grave ad una più acuta, chiamiamo la scala *ascendente*; viceversa, se gli intervalli fanno passare ad altezze più gravi chiamiamo la scala *discendente*.

Questo tipo di impostazione porta alla definizione della *scala naturale*. Nella scala naturale gli intervalli sono definiti sulla base di rapporti tra numeri interi, come diretta conseguenza della teoria fisica degli armonici; gli strumenti antichi sono capaci di suonare i suoni di una o più scale naturali. Il limite che deriva dall'adozione della scala naturale è che non possiamo eseguire la trasposizione di un brano musicale sugli strumenti per i quali il brano è stato originariamente concepito, ma dobbiamo utilizzare strumenti *intonati* secondo la

John Cage (nato a Los Angeles nel 1912) è uno dei maggiori rappresentanti della nuova musica: ne ha esplorato tutti i settori, dall'elettronica alla musica concreta, abbracciando in particolare il concetto di "aleatorietà", di totale casualità nell'esecuzione (e nella composizione) musicale



scala naturale corrispondente alla *tonalità* (vedremo più avanti in dettaglio questo concetto) in cui trasponiamo il brano. In altre parole, l'insieme di altezze a disposizione su uno strumento basato sulla scala naturale dipende da un valore di altezza iniziale rispetto al quale sono stati calcolati i rapporti intervallari della scala.

La scala temperata

Il rimedio a questo inconveniente è stato l'introduzione di strumenti musicali basati sulla *scala temperata*, cioè una scala in cui gli intervalli sono definiti sulla base di un rapporto costante tra una altezza e la successiva altezza della scala, e cioè la radice dodicesima di due (cioè 1,059463). In questo modo dopo dodici intervalli si ottiene l'intervallo di ottava. Le altezze di questa nuova scala più vicine alle altezze della scala naturale vengono usualmente identificate con quelle della scala naturale ed è così possibile combinare molto più facilmente strumenti diversi con elaborazioni del testo musicale che implicano la trasposizione delle altezze (cioè l'abbassamento o l'innalzamento dei valori di altezza delle note di un brano musicale).

L'elaboratore ci permette di impostare il nostro mondo musicale così come lo preferiamo, purché specifichiamo le nostre esigenze; così possiamo compiere le nostre elaborazioni musicali sia basandoci sulla scala naturale che sulla scala temperata; in conseguenza di questa scelta le operazioni musicali avranno una forma e uno svolgimento differente. Vediamo un esempio di definizione secondo le due differenti scale. Usando l'istruzione SOUND possiamo realizzare una scala naturale mediante il seguente programma BASIC:

```
010 INPUT "DURATA:"; Y
020 INPUT "ALTEZZA INIZIALE DELLA SCALA:"; X
030 SOUND X,Y           REM unisono
040 SOUND X*24/25,Y     REM seconda minore
050 SOUND X*8/9,Y       REM seconda maggiore
060 SOUND X*5/6,Y       REM terza minore
070 SOUND X*4/5,Y       REM terza maggiore
080 SOUND X*3/4,Y       REM quarta
090 SOUND X*18/25,Y     REM quarta eccedente
100 SOUND X*2/3,Y       REM quinta
110 SOUND X*5/8,Y       REM sesta minore
120 SOUND X*3/5,Y       REM sesta maggiore
130 SOUND X*5/9,Y       REM settima minore
140 SOUND X*8/15,Y      REM settima maggiore
150 SOUND X/2,Y         REM ottava
160 END
```

Come commento (REM) è indicato il nome dell'intervallo. In questo modo otteniamo una scala ascendente; invertendo i rapporti otteniamo una scala discendente (24/25 diventa 25/24, 8/9 diventa 9/8 ecc.). Notiamo che poiché il primo parametro dell'istruzione SOUND è inversamente proporzionale all'effettiva altezza del suono, in realtà compiamo l'operazione inversa rispetto a quella che faremmo su un valore di

frequenza. Se, nel programma dell'esempio, assegnassimo al parametro X un valore corrispondente a un *do* (4697 o 2348 o 9394 ...) otterremmo la sequenza delle dodici altezze fondamentali della scala naturale calcolata nella tonalità di *do*. Per generare la scala temperata possiamo invece usare un programma come il seguente:

```
030 Z=1.059463
040 SOUND X,Y
050 SOUND X/Z,Y
060 SOUND X/Z^2,Y
070 SOUND X/Z^3,Y
080 SOUND X/Z^4,Y
090 SOUND X/Z^5,Y
100 SOUND X/Z^6,Y
110 SOUND X/Z^7,Y
120 SOUND X/Z^8,Y
130 SOUND X/Z^9,Y
140 SOUND X/Z^10,Y
150 SOUND X/Z^11,Y
160 SOUND X/2,Y
170 END
```

Con questo programma otteniamo la scala temperata ascendente; otteniamo quella discendente sostituendo il segno di divisione / con il segno di moltiplicazione *. Un modo più rapido per descrivere la generazione di una scala temperata può essere però anche il seguente:

```
030 Z=1.059463
040 SOUND X,Y
050 FOR I=1 TO 12
060 X=X/Z
070 SOUND X,Y
080 NEXT I
090 END
```

In questo secondo modo, tra l'altro, richiediamo una minore quantità di calcoli all'elaboratore, cioè dodici divisioni invece di dieci elevazioni a potenza più dodici divisioni.

È abbastanza immediato a questo punto realizzare qualunque tipo di *temperamento* dell'ottava desiderata. Nella musica moderna, ad esempio, sono spesso usati i terzi di tono e i quarti di tono. Per ottenere questi valori di altezza bisognerà dividere l'ottava rispettivamente in 18 e 24 intervalli uguali tra loro e quindi usare come *rapporto intervallare elementare* la radice diciottesima di due e la radice ventiquattresima di due rispettivamente. Il nostro programma diventa quindi il seguente, nel caso dei terzi di tono:

```
030 Z=1.039259
040 SOUND X,Y
050 FOR I=1 TO 18
060 X=X/Z
070 SOUND X,Y
080 NEXT I
090 END
```

e per i quarti di tono:

```
030 Z=1.029302
040 SOUND X,Y
050 FOR I=1 TO 24
060 X=X/Z
070 SOUND X,Y
080 NEXT I
090 END
```

Possiamo quindi sbizzarrirci a costruire scale; le scale che costruiremo saranno il nostro alfabeto di altezze e di intervalli; sulla base di questo alfabeto costruiremo il nostro mondo musicale.

Scala diatonica, pentatonica e a toni interi

Vediamo come possiamo realizzare alcune tra le scale più usate. Cominciamo con la *scala diatonica*, la scala che è alla

base della musica tonale nel *modo maggiore* (discuteremo il concetto di modo maggiore più avanti):

```
030 Z=1.059463
040 SOUND X,Y
050 X=X/Z^2
060 SOUND X,Y
070 X=X/Z^2
080 SOUND X,Y
090 X=X/Z
100 SOUND X,Y
110 X=X/Z^2
120 SOUND X,Y
130 X=X/Z^2
140 SOUND X,Y
150 X=X/Z^2
160 SOUND X,Y
170 X=X/Z
180 SOUND X,Y
190 END
```

REM tonica
REM seconda
REM terza maggiore
REM quarta
REM quinta
REM sesta
REM settima
REM ottava



Pietro Grossi al lavoro sulle apparecchiature del CNUCE (Centro Nazionale Universitario di Calcolo Elettronico), di cui dal 1973 dirige la sezione musicale. Nato a Venezia nel 1917, compositore e violoncellista, Grossi è stato fra i primi in Italia a interessarsi al calcolatore come "strumento" per la musica, per l'esecuzione così come per la creazione musicale.

Come si può vedere dalle operazioni compiute su X, la scala diatonica è formata dalla sequenza di intervalli tono-tono-semitono-tono-tono-tono-semitono in cui *tono* è l'intervallo doppio del *semitono*, che a sua volta corrisponde al rapporto intervallare ottenuto mediante moltiplicazione per la radice dodicesima di due; un semitono è l'intervallo tra un tasto e il successivo del pianoforte (considerando sia i tasti bianchi che i tasti neri).

Un'altra scala molto usata nella musica popolare è la *scala pentatonica*, costituita da un sottoinsieme di altezze della scala diatonica.

Un programma con il quale otteniamo la scala pentatonica è il seguente:

```
030 Z=1.059463
040 SOUND X,Y
050 X=X/Z^2
060 SOUND X,Y
070 X=X/Z^2
080 SOUND X,Y
090 X=X/Z^3
100 SOUND X,Y
110 X=X/Z^2
120 SOUND X,Y
130 X=X/Z^3
```

```
140 SOUND X,Y
150 END
```

Questa scala è costituita dalla sequenza tono-tono-tre semitoni-tono-tre semitoni. Vediamo come ottenere la *scala per toni interi* (*esatonale*), molto usata ad esempio da Debussy:

```
030 Z=1.059463
040 Z=Z^2
050 SOUND X,Y
060 FOR I=1 TO 6
070 X=X/Z
080 SOUND X,Y
090 NEXT I
100 END
```

Nell'illustrazione è mostrata la notazione su pentagramma delle scale che abbiamo imparato a generare. L'adozione di una scala musicale piuttosto che un'altra caratterizza fortemente la musica che possiamo comporre. I prossimi argomenti che affronteremo riguarderanno appunto gli aspetti della sintassi musicale che più direttamente sono influenzati dalla scala a cui facciamo riferimento: la melodia, l'accordo e i modi (con particolare attenzione al modo maggiore e al modo minore).

Qui sotto, la notazione su pentagramma delle scale musicali generate mediante i programmi in BASIC che sono stati descritti nel testo.

scala naturale e scala temperata (notazione identica)

scala diatonica

scala pentatonica

scala per toni interi

scala per quarti di tono

ISTRUZIONI PRESET, PSET

Due nuove istruzioni grafiche per attivare e disattivare i pixel dallo schermo.

Per affrontare il problema della grafica pittorica dobbiamo prima di tutto vedere come si può rappresentare un'immagine scomposta per punti. Questa descrizione ci aiuterà a comprendere le basi concettuali di una vastissima branca della grafica: l'elaborazione di immagini. Si tratta di un campo che apre una grande varietà di applicazioni, che vanno dallo studio del territorio all'analisi di dati medici, dalla simulazione di volo al riconoscimento di scene e movimenti.

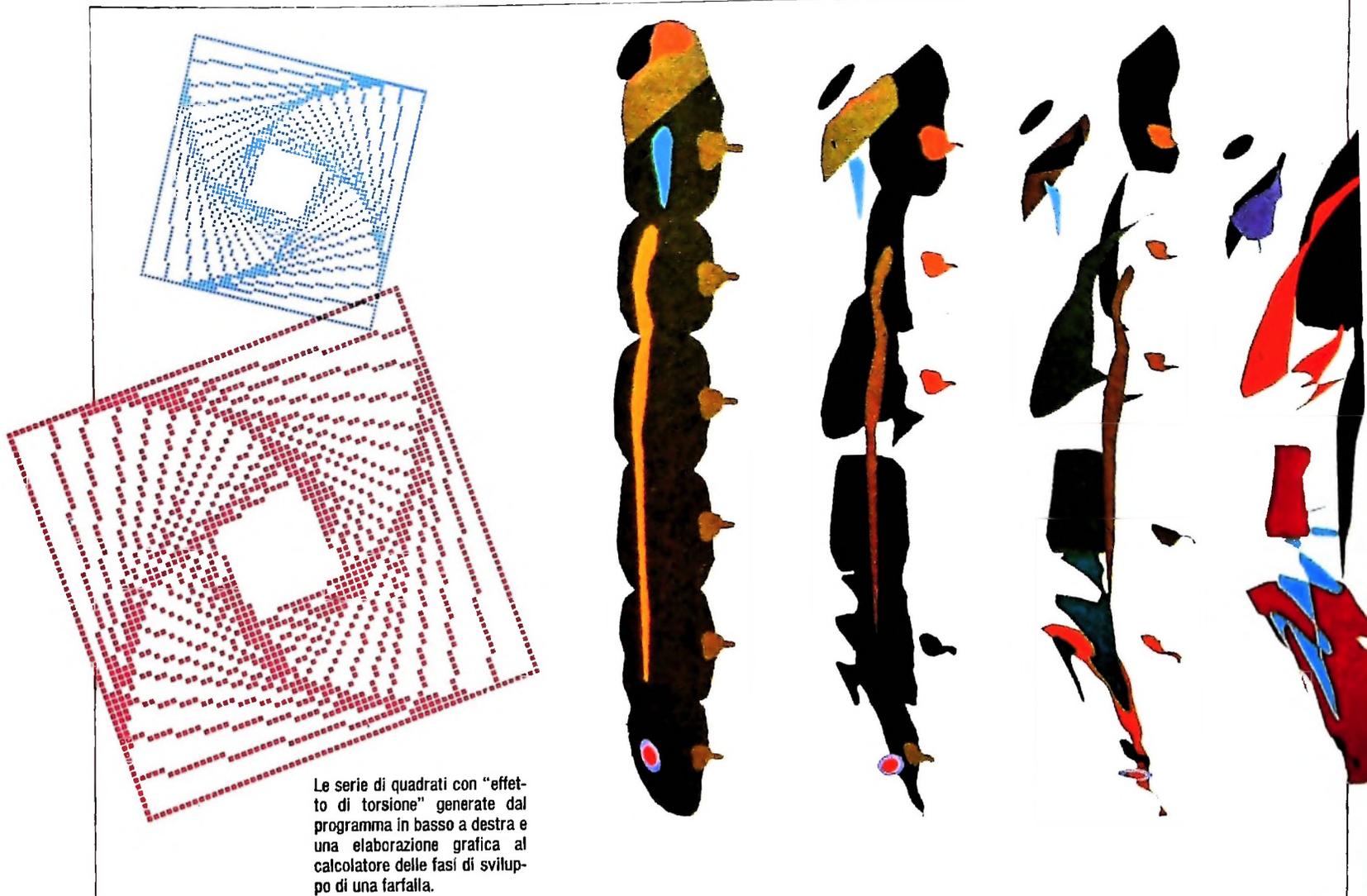
Al livello più semplice un'immagine può venire rappresentata come un insieme di punti distribuiti uniformemente su un foglio, i quali possono assumere due colori: bianco o nero. Questo è ciò che succede ad esempio nella stampa delle fotografie su un quotidiano: la fotografia (provare con una lente per credere!) viene rappresentata con una fitta maglia di punti neri; i punti sono più densi nelle zone in cui l'immagine appare nera e molto radi o del tutto assenti nelle zone



Rappresentazione grafica a colori della soluzione di un'equazione matematica.

P 5 8 3

ACM ASSOCIATION - ARCHIVIO EDOSS



Le serie di quadrati con "effetto di torsione" generate dal programma in basso a destra e una elaborazione grafica al calcolatore delle fasi di sviluppo di una farfalla.

bianche. I grigi d'altra parte vengono realizzati con densità di punti variabili.

Lo schermo del computer si suddivide in *pixel* (*picture cells*, unità elementari d'immagine): se un pixel è attivo abbiamo un punto nero, se è disattivato abbiamo il fondo bianco. Per attivare e disattivare i pixel disponiamo di due istruzioni, equivalenti e complementari: PRESET e PSET.

Provate a eseguire le istruzioni:

```
10 CLS
20 PRESET (120,32,0)
```

Apparirà un punto al centro dello schermo. Se invece eseguite le istruzioni:

```
10 CLS
20 PSET (120,32,0)
```

non apparirà alcun punto. Le due istruzioni vengono cioè interpretate in modo opposto dal computer per quanto riguarda l'ultimo parametro, che, se vale 0, per la PRESET attiva il pixel e per la PSET lo disattiva, mentre, se vale 1, per la PRESET disattiva il punto e per la PSET lo attiva.

Quadrati in serie

Vediamo qui un programma che disegna tre serie di 14 quadrati concentrici e ruotati rispetto agli assi, in modo tale da generare un effetto visivo di torsione.

```
10 CLS
20 FOR J=1 TO 3
30 X1=28+60*(J-1)
40 X2=88+60*(J-1)
50 X3=88+60*(J-1)
60 X4=28+60*(J-1)
70 Y1=2
80 Y2=2
90 Y3=62
100 Y4=62
110 FOR I=1 TO 14
120 LINE (X1,Y1)-(X2,Y2)
130 LINE (X2,Y2)-(X3,Y3)
140 LINE (X3,Y3)-(X4,Y4)
```



```

150 LINE (X4,Y4)-(X1,Y1)
160 X1=.9*X1+.1*X2
170 X2=.9*X2+.1*X3
180 X3=.9*X3+.1*X4
190 X4=.9*X4+.1*X1
200 Y1=.9*Y1+.1*Y2
210 Y2=.9*Y2+.1*Y3
220 Y3=.9*Y3+.1*Y4
230 Y4=.9*Y4+.1*Y1
240 NEXT I
250 NEXT J
260 END

```

Commento al programma

L'istruzione 10 serve per ripulire lo schermo. La 20 e la 250 costituiscono un ciclo di FOR che serve per ottenere le 3 ripetizioni dell'immagine. Dalla 30 alla 100 si calcolano le coordinate dei pixel che costituiscono i vertici dei tre quadrati più esterni. Si noti che le coordinate verticali, cioè le Y, di tali quadrati rimangono sempre inalterate a ogni incremento della J. Questo perché il disegno si sviluppa in orizzontale e, fra una serie e l'altra, le coordinate iniziali che cambiano sono soltanto quelle delle X. La 110 e la 240 costituiscono un secondo ciclo di FOR, nidificato nel precedente, che ha lo scopo di

disegnare, mediante le istruzioni dalla 120 alla 150, per 14 volte un quadrato. Le coordinate dei vertici sono ogni volta calcolate dalle istruzioni che vanno dalla 160 alla 230. I fattori moltiplicativi .9 e .1 consentono di ottenere un effetto di rotazione e di riduzione dei quadrati. Possiamo allora osservare che variando i valori della J, cioè l'istruzione 20, si possono ottenere più serie uguali dello stesso disegno. Bisognerà però avere l'accortezza di ridurre le dimensioni del quadrato iniziale (dalla 30 alla 100), affinché le dimensioni del disegno complessivo siano compatibili con quelle dello schermo.

Semicerchio con sfumatura

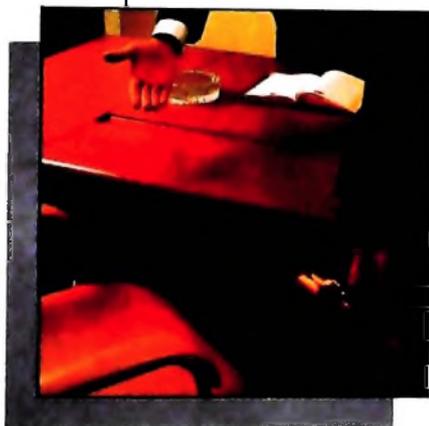
Forniamo un programma che ci permette di realizzare un semicerchio in cui è stato inserito un effetto di sfumatura. Il programma lavora esclusivamente con le istruzioni FOR, NEXT, LINE e PRESET. Il dise-

gno viene eseguito selezionando ogni singolo pixel che si vuole illuminare, mediante le sue coordinate. Non aspettatevi grandi risultati da questo esercizio: costituisce solo un buon allenamento.

```
10 CLS
20 LINE (136,28)-(136,36)
30 LINE (135,26)-(135,38)
40 LINE (134,24)-(134,40)
50 LINE (133,22)-(133,42)
60 LINE (132,21)-(132,27)
70 LINE (132,43)-(132,37)
80 FOR I=35 TO 29 STEP -2
90 PRESET (132,I,0)
100 NEXT I
110 LINE (131,20)-(131,25)
120 LINE (131,43)-(131,39)
130 FOR I = 36 TO 28 STEP -2
140 PRESET (131,I,0)
150 NEXT I
160 FOR I = 31 TO 33 STEP 2
170 PRESET (133,I,1)
180 NEXT I
190 LINE (130,19)-(130,24)
200 LINE (130,41)-(130,44)
210 FOR I= 27 TO 37 STEP 2
220 PRESET (130,I,0)
230 NEXT I
240 PRESET (130,38,0)
250 PRESET (130,26,0)
260 FOR I = 34 TO 30 STEP -2
270 PRESET (129,I,0)
280 NEXT I
290 LINE (129,18)-(129,22)
300 LINE (129,24)-(129,25)
310 LINE (129,27)-(129,28)
320 LINE (129,36)-(129,37)
330 LINE (129,39)-(129,40)
340 LINE (129,42)-(129,45)
350 LINE (128,18)-(128,22)
360 LINE (128,43)-(128,46)
370 PRESET (128,23,0)
380 PRESET (128,41,0)
390 LINE (128,39)-(128,38)
400 LINE (128,25)-(128,26)
410 FOR I=29 TO 35 STEP 2
420 PRESET (128,I,0)
430 NEXT I
440 LINE (127,18)-(127,20)
450 LINE (127,44)-(127,46)
460 FOR I=24 TO 30 STEP 3
470 PRESET (127,I,0)
480 NEXT I
490 FOR I=34 TO 40 STEP 3
500 PRESET (127,I,0)
510 NEXT I
520 FOR I=22 TO 42 STEP 11
530 PRESET (127,I,0)
540 NEXT I
550 PRESET (126,21,0)
560 PRESET (126,23,0)
570 PRESET (126,25,0)
580 PRESET (126,28,0)
590 PRESET (126,31,0)
600 PRESET (126,33,0)
610 PRESET (126,36,0)
620 PRESET (126,39,0)
630 PRESET (126,41,0)
640 PRESET (126,43,0)
650 LINE (126,17)-(126,19)
660 LINE (126,45)-(126,47)
670 FOR I= 22 TO 42 STEP 6
690 PRESET (125,I,0)
700 NEXT I
710 LINE (125,17)-(125,19)
720 LINE (125,45)-(125,47)
730 LINE (124,17)-(124,18)
740 LINE (124,46)-(124,47)
750 END
```

Banca Aperta

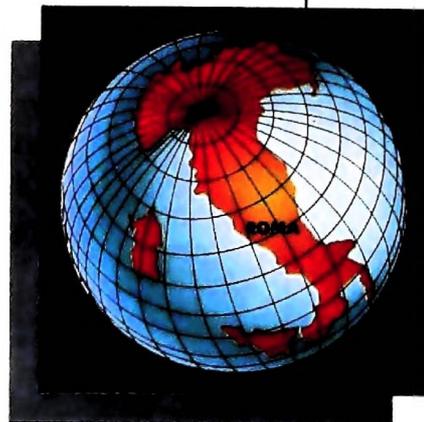
LE NUOVE RISPOSTE DEL BANCO DI ROMA.



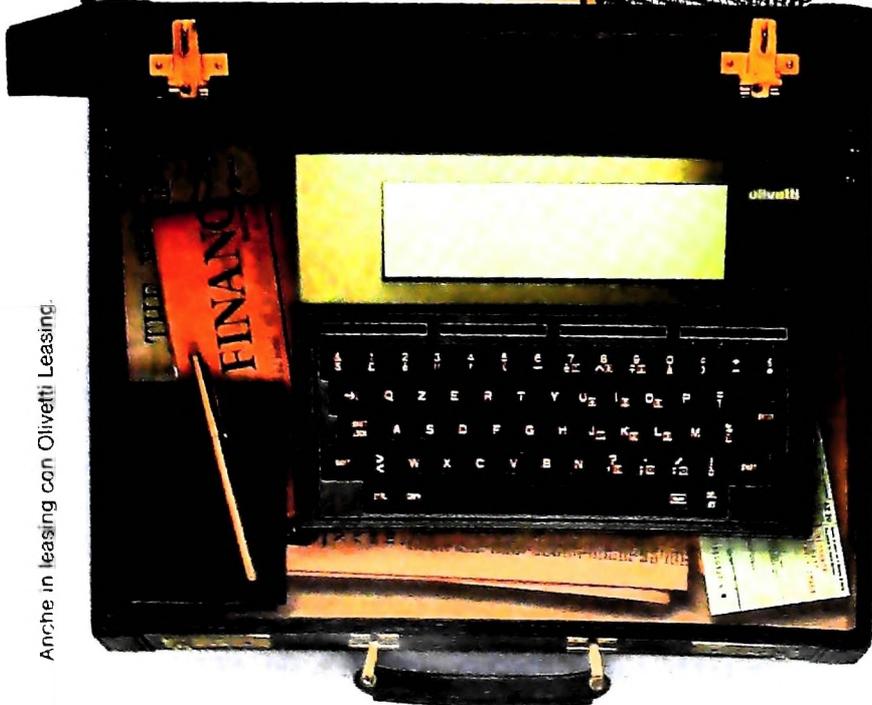
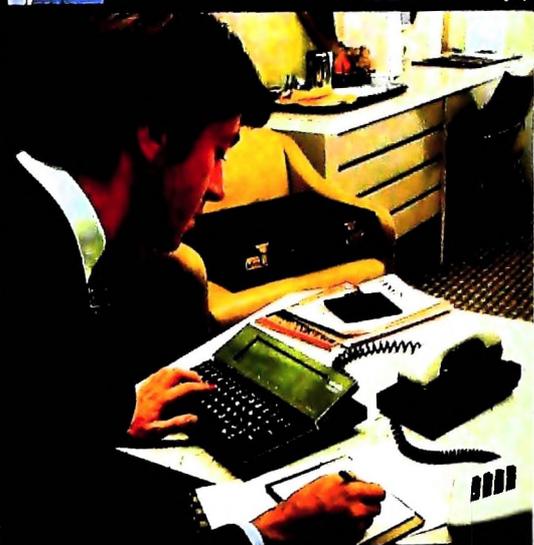
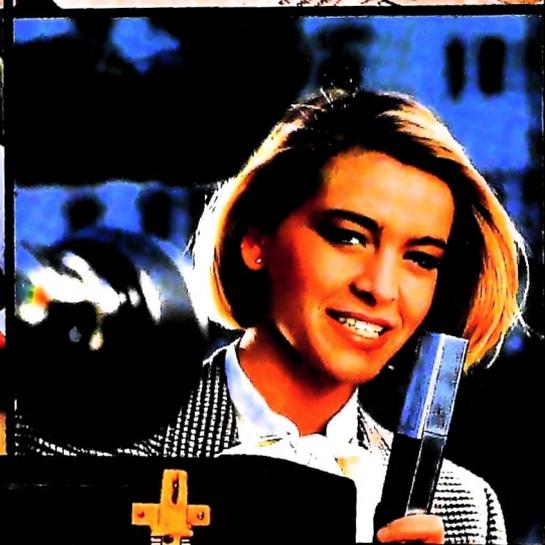
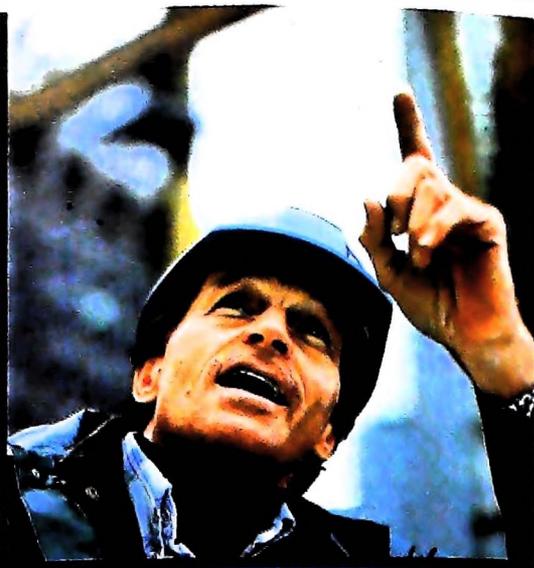
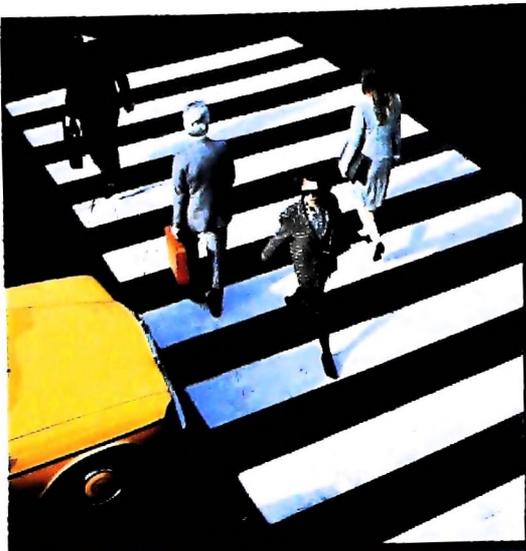
*Vorrei avere
un rapporto più diretto
con la mia banca...*

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 ventiquattre. 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.

olivetti

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

NOME/COGNOME

VIA/N°

CAP/CITTA'

TELEFONO