

9 CORSO PRATICO COL COMPUTER

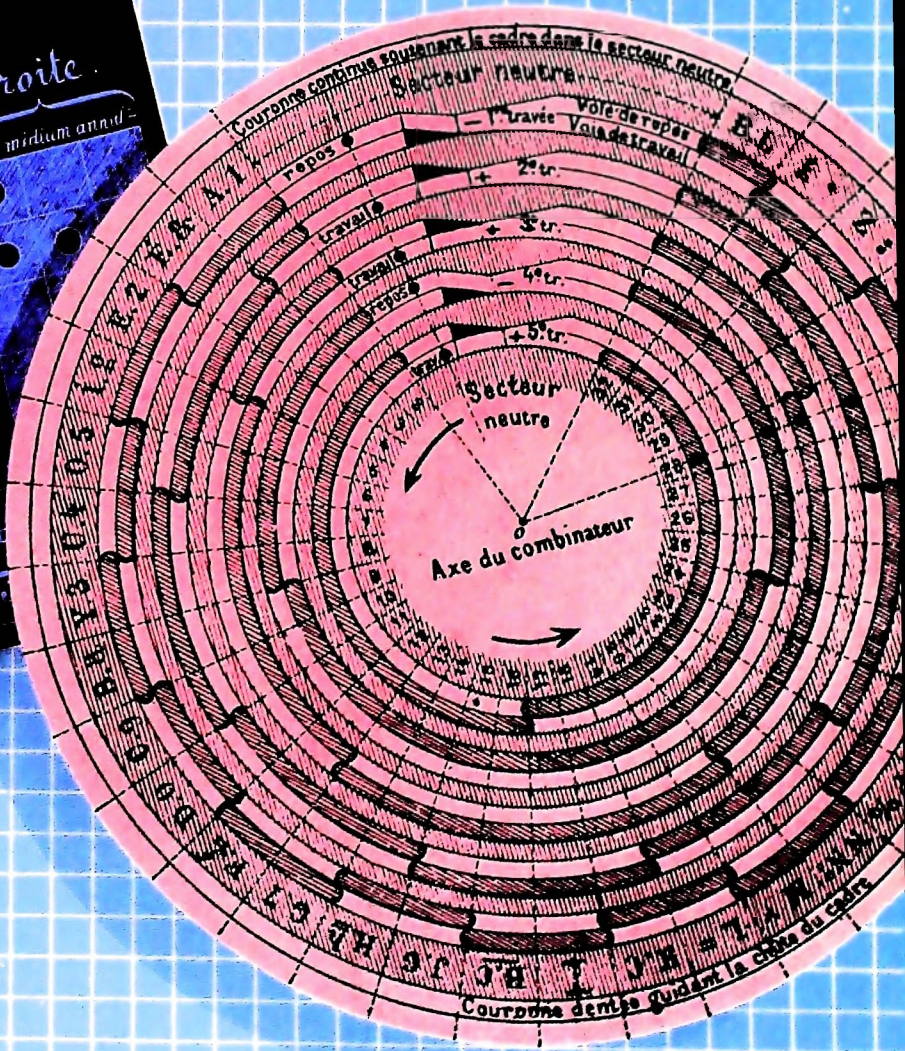
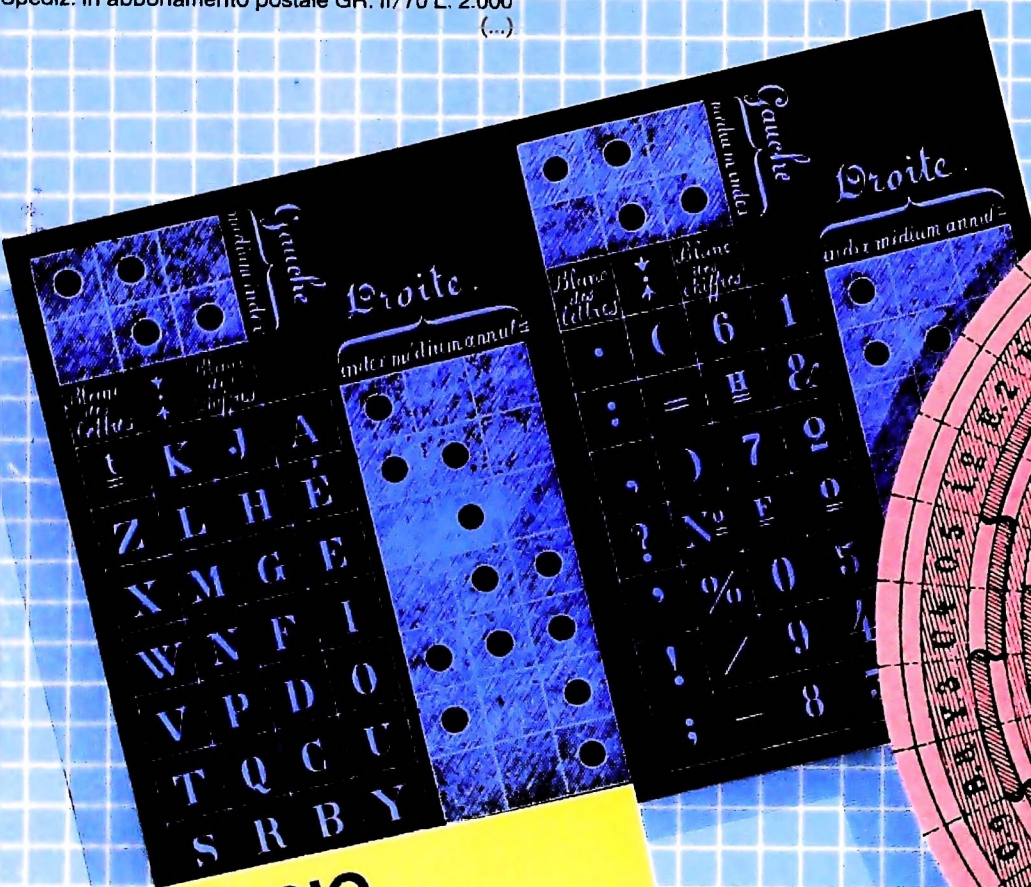
è una iniziativa
FABBRI EDITORI

in collaborazione con
BANCO DI ROMA

e **OLIVETTI**

diretto da **GIANNI DEGLI ANTONI**

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



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

**FABBRI
EDITORI**

IN "ONDA" SU RADIOTEXT I PROGRAMMI DEL CORSO PRATICO COL COMPUTER

Venerdì 11 maggio ha preso il via, su Radio 3, "Radiotext, una radio dentro il computer", una serie di otto trasmissioni, realizzate da Arcimedia — lega informatica —, promosse da Radio 3 e dalla divisione ricerca e sperimentazione della RAI.

Radiotext manda in onda, tra le altre cose, una serie di programmi software di tipo soprattutto non professionale, direttamente fruibili da parte dei possessori di Home-computer più registratore a cassetta più una normale radio.

Gli impulsi trasmessi sono compatibili con Olivetti M10, Spectrum Sinclair 48K, Commodore CBM 64.

Le trasmissioni vanno in onda ogni venerdì e domenica alle ore 16.

Venerdì 25 maggio verrà trasmesso del software già pubblicato dal "Corso pratico col computer" della Fabbri Editori. Poiché la trasmissione tende a promuovere condizioni di ascolto collettivo, proponiamo ai lettori di questa pubblicazione di cogliere l'occasione per ritrovarsi presso le sedi sottoelencate per conoscersi e confrontarsi su problemi ed esperienze in campo informatico. I gruppi d'ascolto nelle sedi della lega informatica Arci.

Presso queste stesse sedi attrezzate con computer previsti dall'iniziativa Fabbri, sarà possibile esercitarsi con le cassette proposte dal "Corso pratico col computer".

- MILANO — via P. Sottocorno 46
- SAVONA — via N. Sauro 3/8
- GENOVA — via Sottoripa 1/B
- PADOVA — Riviera Tito Livio 29
- RIMINI — via Varthema 2
- RAVENNA — via Castel S. Pietro 71
- REGGIO EMILIA — viale Isonzo 72
- ANCONA — via Goito 3
- FIRENZE — via Reginaldo Giuliani 250
- ROMA — via Otranto 18
- NOCERA INFERIORE — via Cucci 83
- SALERNO — via Carmine 85
- TERAMO — via G. D'Annunzio 69
- BARI — Largo Adua 24
- TARANTO — via Dante 66 B
- MATERA — recinto privato La Torre 8
- CAGLIARI — via Fara 7
- NAPOLI — via S. Biagio dei Librai 46
- REGGIO CALABRIA (CATONA) — via Garibaldini 42

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 MAIOCCCHI
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 del Calcolatori all'Università Autonoma Metropolitana di Città del Messico)
GROFFREDO HAUS, MARCO MAIOCCCHI, DANIELE MARINI, GIANCARLO MAURI, CLAUDIO PARPELLI, ENNIO PROVERA

Testi
Eidos (TIZIANO BRUGNETTI), ADRIANO DE LUCA, ENNIO PROVERA, LUC SPAMPINATO, 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. - 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 BREGGÉ

Segretarie di Redazione
RENATA FRIGOLI
LUCIA MONTANARI

AVVISO AI LETTORI

La prossima settimana sarà in edicola la copertina per rilegare il primo volume del "Corso pratico col computer". Prenotate la dal vostro edicolante.

Corso Pratico col Computer - Copyright © sul fascicolo 1984 Gruppo Editoriale Fabbri, Bompiani, Sordogno, Etas S.p.A., Milano - Copyright © sull'opera 1984 Gruppo Editoriale Fabbri, Bompiani, Sordogno, Etas S.p.A., Milano - Prima Edizione 1984 - Direttore responsabile GIOVANNI GIOVANNINI - Registrazione presso il Tribunale di Milano n. 135 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. 9 - 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.

IL LINGUAGGIO DI MACCHINA

La rappresentazione dei numeri in codice binario, esadecimale e decimale.

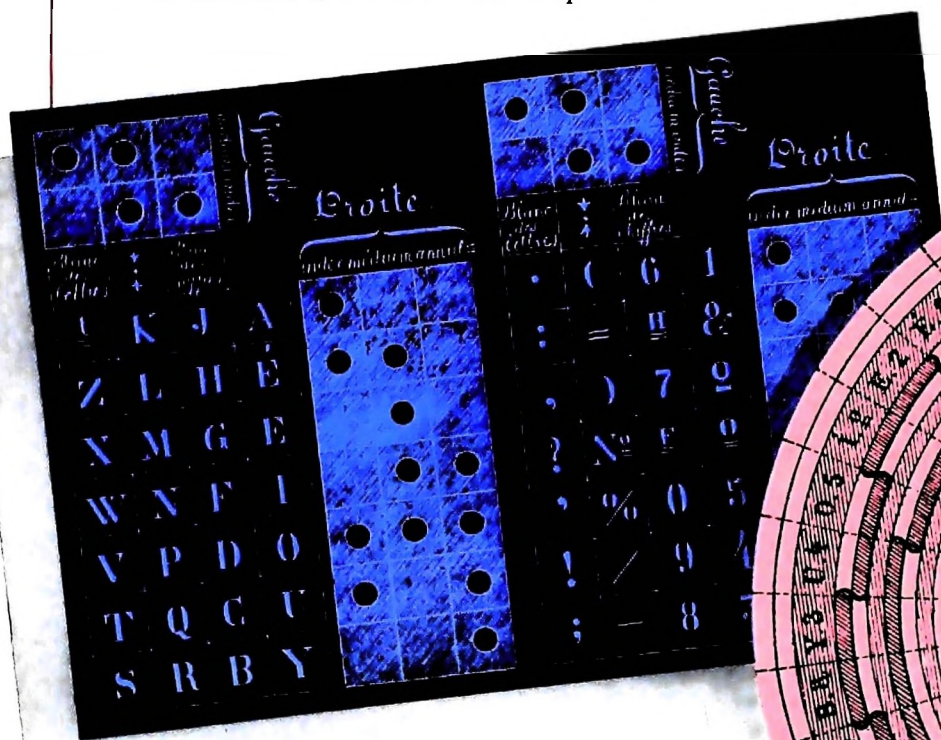
Daremo un'occhiata, anche se molto sommaria, al "sistema numerico" usato nei calcolatori. Prima di tutto cominciamo a definire il concetto di "radice" o "base" di un sistema numerico. La base di un sistema numerico (termine che useremo d'ora in poi) è il numero di simboli permessi nel sistema. In un sistema numerico di base 2 (Tabella 1), gli unici simboli permessi nel sistema sono due, cioè "zero" e "uno", mentre in un sistema con base 10 i simboli permessi vanno dallo 0 al 9 e, infine, in un sistema esadecimale i numeri permessi sono sedici e vanno dallo 0 a F (Tabella 2).

I sistemi che prenderemo in considerazione come i più rispondenti alle nostre esigenze rispetto all'hardware saranno quindi il decimale, per ovvi motivi, l'esadecimale e, internamente al calcolatore, il binario. Comunque non c'è alcuna

difficoltà nel passaggio da una rappresentazione numerica a un'altra, in quanto vi sono varie formule che permettono una rapida trasformazione.

Tutti i calcolatori digitali lavorano internamente con il sistema binario; l'unica cosa che varia è il numero dei bit usati per formare una "parola" come pure il valore che essi rappresentano.

Per esempio, nella rappresentazione binaria (Tabella 3), la prima colonna da sinistra ha valore (o "peso") 8, la seconda 4, la terza 2 e l'ultima 1, per cui il corrispondente numero decimale si ricava sommando i valori dei pesi in corrispondenza agli 1 presente nelle parole binarie. Il numero decimale 5, per esempio, corrisponde al numero binario 0101: la somma dei "pesi" corrispondenti agli 1 è infatti pari a 5.



Gli inizi del codice binario. Il telegrafo di Baudot (1878) usava un codice a permutazione ciclica (chiamato codice Gray negli USA). In alto, la tastiera trasmittente, a lato, la ruota ricevente. Sulla tastiera, ogni tasto vuoto rappresenta uno zero, e ogni tasto con cerchio un uno. Per trasmettere la lettera F, a esempio, il codice era 01 per la mano sinistra e 110 per la

mano destra. Alla ricezione, venivano introdotte le cifre binarie e veniva fatta ruotare la ruota fino a raggiungere un angolo in cui l'elemento sotto la testina di lettura era uguale a quello trasmesso. La ruota era collegata direttamente a un disco scrivente e, quando appariva l'elemento ricevuto, il disco era messo in moto e stampava il carattere trasmesso.

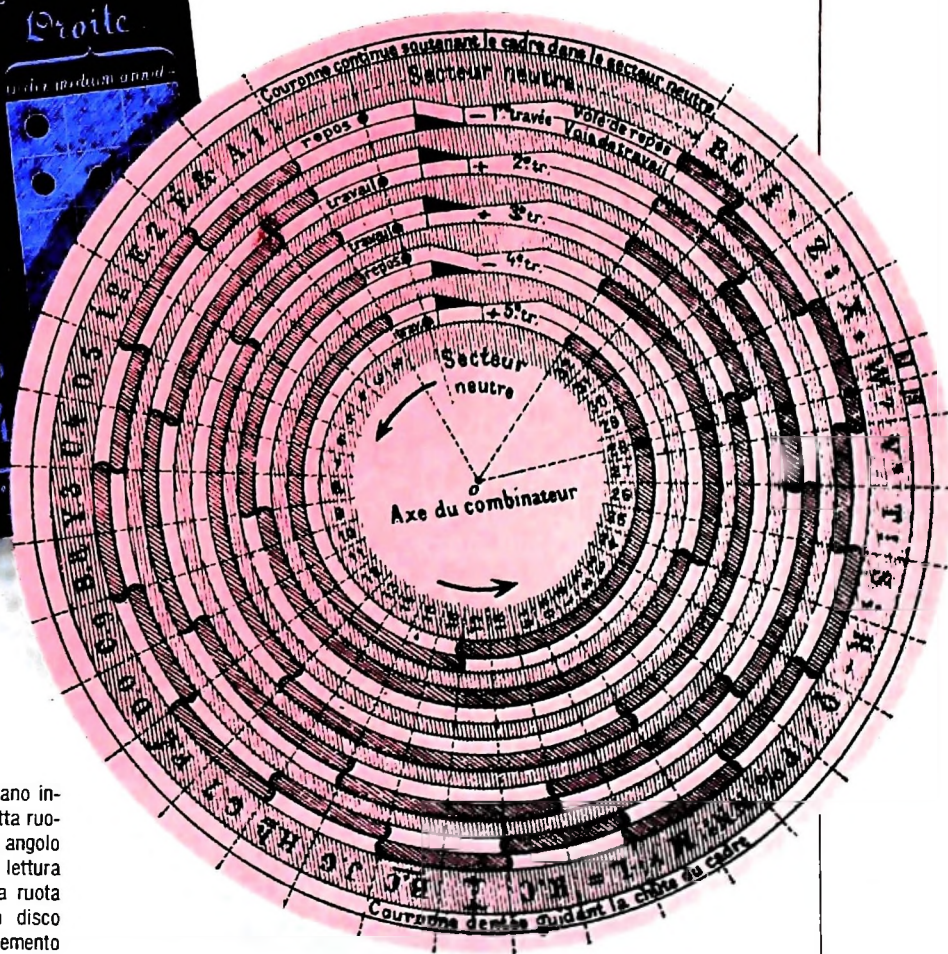


Tabella 1 - Sistemi di numerazione

Base, b	Nome
2	binario
3	ternario
4	quaternario
8	ottale
10	decimale
16	esadecimale

In questa pagina sono rappresentati i tre sistemi di numerazione maggiormente utilizzati nei calcolatori, quello binario, usato nel "linguaggio di macchina", quello esadecimale, usato nel linguaggio assembleatore (detto anche Assembler), e quello decimale, usato nei linguaggi di alto livello. Naturalmente, sia i valori nel sistema decimale sia quelli nel sistema esadecimale devono

essere "tradotti" nel sistema binario perché la macchina — che opera solamente su cifre binarie — possa elaborarli. Alcuni dei principali codici di traduzione sono rappresentati nella pagina a fronte e nella pagina successiva, dove compare il codice "ASCII" di rappresentanza dei caratteri alfabetici e della punteggiatura, anch'essi tradotti in cifre binarie.

I numeri negativi e positivi

Nel sistema binario esistono due modi per rappresentare un numero negativo. La prima è quella di mettere un'altra cifra binaria (o "bit", termine usato universalmente che viene da *binary digit*, cifra binaria) alla sinistra del numero espresso in cifre binarie. Se tale bit è zero, indica per convenzione che il numero è positivo, se è uno indica che il numero è negativo (Tabella 4). In questa forma esiste una differenza fra uno 0 con segno positivo e uno 0 con segno negativo.

L'altro modo per rappresentare un numero dotato di segno è quello di tenere fisso il numero di elementi e di trasformare invece un numero positivo in rappresentazione binaria nel suo negativo attraverso la formula del "complemento a due" (Tabella 5).

Questa forma è quella usata in molti circuiti ALU che, come si era già detto, sono solo capaci di sommare, per cui l'operazione di sottrazione fra due numeri è possibile solo sommando al primo il complemento a due del secondo. In questa rappresentazione sparisce ogni differenza fra uno zero dotato di segno + e uno dotato di segno -.

Sorge tuttavia l'inconveniente che la gamma di numeri rappresentabili con soli quattro bit si è dimezzata: mentre nel campo dei numeri positivi si potevano rappresentare sedici numeri dallo zero al quindici, ora si possono rappresentare otto numeri positivi dallo zero al sette e sette numeri negativi da — uno a — sette.

Tabella 2 - Corrispondenze numeriche fra i vari sistemi

b=10	b=2	b=3	b=4	b=8	b=16
0	0	0	0	0	0
1	1	1	1	1	1
2	10	2	2	2	2
3	11	10	3	3	3
4	100	11	10	4	4
5	101	12	11	5	5
6	110	20	12	6	6
7	111	21	13	7	7
8	1000	22	20	10	8
9	1001	100	21	11	9
10	1010	101	22	12	A
11	1011	102	23	13	B
12	1100	110	30	14	C
13	1101	111	31	15	D
14	1110	112	32	16	E
15	1111	120	33	17	F
16	10000	121	100	20	10
17	10001	122	101	21	11
18	10010	200	102	22	12
19	10011	201	103	23	13
20	10100	202	110	24	14
21	10101	210	111	25	15
22	10110	211	112	26	16

Tabella 3 - I tre sistemi più usati in elettronica digitale

Decimale	Binario	Esadecimale
	peso: 8421	
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Tabella 4 - Numeri binari positivi e negativi con bit di segno

Decimale con segno	Binario con segno	Decimale con segno	Binario con segno
+0	00000	-0	10000
+1	00001	-1	10001
+2	00010	-2	10010
+3	00011	-3	10011
+4	00100	-4	10100
+5	00101	-5	10101
+6	00110	-6	10110
+7	00111	-7	10111
+8	01000	-8	11000
+9	01001	-9	11001
+10	01010	-10	11010
+11	01011	-11	11011
+12	01100	-12	11100
+13	01101	-13	11101
+14	01110	-14	11110
+15	01111	-15	11111

Tabella 5 - Numeri binari positivi e negativi con complemento a 2

Decimale positivo	Binario positivo	Decimale negativo	Binario negativo
0	0000	-0	0000
1	0001	-1	1111
2	0010	-2	1110
3	0011	-3	1101
4	0100	-4	1100
5	0101	-5	1011
6	0110	-6	1010
7	0111	-7	1001

I codici dei calcolatori

Prima di cominciare a trattare i codici esistenti, è opportuno chiarire un po' meglio il significato del termine "codice". Nella vita di ogni giorno, un'infinità di messaggi ci viene trasmessa attraverso dei codici: basta pensare al comune semaforo che quando è rosso ci comunica di non attraversare, quando è giallo ci avverte che è in procinto di cambiare stato per cui bisogna prepararsi, quando è verde ci permette di passare liberamente.

Così come il semaforo trasmette informazioni con un codice di colori, vi sono altri codici grafici molto usati, come frecce, cartelli segnaletici ecc.

Anche nei calcolatori esistono codici, che servono a rappresentare numeri, lettere e altri segnali di comando. Si è già visto come si potevano rappresentare i numeri esadecimali da 0 a 15 attraverso un sistema di quattro bit con pesi 8, 4, 2, 1, ma questa non è la sola forma; c'è un altro codice importante il BCD (*Binary Coded Decimal*, cioè decimale codificato binario), (Tabella 6) chiamato appunto decimale per la sua corrispondenza biunivoca con il sistema decimale, anche se lascia senza senso (e quindi incomprensibili) tutte le altre combinazioni possibili dell'insieme di quattro bit.

Una variante del codice BCD, usato in molti calcolatori, è il codice 4221 (Tabella 7): esso è interessante per la sua velocità nel trasformare in complemento a 9 il numero decimale rappresentato semplicemente trasformando da 0 a 1 e da 1 a 0 tutti i bit binari. Altri codici molto usati sono quello GRAY (Tabella 8) che ha la caratteristica di mutare un solo bit nel passare da un numero a quello contiguo, l'EXCESS-3 (Tabella 9) che corrisponde al BCD più 3 (binario 0011). Questo codice, come pure il 4221, si autocomplementa cambiando tutti gli 1 in 0 e tutti gli 0 in 1.

Il TWO-OUT-OF-FIVE è un codice (Tabella 10) che permette di scoprire errori, usato specialmente in trasmissioni di dati fra registri. Il BIQUINARY CODE (Tabella 11) è anch'esso usato per scoprire errori di trasmissione. Infine, vi sono le configurazioni ASCII (pronuncia "aschi") da American Standard Code for Information Interchange e l'EBCDIC

Tabella 6 - Codice BCD

0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

Tabella 7 - Codice 4221

0	0000	5	0111
1	0001	6	1100
2	0010	7	1101
3	0011	8	1110
4	1000	9	1111

Tabella 8 - Codice GRAY

0:	0000	8:	1100
1:	0001	9:	1101
2:	0011	10:	1111
3:	0010	11:	1110
4:	0110	12:	1010
5:	0111	13:	1011
6:	0101	14:	1001
7:	0100	15:	1000

Tabella 9 - Codice EXCESS-3

0:	0011	5:	1000
1:	0100	6:	1001
2:	0101	7:	1010
3:	0110	8:	1011
4:	0111	9:	1100

Tabella 10 - Codice TWO-OUT-OF-FIVE

0:	11000	5:	01010
1:	00011	6:	01100
2:	00101	7:	10001
3:	00110	8:	10010
4:	01001	9:	10100

Tabella 11 - Codice BIQUINARY

0: 01 00001	5: 10 00001
1: 01 00010	6: 10 00010
2: 01 00100	7: 10 00100
3: 01 01000	8: 10 01000
4: 01 10000	9: 10 10000

(Extended BCD Interchange Code), (Tabella 12). Questi due codici sono quelli usati normalmente per la rappresentazione di dati alfanumerici (cioè formati da lettere dell'alfabeto e numeri) e comandi.

Ma perché il codice usato all'interno delle macchine (il "linguaggio di macchina") è sempre quello binario? Perché corrisponde a due stati elettrici di tensione più bassa (lo "zero") e più alta ("uno") presenti in tutti i circuiti logici di un calcolatore.

Un calcolatore, per così dire, conta con solo due dita mentre noi ne utilizziamo dieci. Da qui la necessità di tradurre il nostro codice decimale in uno che sia più facilmente accessibile al calcolatore.

Tabella 12 - Codici Alfanumerici - N.B.: P è il bit di parità

Carattere	Codice EBCDIC	Codice ASCII	Carattere	Codice EBCDIC	Codice ASCII
blank	0100 0000	P010 0000	A	1100 0001	P100 0001
	0100 1011	P010 1110	B	1100 0010	P100 0010
(0100 1101	P010 1000	C	1100 0011	P100 0011
+	0100 1110	P010 1011	D	1100 0100	P100 0100
\$	0101 1011	P010 0100	E	1100 0101	P100 0101
*	0101 1100	P010 1010	F	1100 0110	P100 0110
)	0101 1101	P010 1001	G	1100 0111	P100 0111
—	0110 0000	P010 1101	H	1100 1000	P100 1000
/	0110 0001	P010 1111	I	1100 1001	P100 1001
.	0110 1011	P010 1100	J	1101 0001	P100 1010
'	0111 1101	P010 0111	K	1101 0010	P100 1011
=	0111 1110	P011 1101	L	1101 0011	P100 1100
			M	1101 0100	P100 1101
0	1111 0000	P011 0000	N	1101 0101	P100 1110
1	1111 0001	P011 0001	O	1101 0110	P100 1111
2	1111 0010	P011 0010	P	1101 0111	P101 0000
3	1111 0011	P011 0011	Q	1101 1000	P101 0001
4	1111 0100	P011 0100	R	1101 1001	P101 0010
5	1111 0101	P011 0101	S	1110 0010	P101 0011
6	1111 0110	P011 0110	T	1110 0011	P101 0100
7	1111 0111	P011 0111	U	1110 0100	P101 0101
8	1111 1000	P011 1000	V	1110 0101	P101 0110
9	1111 1001	P011 1001	W	1110 0110	P101 0111
			X	1110 0111	P101 1000
			Y	1110 1000	P101 1001
			Z	1110 1001	P101 1010

Glossario

ALU (Arithmetic Logic Unit) - sigla che indica l'unità logica e aritmetica, una delle parti costitutive della CPU. È l'unità in cui vengono effettivamente svolte le operazioni di carattere logico e aritmetico, cioè le operazioni booleane e quelle di somma, sottrazione, moltiplicazione, divisione ecc.

Baud - unità di velocità di trasferimento di dati. Normalmente si definisce come una velocità di trasferimento di un bit al secondo. "Trasmettere a 4800 baud", per esempio, significa trasmettere informazioni alla velocità di 4800 bit al secondo.

Clock - letteralmente, "orologio". È reso meglio, in italiano, da "sincronizzatore" o "temporizzatore". Un circuito elettronico che fornisce i segnali di sincronizzazione per tutte le attività interne a un computer. Un ciclo di clock è il tempo minimo impiegato dall'hardware della macchina per l'esecuzione di una operazione elementare. La velocità del sincronizzatore è data normalmente in Hertz, per lo più in megahertz (MHz, milioni di Hertz). 1 Hertz è pari a un ciclo al secondo. Una macchina che disponga di un clock a 4 MHz è una macchina in grado di eseguire 4 milioni di operazioni elementari al secondo.

Comando - una istruzione di un linguaggio di programmazione usata in modo immediato, cioè per "dire" al computer di eseguire immediatamente una operazione. La maggior parte delle istruzioni del BASIC possono essere usate come comandi, in modo immediato; alcune hanno senso invece solo in modo differito, usate cioè all'interno di un programma. Quando un'istruzione è usata in modo differito non è più indicata come "comando" ma come "enunciato" o, all'inglese, "statement".

Drive - unità di lettura e registrazione di informazioni su disco magnetico.

Floppy disk - dischetto flessibile. Un supporto di memoria di massa costituito da un dischetto di mylar rivestito di un materiale magnetico, su cui possono essere immagazzinate informazioni (con un principio analogo a quello su cui si basa la registrazione fonografica). Esistono floppy disk di formati diversi: i più comuni sui microcomputer sono quelli del diametro di 5 pollici e 1/4; esistono dischetti anche del diametro di 8 pollici, usati più frequentemente su macchine di maggiori dimensioni.

Hard copy - letteralmente, "copia dura", cioè copia permanente dell'output di un calcolatore. Indica, normalmente, uno stampato, in contrapposizione alla "copia non permanente" o "soft copy" fornita da un monitor o dal video del televisore.

IC - sigla di Integrated Circuit, circuito integrato.

Input - ingresso, inserimento di informazioni.

Input, dispositivo di - un dispositivo che consente l'ingresso o l'inserimento di informazioni in un computer o in qualunque apparecchiatura di un sistema di calcolo. Il più tipico dispositivo di input è la tastiera.

Mainframe - un calcolatore di grandi dimensioni, di costo elevato e di elevate prestazioni. Il termine è usato per distinguere la fascia più alta degli elaboratori commerciali, in contrapposizione a "minicomputer" e "microcomputer". Il vocabolo significa letteralmente "armadio principale" e veniva usato originariamente per indicare i grandi telai e i grandi armadi che contenevano le migliaia di valvole termoioniche dei primi calcolatori elettronici.

Periferica - qualunque dispositivo di input/output. Una stampante è una periferica tipica.

L'INTELLIGENZA ARTIFICIALE

Come simulare il comportamento umano per mezzo di macchine programmate.

Chiunque utilizzi un calcolatore elettronico arriva prima o poi a chiedersi fino a che punto il comportamento della macchina che ha di fronte è confrontabile con il proprio, e più in generale in che misura un calcolatore possa essere considerato intelligente. Per esprimere un giudizio a riguardo, sarebbe necessario avere ben chiaro il significato del termine "intelligente", concetto di affascinante profondità. Su queste considerazioni si soffermarono naturalmente anche gli utilizzatori e i costruttori dei primi elaboratori sul finire degli anni Quaranta. Spesso, sull'onda dell'ottimismo suscitata dai brillanti successi dei primi calcolatori e dal sorprendente comportamento delle macchine programmate, nello svolgere compiti normalmente riservati all'uomo, cominciò a circolare, nella giovane comunità degli informatici, l'intenzione di ottenere dalle macchine un comportamento "intelligente".

Perseguendo questa linea di ricerca si pensava di ottenere due generi di vantaggi: da un lato, macchine in grado di simulare il comportamento umano in situazioni in cui è necessaria una certa capacità di giudizio avrebbero potuto assolvere compiti noiosi o pericolosi per l'uomo; dall'altro, si pensava che gli elaboratori avrebbero costituito un sostanziale supporto sperimentale per gli studi sul comportamento intelligente e sul funzionamento della mente umana. Questa idea, ancora oggi relativamente diffusa, è ispirata dal metodo che, in generale, si segue affrontando lo sviluppo di un programma. Questo deve condurre un calcolatore a comportarsi in modo adeguato rispetto a specifiche situazioni.

La modellistica ingenua

Ogni programma può essere visto in questo modo: ad esempio un programma di fatturazione deve far comportare il calcolatore come una macchina stampante che reagisce all'introduzione di alcuni dati con la produzione di una o più fatture. Il metodo, noto implicitamente o esplicitamente ad ogni programmatore, consiste nell'analisi della situazione data, nello sviluppo di una ipotesi di meccanismo che esibisca il comportamento desiderato (il modello), nella traduzione di tale ipotesi in un programma e nel confronto del comportamento di una macchina così programmata con quello desiderato. Questa sequenza di azioni deve chiaramente essere ripercorsa, modificando il programma o addirittura il modello, ogni volta che ci si accorge che non si è ottenuto il risultato desiderato.

Uno studio della mente che si basi sperimentalmente su un tentativo di simulazione del comportamento umano intelli-

gente realizzata con una macchina programmata è quindi essere favorito:

1) La continua costruzione di modelli di comportamento favorisce lo sviluppo di linguaggi e di metodi per descrivere meccanismi di ragionamento e di reazione a stimoli.

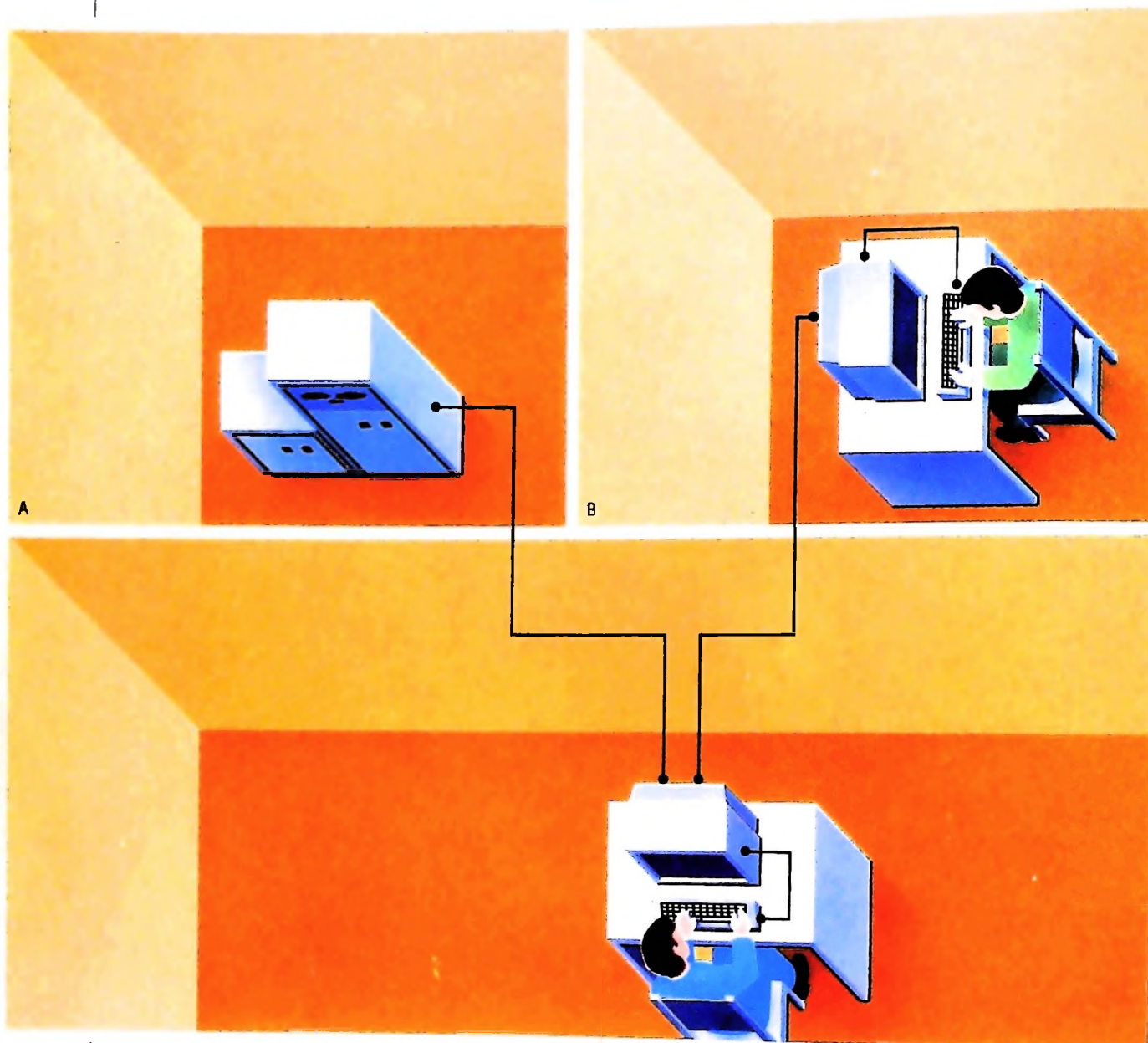
2) I programmi, per funzionare, devono essere definiti in modo completo, cioè prevedendo tutte le possibilità nel contesto assegnato, e preciso, cioè dettagliando completamente ognuna delle azioni richieste. In questo modo la teoria, per avere la possibilità di essere sperimentata, è spinta ad essere il più dettagliata e completa possibile.

3) Le macchine programmate sono un soggetto di analisi sperimentale molto più disponibile degli stessi esseri viventi. È infatti possibile iterare esperimenti un grande numero di volte evitando compromissioni psicologiche. È facile poi rimuovere o aggiungere parti del sistema o delle informazioni a disposizione per verificarne l'influenza sul comportamento del sistema.

Il test di Turing

L'interesse che suscitano queste idee è testimoniato anche dal fatto che già il celebre matematico e precursore dell'informatica Turing si era posto il problema del riconoscimento di un comportamento intelligente eventualmente esibito da una macchina. Turing arrivò alla definizione di un test che un ente "intelligente" deve essere in grado di superare. Un soggetto umano giudicante, uno ausiliario e la macchina sottoposta a test sono posti in stanze separate e possono comunicare solo con dispositivi che di solito servono alla comunicazione uomo-macchina, ad esempio un videoterminale. Il giudicante rivolge arbitrariamente domande ai due enti. Il test viene superato se il giudicante non è in grado di riconoscere in quale stanza si trova la macchina e in quale la persona. Il test è oggi molto criticato per l'enfasi che pone sulla sensibilità del giudicante e soprattutto per il fatto che è stato recentemente superato da macchine e programmi non di enorme sofisticazione.

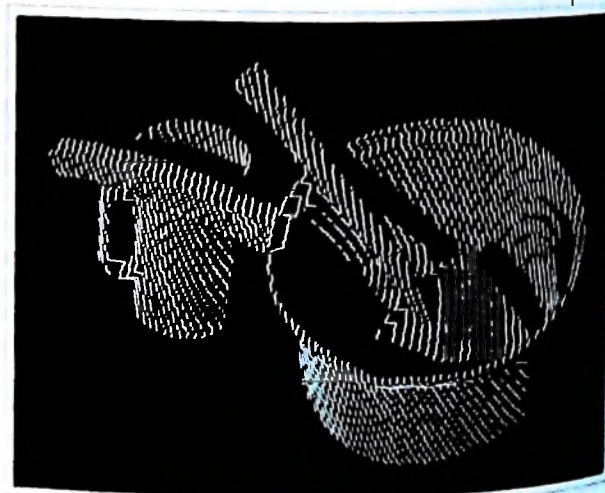




Il test di Turing. L'esaminatore è collegato — via telescrivente — con un essere umano e un elaboratore, cioè un'intelligenza artificiale. Deve stabilire chi dei due (B o A) è l'intelligenza artificiale, in base alle risposte che ottiene alle sue domande. Il compito può essere reso difficile se il calcolatore è stato programmato in modo da reagire con qualche elemento di casualità a domande identiche.

Il problema si complica

Parte della comunità degli informatici iniziò quindi, fin dagli anni Cinquanta, a impegnarsi nel tentativo di progettare e realizzare simulazioni di aspetti del comportamento umano per mezzo di macchine programmate. Questo campo di ricerca assunse ben presto il nome di Intelligenza Artificiale (A.I.), nome che conserva ancora anche se i contorni del campo si sono fatti via via più sfumati e l'atteggiamento dei ricercatori è ora piuttosto diverso. Grosso modo l'obiettivo iniziale era, per molti, quello di produrre un sistema con caratteristiche di comportamento umano, che comprendesse il linguaggio naturale e fosse in grado di apprendere. Il sistema sarebbe poi stato "addestrato" in modo diverso per ogni ap-



plicazione specifica e quindi sarebbe stato in grado di sostituire, nell'ambito di quella particolare attività, gli umani. Era già chiaro infatti che non sarebbe stato possibile realizzare concretamente una simulazione completa del comportamento di un essere umano. Mancando una trattazione formale esauriente dei meccanismi di ragionamento intelligente, il metodo non poteva essere che quello delle approssimazioni successive: ogni modello-tentativo viene realizzato e confrontato con il comportamento umano in opportune situazioni reali; l'analisi delle difficoltà esibite dal programma dà luogo alla formulazione di un nuovo modello e a un nuovo ciclo di raffinamento. D'altra parte, le esorbitanti dimensioni del problema e la quantità di sottoproblemi di vasta portata conducono naturalmente a formulare dei modelli parziali, all'occuparsi cioè di un aspetto del comportamento umano alla volta. Vennero quindi individuati i sottoproblemi importanti verso cui rivolgere l'attenzione.

a) **Ragionamento:** ogni essere umano esibisce una certa capacità di ragionare. La definizione di questo concetto è ciò che rende più difficoltoso il cammino di chi studia la mente. I meccanismi che permettono all'uomo di estrarre nuove informazioni a partire da informazioni acquisite attraverso l'esperienza sono oggetto di studio da sempre. Gli strumenti e le teorie sviluppate in molti anni dalla logica matematica sono utilizzabili per cercare di simulare processi di ragionamento deduttivo (del tipo di quelli che intervengono nella dimostrazione di teoremi in matematica).

Altri strumenti sono stati invece utilizzati per realizzare programmi in grado di giocare (dama, scacchi ecc.). Quello del ragionamento si presentò comunque come il problema centrale e il più spinoso.

b) **Comprensione del linguaggio naturale:** è fondamentale che la comunicazione con un sistema che si candida per superare il test di Turing non può che avvenire con il linguaggio con cui comunicano gli uomini tra loro. È quindi necessario che il sistema prodotto sia in grado di estrarre la massima quantità di informazioni dalle frasi che gli verranno rivolte. È questo il problema che fu affrontato più baldanzosamente e con maggiore spiegamento di forze.

c) **Movimento:** nel caso di simulazione di attività umane che prevedono un intervento diretto e fisico sul mondo reale è

necessario costruire sistemi in grado di muoversi e di agire, ma soprattutto bisogna individuare e riprodurre i processi mentali che sovrintendono al movimento.

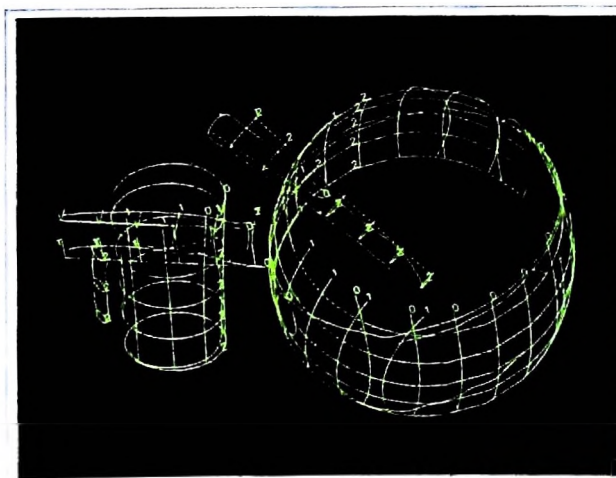
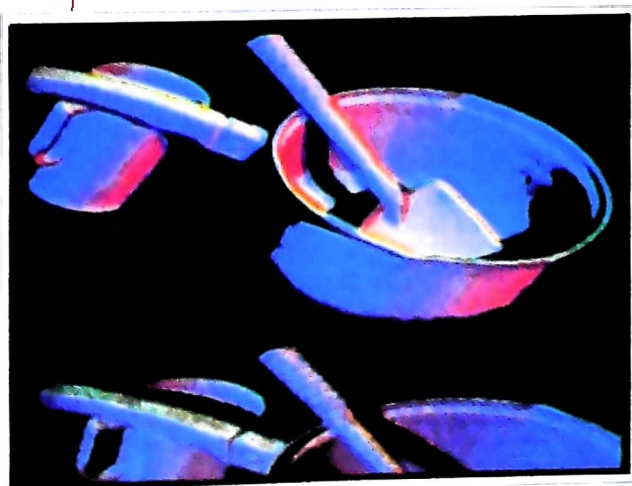
d) **Visione:** oltre alla comunicazione linguistica, il principale strumento di acquisizione di informazioni per un sistema intelligente è quello della visione. Naturalmente, come nel caso del linguaggio, una volta ottenute le immagini è necessario essere in grado di estrarre da queste tutte le informazioni utili. Questo genere di studi si doveva allora però misurare con problemi tecnologici non banali che oggi sono in larga parte risolti.

e) **Apprendimento:** l'intenzione era quella di sviluppare sistemi con capacità generiche in grado di acquisire competenze specifiche. I meccanismi dell'apprendimento, di grande interesse per molte altre discipline umanistiche e scientifiche, sono quindi alla base dei sistemi intelligenti come pensati dai primi studiosi di A.I.

f) **Archiviazione "intelligente":** ogni sistema che simuli un comportamento umano deve essere in grado di conservare in modo organizzato le informazioni in proprio possesso e di riutilizzarle nel numero di casi più alto possibile. Il problema naturalmente è quello di capire quando un'informazione può essere utile o meno.

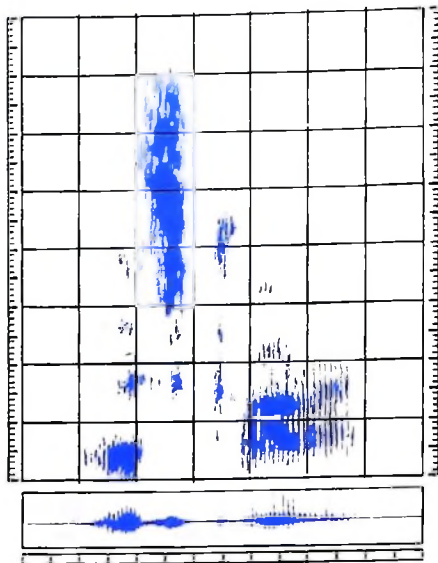
L'evoluzione delle ricerche

Nel corso degli anni, gli studiosi dediti all'A.I. si accorsero che, pur ottenendo brillanti risultati parziali, l'evoluzione di ogni sottocampo si andava separando sempre più da quella degli altri. Chi si applicava alla comprensione del linguaggio naturale, ad esempio, aveva sviluppato tecniche, metodologie e linguaggi specifici ed aveva difficoltà persino a comunicare con chi si occupava dei fenomeni di visione o di movimento. Ci si allontanava progressivamente dall'idea di sottodiscipline strettamente interagenti rivolte ad un unico scopo comune. In fondo non si è mai rinunciato all'idea di realizzare un sistema intelligente. Sulla nota rivista inglese "Information Systems" è recentemente (1978-1981) apparsa una serie di articoli nell'ambito di un dibattito che verteva appunto sulla possibilità o meno di poter formalizzare il concetto di intelli-

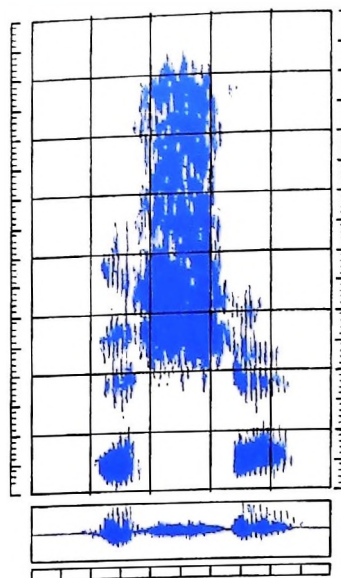


Il riconoscimento delle forme (pattern recognition). Analizzando una scena, il calcolatore la elabora e quindi assegna ognuno degli oggetti presenti in essa ad una determinata categoria, in questo caso una pentola, un boccale e un cucchiaino (da IEEE Spectrum).

Come riconoscere la voce



Il riconoscimento della voce. Esso inizia con l'analisi di fonogrammi di singole parole o frasi, come quelle sopra, che vengono successivamente elaborati con tecniche molto raffinate basa-



te spesso sulla metodologia del Linear Predictive Coding. Il processo termina con l'assegnazione dei fonemi a determinate categorie, come nel caso degli oggetti (da IEEE Spectrum).

genza o di realizzare sistemi automatici intelligenti. Il dibattito, pur senza raggiungere risultati definitivi, ha presentato aspetti di grande interesse e ha testimoniato come un tale argomento sia ancora stimolante.

Ognuno dei gruppi di ricerca che intraprese lo studio di uno dei campi di cui si è parlato venne ostacolato e rallentato, oltre che da problemi specifici tutt'altro che banali del proprio campo d'interesse, da un comune e insidioso nemico: la complessità. Qualunque fosse l'argomento trattato, ci si trovava di fronte a problemi di complessità, per ogni passo dei tentativi di simulazione.

1) Complessità di analisi: è ovvio che l'analisi del comportamento umano, in ogni suo aspetto non banale, non è affrontabile ingenuamente. In generale si tratta di misurarsi con difficoltà epistemologiche e metodologiche oltre che tecniche. Fu subito chiaro che nell'indagine è necessario ricorrere a metodi e risultati di altre discipline scientifiche e umanistiche. Bisogna quindi imparare a comunicare con studiosi di impostazione diversa e spesso diffidenti, e a familiarizzare velocemente con linguaggi e punti di vista diversi.

2) Complessità dei modelli: naturalmente, ogni ipotesi di meccanismo intelligente conterrà una enorme quantità di informazioni e sarà composta di un grande numero di sottoparti collegate tra loro in modo molto complicato. Si presenta il problema di controllare e soprattutto di rappresentare queste vaste costruzioni spesso solo in parte formalizzate.

3) Complessità dei programmi: i modelli, per avere riscontro sperimentale, devono essere tradotti in programmi per le macchine prodotte dalla tecnologia corrente. Si tratta quindi di programmi enormi, in cui ogni parte ha grande contenuto concettuale e che devono essere spesso modificati anche sostanzialmente.

L'atteggiamento verso il software è forzatamente quello del-

l'usa e getta. Le macchine, soprattutto nei primi anni dell'A.I., si presentavano all'utente in modo molto poco amichevole. Si comunicava per mezzo di nastri e schede perforate o di lente e rumorose telescriventi. Gli strumenti per sviluppare programmi erano ancora piuttosto rudimentali. Le metodologie di sviluppo software stavano nascendo ma erano orientate alle esigenze dei grossi sistemi EDP: tendevano quindi a facilitare la costruzione di programmi che rispondessero a specifiche molto precise e con una architettura molto solida ma cristallizzata e non facilmente modificabile; ad esempio, l'alto costo di un voluminoso programma di gestione magazzino è dovuto soprattutto alla sua difficile manutenzione, legata ai continui aggiornamenti legislativi.

I problemi di complessità fecero, col trascorrere degli anni e coll'approfondirsi degli studi, invecchiare e poi quasi scomparire l'idea iniziale di produrre sistemi intelligenti in generale, in grado di essere addestrati e di risolvere ogni tipo di problema. La comunità A.I. eliminò progressivamente dai propri scritti le introduzioni ottimistiche sui sistemi intelligenti in generale. Le difficoltà furono comunque origine e alimento per una serie di ricerche e di realizzazioni altamente produttive. I risultati intermedi conseguiti lottando con i problemi di complessità sono stati determinanti per l'informatica in senso lato.

Uno sguardo avanti

I prossimi articoli di questa serie cercheranno di illustrare questi risultati e la loro influenza sull'attuale sviluppo informatico. Verrà anche fornita una visione sintetica dello stato dei vari campi di ricerca che vengono indicati collettivamente con il nome di Intelligenza Artificiale.

Lezione 8

Linguaggi per descrivere linguaggi

Siamo da sempre abituati a descrivere le caratteristiche della nostra lingua, l'italiano, attraverso quella che chiamano "grammatica", cioè un insieme di regole, pure espresse in italiano. L'uso di un linguaggio per descrivere il linguaggio stesso non è esente da rischi di incomprensioni e ambiguità; per esempio, ci si può trovare a fare affermazioni di questo tipo:

VERBO è un sostantivo

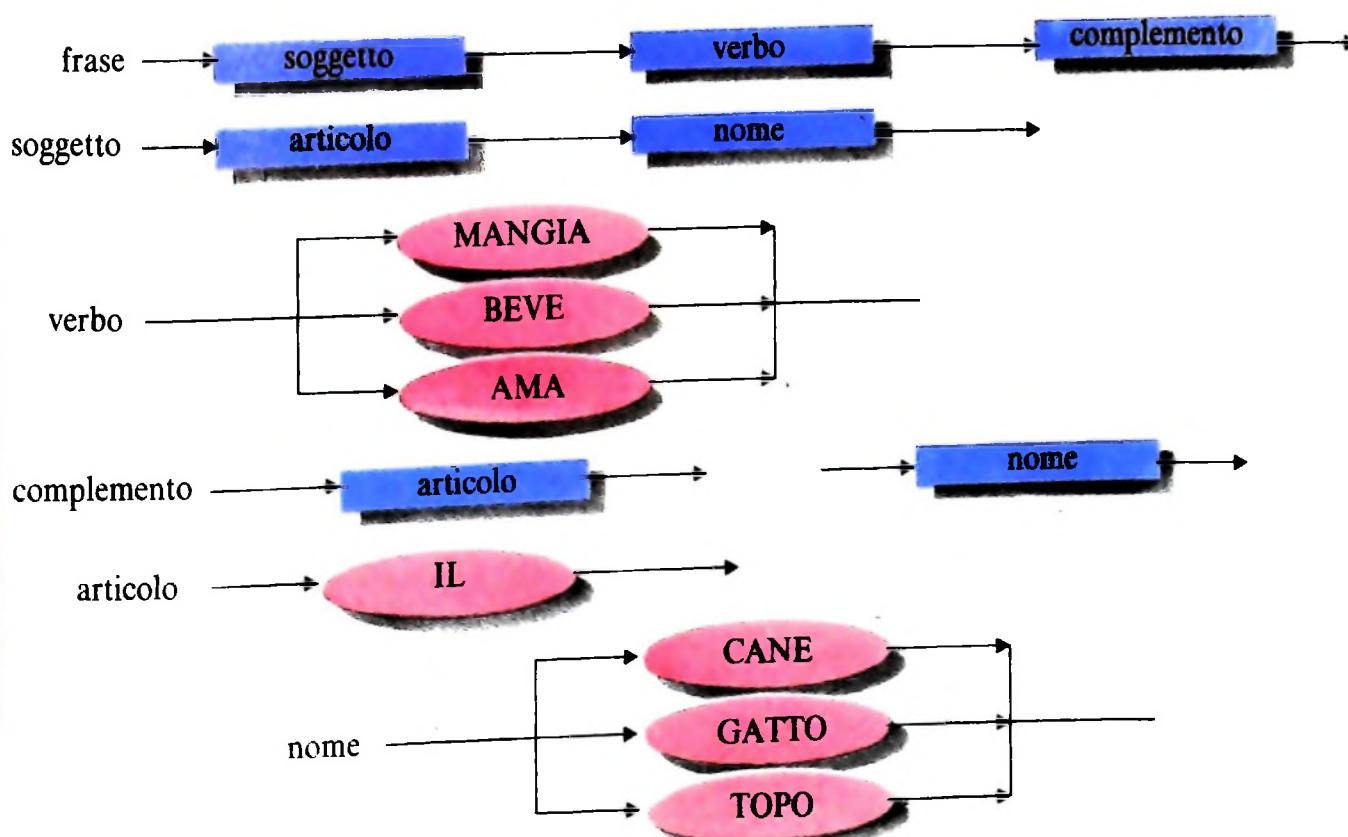
che sembra una contraddizione, ma che è corretta se opportunamente interpretata: l'elemento grammaticale che descrive un'azione è detto **VERBO**

la parola **VERBO** è un sostantivo che identifica l'elemento grammaticale che descrive un'azione, come possiamo facilmente ricavare dalla lettura di un dizionario.

Allo scopo di evitare ambiguità e difficoltà di comunicazione, è opportuno che, per descrivere la grammatica di un linguaggio, se ne usi un altro, eventualmente appositamente definito, che non possa essere confuso con il precedente. Parliamo allora di **METALINGUAGGIO**. La grammatica di un linguaggio, descritta in un metalinguaggio, vedrà allora almeno tre tipi di elementi:

- i cosiddetti **SIMBOLI TERMINALI** (cioè le parole del linguaggio che stiamo descrivendo: **IL, CANE, MANGIA...**)
- i cosiddetti **SIMBOLI NON TERMINALI** (cioè simboli che denotano i vari elementi grammaticali, come ad esempio il soggetto, il verbo, il complemento oggetto...)
- le cosiddette **REGOLE DI PRODUZIONE** (cioè regole che permettono di costruire frasi corrette).

Tra i possibili metalinguaggi, adotteremo le **CARTE SINTATTICHE**, di cui forniamo un semplice esempio:



Proviamo a usare la grammatica: partendo dalla definizione di "frase", seguiamo le frecce e:

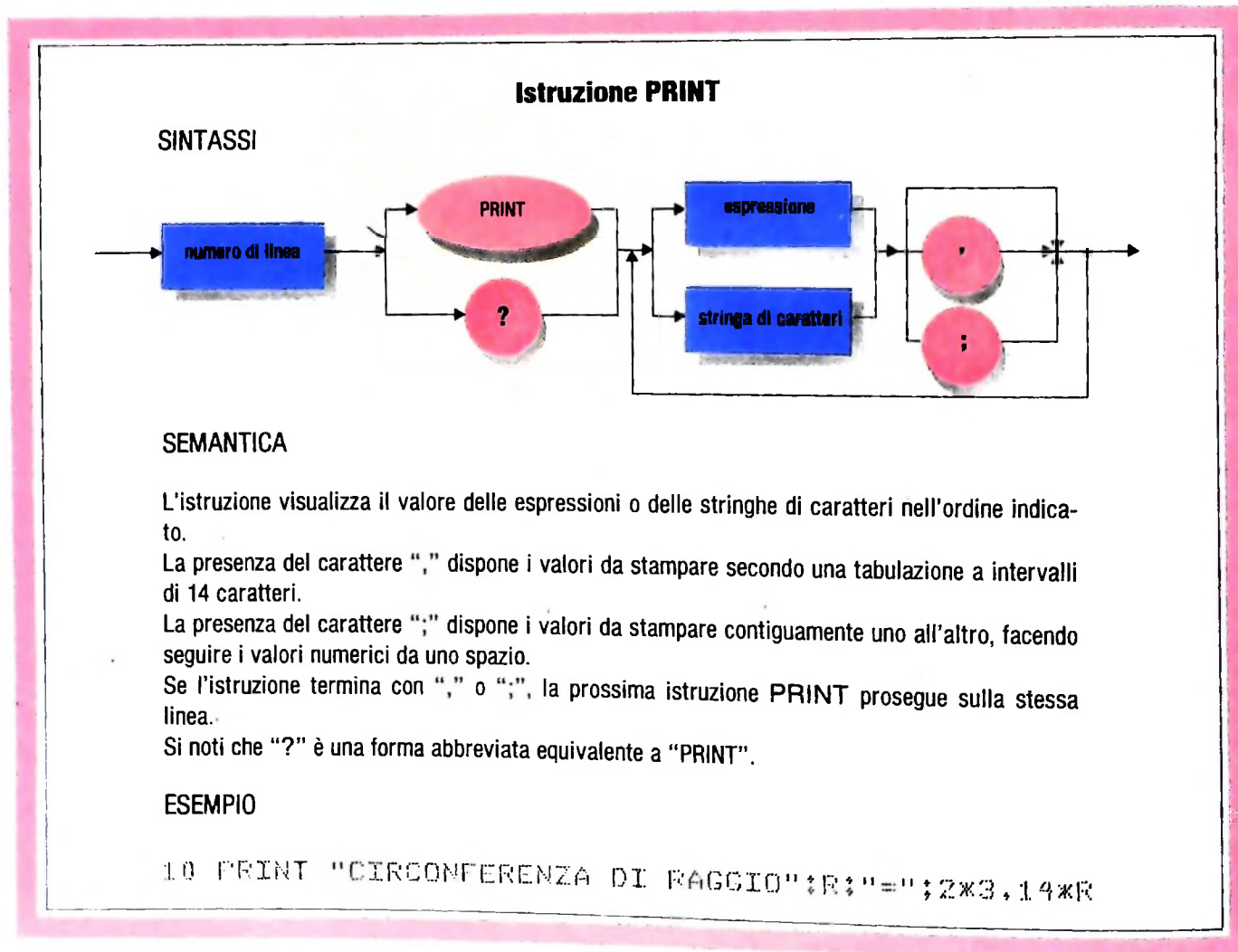
- percorriamo la carta sintattica seguendo le frecce, e, davanti a un bivio, scegliamo a nostro piacimento la strada
- ogni volta che troviamo un simbolo terminale (quelli in caratteri maiuscoli in caselle con angoli arrotondati) lo trascriviamo pari pari
- ogni volta che incontriamo un simbolo non terminale (quelli minuscoli in caselle rettangolari) ripetiamo il procedimento facendo riferimento alla carta sintattica che lo descrive

Così, applicando successivamente le regole, passiamo dalla "frase" a una particolare frase:

frase
 soggetto verbo complemento
 articolo nome MANGIA articolo nome
 IL GATTO MANGIA IL CANE

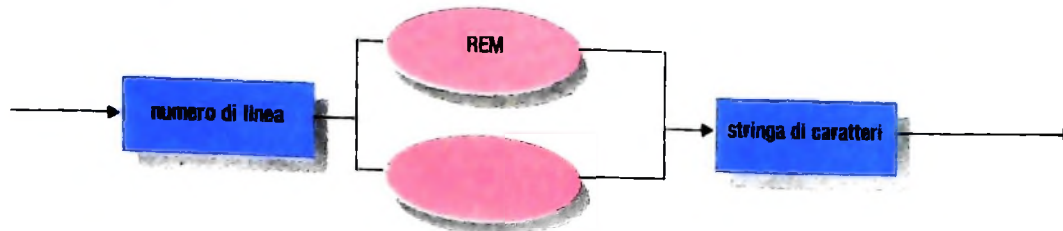
La grammatica fornita descrive un frammento piccolo e rozzo della lingua italiana, e sarebbe veramente arduo descrivere tutta la lingua con tali meccanismi, ma non è così per i linguaggi di programmazione, estremamente più semplici: tutti i linguaggi di programmazione vengono descritti con una grammatica di questo tipo, e noi useremo le carte sintattiche per descrivere il BASIC.

Iniziamo a esaminare la sintassi dell'istruzione PRINT, già largamente usata.



Istruzione REM

SINTASSI



SEMANTICA

L'istruzione permette di inserire commenti nel codice, cioè "frasi" che vengono ignorate in fase di esecuzione, ma che compaiono nella lista del programma.

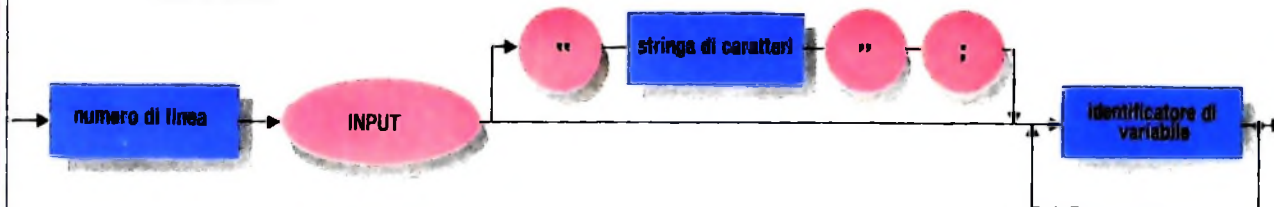
L'istruzione prevede due formati, rispettivamente REM e ' ; pertanto un commento può essere scritto in entrambe le seguenti forme:

```

10 REM Questo e' un commento
20 'Questo e' un commento
  
```

Istruzione INPUT

SINTASSI



SEMANTICA

L'istruzione INPUT permette di fornire dati al programma da tastiera.

La richiesta di dati può essere preceduta dalla visualizzazione di un messaggio chiarificatore.

L'effetto dell'istruzione è la visualizzazione di un "?" dopo cui l'utente dovrà digitare i valori richiesti. È possibile in particolare richiedere più dati con la stessa istruzione. In tal caso il programma resta in attesa fino a quando tutti i valori attesi sono stati forniti.

Qualora il tasto "ENTER" venga digitato prima che tutti i dati attesi siano stati forniti, il programma visualizzerà "??" a nuova linea per indicare che è ancora in attesa di dati.

Qualora al contrario vengano fornite più informazioni di quelle richieste, quelle aggiuntive vengono ignorate.

I dati richiesti possono essere sia valori numerici sia informazioni in caratteri. Nel caso in cui caratteri vengano forniti ove numeri sono attesi, il messaggio "Redo from start" segnala l'errore e ripropone la richiesta.

Un esempio di istruzione INPUT è il seguente:

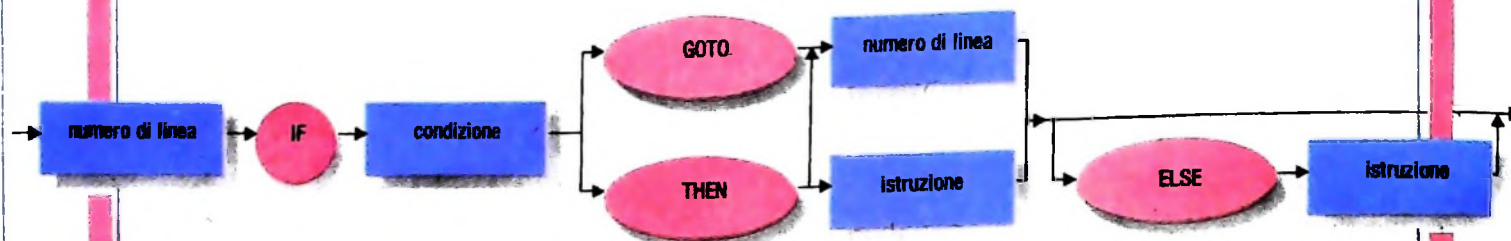
```

10 INPUT "Inserire la data (gg,mm,aa)";G,M,A
  
```

che produce la visualizzazione del messaggio di richiesta e attende quindi l'inserimento di tre valori numerici separati da ",",.

Istruzione IF

SINTASSI



SEMANTICA

L'istruzione IF consente di costruire selezioni, scegliendo tra due differenti sequenze d'esecuzione sulla base del valore di una condizione.

Se la condizione indicata risulta vera, viene eseguita l'istruzione GOTO seguente o l'istruzione introdotta da THEN, ovvero si salta al numero di linee che segue THEN, se presente. Il ramo ELSE non è obbligatorio. Qualora non sia espresso, se la condizione indicata è falsa l'esecuzione prosegue con l'istruzione seguente la IF. Se invece il ramo ELSE è indicato, allora se la condizione è falsa viene eseguita l'istruzione introdotta dalla ELSE stessa.

La condizione espressa può assumere differenti formati, come per esempio:

```
A=B  
A>B  
A<=B+1/C  
3*D=67/(B+Y)  
4/T=F AND 1=Z  
1=1 OR 2=2
```

È possibile innestare istruzioni IF purché non siano spezzate su più linee: IF è infatti una sola istruzione e come tale non può essere spezzata su più linee. È quindi possibile scrivere:

```
10 IF A=B THEN IF C>Z THEN PRINT " OK"
```

Cosa abbiamo imparato

In questa lezione abbiamo introdotto:

- il concetto di METALINGUAGGIO, cioè un linguaggio destinato appositamente alla descrizione di linguaggi, in particolare di programmazione
- il concetto di GRAMMATICA per un linguaggio di programmazione, insieme di regole comprendente
 - i SIMBOLI TERMINALI (cioè le parole del linguaggio da descrivere)
 - i SIMBOLI NON TERMINALI (cioè i simboli che descrivono elementi logici del linguaggio)
 - le REGOLE DI PRODUZIONE (cioè le regole da seguire per costruire le frasi del linguaggio)
- le CARTE SINTATTICHE, cioè un particolare tipo, semplice e intuitivo, ma preciso e potente, di metalinguaggio
- la sintassi completa delle istruzioni PRINT, REM, INPUT, IF

STRUMENTI PER IMPARARE

Usato dallo studente, l'elaboratore è uno strumento prezioso per acquisire speciali abilità e per trasmettere contenuti.

Riprendiamo il discorso dell'informatica applicata alla didattica esaminando il CAI (*Computer Aided Instruction*) al discente, ossia la situazione in cui l'elaboratore è usato dallo studente per migliorare o addirittura per reinventare l'apprendimento.

In questo campo applicativo si dischiude una vasta gamma di possibilità, correlate a impostazioni didattiche diverse, a diverse metodologie, e infine a differenti modalità di utilizzazione dell'elaboratore e, più in generale, degli strumenti dell'informatica.

Del resto faremo un discorso a parte, nei prossimi capitoli,

per quanto riguarda le metodologie impiegate in questo particolare settore.

Per ora cerchiamo di esaminare la situazione complessiva, attraverso un certo numero di esempi concreti.

L'allievo si trova di fronte a una macchina con la quale stabilisce un dialogo finalizzato all'apprendimento. Chi ha preparato il programma ha tenuto presenti particolari obiettivi e ha impiegato determinati metodi. In generale possiamo dire che l'impiego dell'elaboratore persegue l'obiettivo di trasmettere contenuti oppure di fare acquisire speciali abilità allo studente.

OLIVETTI



La funzione indicata in alto e' calcolata
per ciascuno dei punti del grafico sullo schermo.

Alcuni esempi concreti

Presentiamo per prima cosa un programma scritto per l'elaboratore M10 della Olivetti che serve per illustrare le operazioni eseguite con i numeri reali visti come coppie di classi contigue.

Lo studente imposta due numeri (interi, razionali, o reali, nel senso che sono, a esempio, esecuzione di operazioni irrazionali come estrazioni di radici) e imposta anche l'operazione da eseguirsi su di essi, scelta tra le quattro operazioni aritmetiche più l'elevamento a potenza.

L'elaboratore presenta anzitutto i due numeri di partenza come coppie di classi contigue, quindi esegue l'operazione scelta tra gli elementi corrispondenti della rappresentazione dei due numeri.

Il risultato delle operazioni viene rappresentato ancora nella forma delle classi contigue: come si può constatare dall'illu-

strazione, si tratta proprio di una coppia di tali classi, cioè di un numero reale.

Presentiamo ora un programma più complesso per permettere di studiare le funzioni. È stato scritto per l'elaboratore M20 della Olivetti.

Lo studente deve scegliere la funzione o le funzioni da studiare e deve impostarle nel programma secondo determinate regole. Deve quindi scegliere l'intervallo (x minimo-x massimo) entro il quale la funzione viene rappresentata.

L'elaboratore rappresenta la funzione entro l'intervallo indicato. A questo proposito, lo studente ha la possibilità di variare l'intervallo di rappresentazione della funzione, ingrandendolo o rimpicciolendolo: per ingrandirlo può servirsi di due cursori (figura c) e in questo caso viene rappresentata solo la parte compresa tra i due cursori (figura d).

Lo studente può inoltre utilizzare un cursore a croce che si sposta a piacere sullo schermo, mentre sul fondo dell'elabo-

Primo numero, H1:

0	1
3.7	1.9
1.56	1.88
1.81	1.81
1.4619	1.813
1.16777	1.81277
1.1127967	1.812796
1.10278107	1.81278107
1.100781003	1.812781003
1.10027810003	1.81278100003
1.1000278100003	1.8127810000003

Primo numero, H2:

9	10
18	20
18.9	19.1
18.99	19.01
18.999	19.001
18.9999	19.0001
18.99999	19.00001
18.999999	19.000001
18.9999999	19.0000001
18.99999999	19.00000001
18.999999999	19.000000001

Primo numero, H3:

1	13
1.4	2.6
1.53	2.55
1.542	2.544
1.5429	2.5431
1.54299	2.54301
1.542999	2.543001
1.5429999	2.5430001
1.54299999	2.54300001
1.542999999	2.543000001
1.5429999999	2.5430000001
1.54299999999	2.54300000001

Secondo numero, H1:

2	6
4.1	4.4
4.37	4.1
4.356	4.36
4.3746	4.374
4.37467	4.37461
4.374666	4.37466
4.3746667	4.3746667
4.37466667	4.37466667
4.374666667	4.374666667
4.3746666667	4.3746666667
4.37466666667	4.37466666667

Secondo numero, H2:

3	7
6	14
6.6	13.7
6.66	6.76
6.666	6.667
6.6666	6.6667
6.66666	6.66667
6.666666	6.666667
6.6666666	6.6666667
6.66666666	6.66666667
6.666666666	6.666666667
6.6666666666	6.6666666667

Secondo numero, H3:

5	9
5.1	4.2
6.56	4.67
6.764	4.667
6.7666	4.6667
6.76666	4.66667
6.766666	4.666667
6.7666666	4.6666667
6.76666666	4.66666667
6.766666666	4.666666667
6.7666666666	4.6666666667
6.76666666666	4.66666666667

Esultato operazioni, H1:

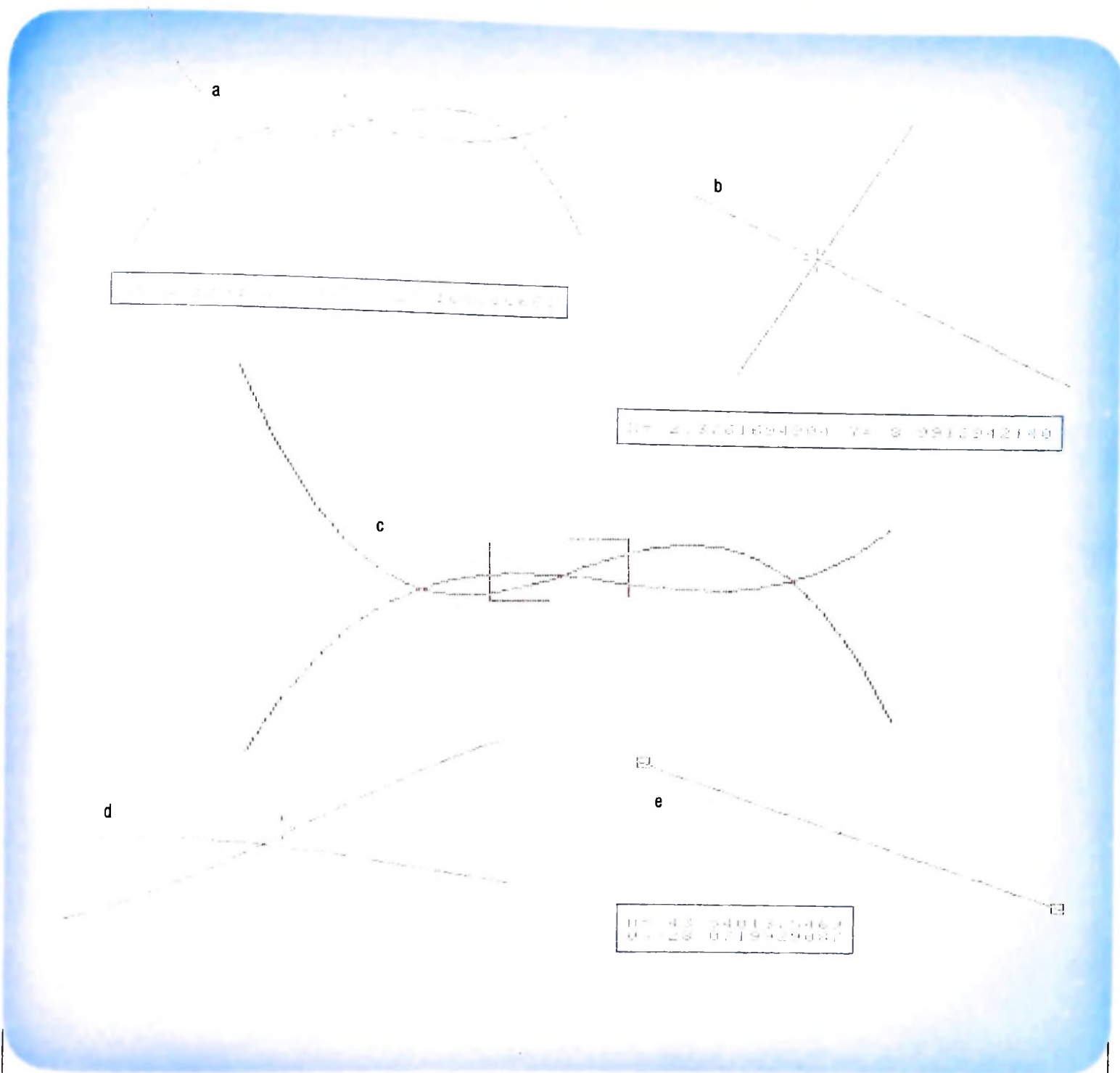
1	10
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000

Esultato operazioni, H2:

1	10
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000

Esultato operazioni, H3:

1	10
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000
1.1000000000000000	1.1000000000000000



ratore compare il valore delle coordinate della posizione assunta di volta in volta (figura a).

Ciò è utile, ad esempio, per la soluzione grafica di un sistema di due equazioni. È sufficiente trasformare le equazioni in funzioni e rappresentarle graficamente. Per questo bisogna ingrandire parecchie volte il diagramma nei punti di intersezione, usando gli appositi cursori, quindi portare il cursore nel punto esatto di intersezione e leggere le coordinate. I risultati sono assolutamente notevoli, purché il grafico sia sufficientemente ingrandito. Ecco la rappresentazione di un esempio. Le due funzioni sono:

$$y = -x^3 + 8x^2 - 10x + 1$$

$$y = -0.5x^3 - 5x^2 + 12x + 2$$

La soluzione, come è possibile vedere anche dal grafico (figura b), è:

$$x = 2.3762$$

$$y = 8.9913$$

Esiste un'ulteriore possibilità per l'allievo, e cioè quella di determinare l'equazione della tangente alla curva rappresentata in un punto, praticamente in un piccolo intervallo.

Premendo certi tasti, sul video compare un piccolo cursore quadrato che lo studente sposta. Se il cursore viene posto sul grafico prima a un estremo dell'intervallo e poi all'altro il calcolatore fornisce il valore di M e di Q della retta passante per i centri dei due cursori (figura e).

Un esempio di geometria analitica

Un altro programma, scritto anch'esso per l'elaboratore M20, si propone di illustrare la geometria analitica solitamente insegnata nella scuola media superiore.

Esso si compone di quattro parti. La prima è una semplice illustrazione delle curve coniche solitamente studiate: la circonferenza (figura *b*), la parabola con asse parallelo all'asse delle *y* (figura *c*), l'iperbole equilatera riferita agli asintoti (figura *a*). Si vuole offrire la possibilità allo studente di vedere facilmente rappresentate le curve che sta studiando e di cogliere il ruolo che i diversi parametri hanno nella rappresentazione stessa.

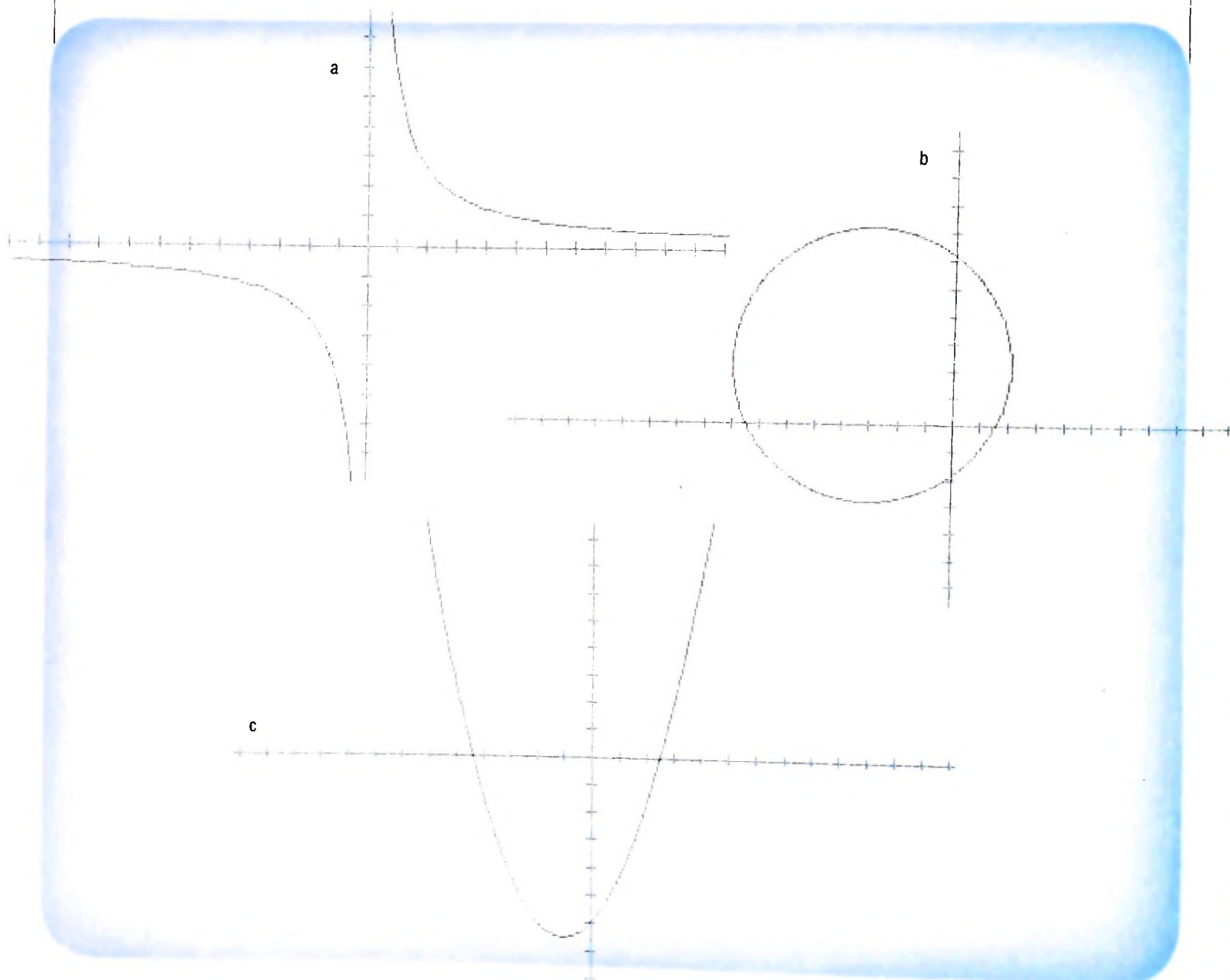
La seconda parte presenta esercizi sulle coniche, soprattutto intersezioni di figure diverse. È lo studente stesso che sceglie il tipo di esercizio, che assegna i parametri delle diverse fun-

zioni. A sua scelta può chiedere al calcolatore la rappresentazione delle figure dell'esercizio, come aiuto alla soluzione.

L'esercizio viene risolto separatamente dallo studente e dall'elaboratore, dopo di che si confrontano i risultati. Se lo studente non riesce a risolverlo può chiedere i risultati direttamente all'elaboratore e quindi riprovare.

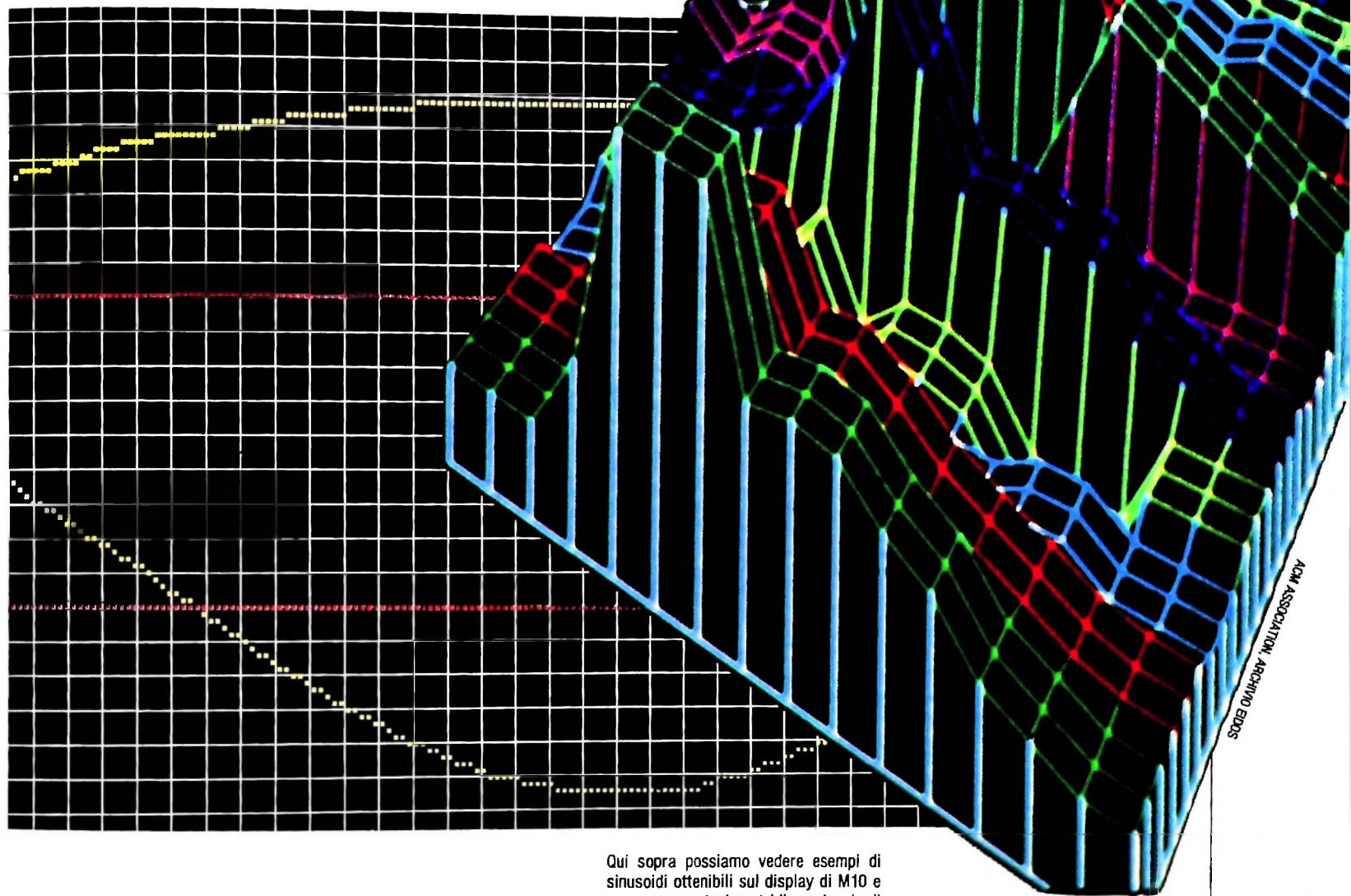
La terza parte consiste nel controllo dell'apprendimento: presenta ancora gli stessi esercizi della parte precedente, ma i dati numerici sono assegnati dall'elaboratore in modo pseudo-casuale, secondo certi intervalli prefissati. Inoltre l'elaboratore prende nota delle risposte dell'allievo e registra i risultati positivi e negativi.

L'ultima parte, come la prima, è riservata alla rappresentazione delle coniche, ma senza le limitazioni precedenti. A seconda del parametro impostato dallo studente, l'elaboratore rappresenterà i vari tipi di coniche.



GRAFICI E FUNZIONI

Come si può visualizzare sul display di M10 il grafico di una funzione.



Qui sopra possiamo vedere esempi di sinusoidi ottenibili sul display di M10 e una rappresentazione tridimensionale di un gruppo a due variabili.

Molti sono i campi nei quali hanno trovato applicazione i grafici. Per la loro grande capacità di comunicare informazioni in maniera visiva, hanno trovato il terreno adatto alla loro utilizzazione praticamente in tutti i settori: dal didattico al finanziario, dal medico all'artistico.

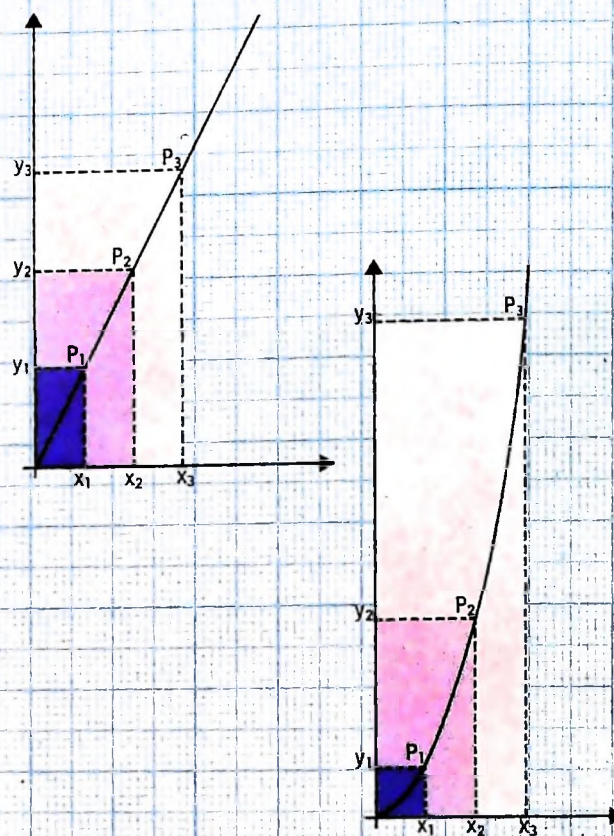
In questa lezione ci occuperemo di come impostare il problema di disegnare un grafico.

Capita sempre più spesso di sentire fare riferimento ai grafici, da quelli matematici a quelli economici e finanziari. Un unico concetto ne lega i vari tipi: costituiscono tutti un valido metodo di visualizzazione dell'andamento di una variabi-

le in funzione di un'altra. Alcune relazioni fra variabili che si possono visualizzare in grafici sono date, per esempio, dall'incremento del raccolto all'aumentare della quantità di fertilizzante usato, oppure dalla diminuzione delle riserve idriche comunitarie all'aumentare del consumo pro capite. Per disegnare un grafico bisogna avere quindi due variabili, di cui una dipende dall'altra (per es., la quantità di raccolto dipende dalla quantità di fertilizzante, la quantità di riserve idriche dipende dal consumo) e un'adeguata unità di misura per quantificare i risultati, evidenziabili poi su una coppia di assi cartesiani.

I grafici nel piano cartesiano

Disegnando su un piano un sistema di riferimento si può esprimere ogni curva geometrica con equazioni algebriche, e quindi trasformare un problema di geometria in uno di algebra; vediamo come si procede. Tracciamo sul piano due rette perpendicolari, gli assi x e y ; associamo ad ogni punto di ciascun asse la sua distanza dall'incrocio degli assi, l'origine. Si può misurare questa distanza in qualsiasi unità, ma a ogni punto deve corrispondere un numero e viceversa. Se prendiamo un punto qualsiasi del piano, P_1 , a P_1 corrispondono due numeri, x_1 e y_1 , coordinate dei punti più vicini a P_1 sui due assi. Descriviamo con equazioni algebriche una figura geometrica semplice, una retta. Supponiamo allora che x_1 sia, per esempio, il doppio di y_1 . Prendiamo poi un punto P_2 di coordinate x_2 e y_2 , in modo che sia anche $y_2 = 2x_2$, e tracciamo la retta per P_1 e P_2 . Se ora prendiamo un altro punto P_3 sulla retta tra le sue coordinate ci sarà la stessa relazione: $y_3 = 2x_3$. Ecco quindi come si può descrivere una retta algebricamente, indicando un'equazione che è caratteristica di tutti i suoi punti: $y = 2x$. Con un foglio di carta a quadretti e una matita è facile vedere che anche $y = 3x$, $y = 4x$ e in generale $y = ax$ sono rette per ogni a costante. Lo stesso vale per $y = 2x + 1$, che è la parallela a $y = 2x$ alzata di un quadretto, e in generale $y = ax + b$ per ogni coppia di costanti a e b : sono tutte rette del piano. Se ora scriviamo un'altra equazione algebrica semplice, per esempio $y = x^2$, e tracciamo la linea corrispondente, otterremo un'altra curva, una parabola. È facile vedere (ma qui il calcolatore è più comodo della carta) che, proprio come ogni equazione in x alla prima potenza corrisponde a una retta, ogni equazione in x alla seconda potenza, cioè ogni equazione di secondo grado $y = ax^2 + bx + c$ con a , b e c costanti qualsiasi, è una parabola.



La rappresentazione di funzioni

L'ambiente matematico, attraverso lo studio delle "funzioni", è quello che ha sintetizzato il concetto di grafico. Ricordiamo che in matematica viene chiamata "funzione" una relazione che intercorre fra una o più variabili, dette indipendenti, e un'altra variabile, detta dipendente. Per esempio, se consideriamo le espressioni matematiche: $y = 2x + 1$; $y = x$; $y = \sin(x)$, queste definiscono in maniera inequivocabile quale sarà il valore di y una volta assegnato un certo valore alla x . Ossia, per ogni valore che assegnamo alla x , ci sarà sempre un solo valore della y che gli corrisponde. Questo concetto matematicamente si scrive $y = f(x)$ e si legge "y è funzione di x". Le espressioni viste sopra sono quindi tre semplici esempi di funzioni a una variabile indipendente, indicata con "x". Dallo studio delle funzioni matematiche emerge poi l'esigenza di avere a disposizione, oltre alla formula analitica della funzione stessa, anche una sua rappresentazione grafica, che gli permetta di rilevarne visivamente il comportamento. Per fare ciò, si assegna un asse di riferimento a ognuna delle variabili. Per esempio, nelle funzioni a una variabile, tradizionalmente si assegna alla variabile indipendente x l'asse orizzontale (*ascisse*) e alla variabile dipendente y l'asse verticale (*ordinate*). Ogni asse è concettualmente infinito ed è costituito da infiniti punti. Ogni punto è allora identificabile con un numero reale, ovvero un numero che può essere sia intero

che decimale; viceversa, ogni numero reale può essere rappresentato sull'asse da un punto posto in una posizione ben precisa. Il punto in cui gli assi si incrociano si chiama "origine" e assume il valore 0. Si assegnano allora valori crescenti, positivi o negativi, ai punti dei due assi che cadono, rispetto al punto 0, a destra o a sinistra (asse x), oppure sopra o sotto (asse y). Se assegnamo alla x della nostra funzione un certo valore, ad esso corrisponderà il valore $y = f(x)$. Ad ogni valore di x siamo così in grado di associare un unico valore per y . La coppia di numeri $(x, f(x))$ determina un punto nel piano. L'insieme di tutte queste coppie di punti descrive il grafico della funzione $y = f(x)$. Per esempio, la funzione $y = \sin(x)$ significa che, assegnato ad x un certo valore, y assumerà il valore determinato dalla espressione $\sin(x)$.

La visualizzazione su display

Quindi, gli ingredienti per disegnare un grafico sono: una legge ben definita che associ ad ogni valore di x un unico valore di y e un sistema di riferimento che permetta di visualizzare i risultati.

Per le nostre applicazioni sull'M10, dovremo però fare i conti anche con la capacità di risoluzione dello schermo. Infatti, fissati due assi di riferimento, essi risultano costituiti da un numero finito di punti indirizzabili, distribuiti in maniera di-

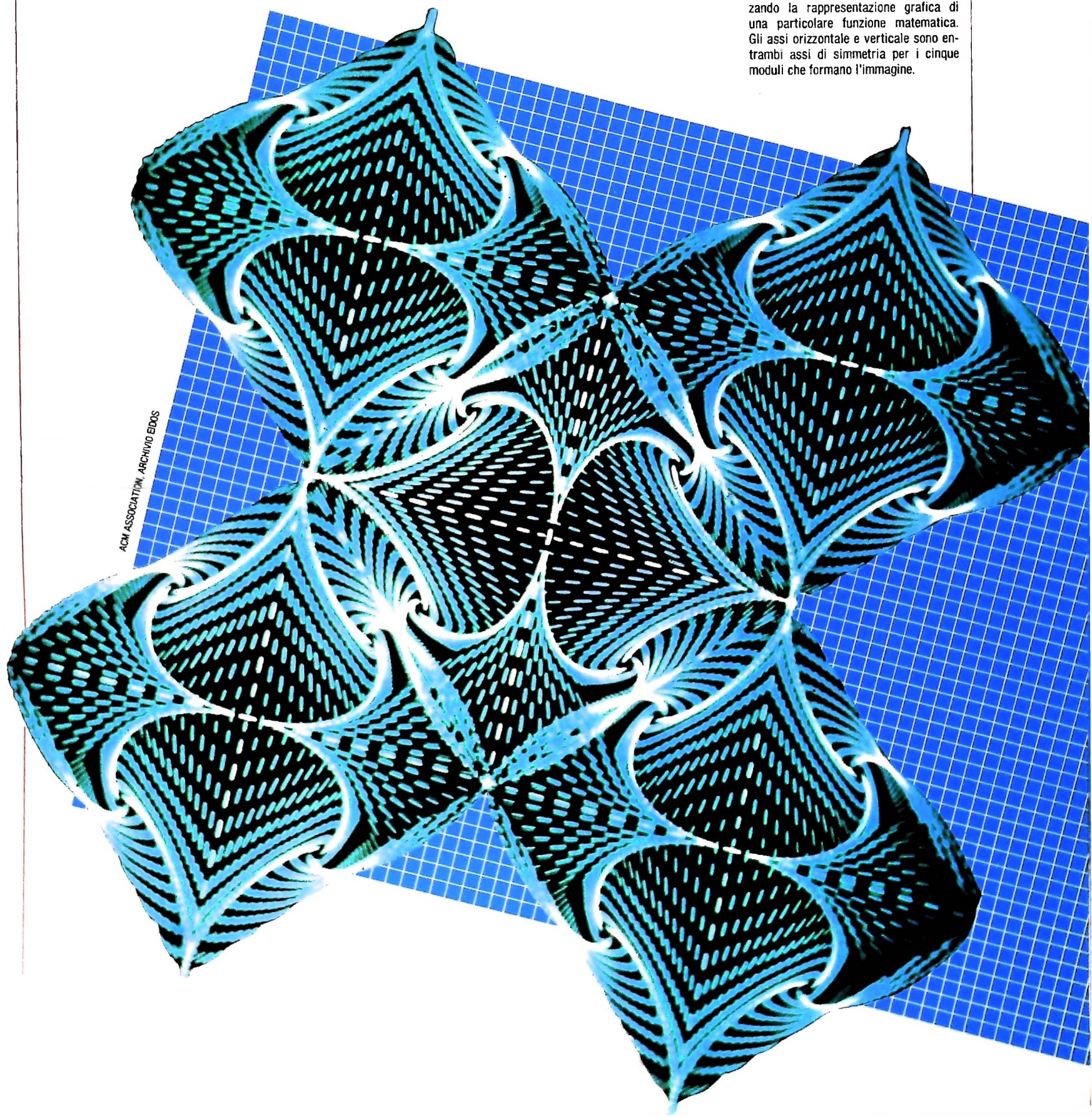
scarta sullo schermo. Ossia, fra i pixel di coordinate (130,32) e (131,32) non esiste, per esempio, il pixel (131.5,32). Ciò si ripercuote sul disegno del grafico della funzione considerata, che risulterà necessariamente un'approssimazione del grafico ideale, che è definito invece anche per i numeri decimali.

Questa limitazione vale per qualsiasi computer digitale, che scompone sempre in un numero finito di punti, uno per ogni locazione di memoria, le immagini che elabora.

Un altro tipo di problema è quello legato alla necessità di studiare l'andamento di una funzione, ovvero il suo grafico,

fra un certo intervallo di valori, che può essere maggiore o minore delle dimensioni dello schermo. Si deve allora effettuare una trasformazione di scala fra l'intervallo e il dispositivo di visualizzazione del nostro computer. Un esempio di trasformazione di scala viene fornito nel programma che esegue il grafico della funzione $y = \sin(x)$ riportato nella pagina seguente. È anche interessante segnalare come le soluzioni grafiche di alcuni tipi di equazioni matematiche diano origine a motivi grafici particolarmente gradevoli per le loro simmetrie e complessità di forme.

Immagine prodotta per simmetria utilizzando la rappresentazione grafica di una particolare funzione matematica. Gli assi orizzontale e verticale sono entrambi assi di simmetria per i cinque moduli che formano l'immagine.





PERSONAL COMPUTER OLIVETTI M20

PERSONAL COMPUTER OLIVETTI. UNA FAMIGLIA CHE CRESCE

Nella famiglia di personal computer Olivetti M 20, il modello M 20D dispone di una memoria 30 volte più grande rispetto al modello base, ed è anche capace di gestire una rete di M 20 collegati fra loro, rendendo sempre più potente e coordinata la capacità di lavoro degli uffici.

I personal computer Olivetti sono dunque una famiglia che cresce e che, per rispondere a esigenze diverse, offre differenti capacità di memoria e un'ampia scelta di sistemi operativi (MS-DOS, CP/M-86, PCOS, UCSD-P). Sono personal computer con tecnolo-



gia a 16 bit e capacità di "communication", progettati quindi per essere validi anche domani e per integrarsi agevolmente nelle strutture di elaborazione dati e di automazione dell'ufficio presenti e future. Perché Olivetti protegge i vostri investimenti in macchine e programmi.

Con M 20 in ufficio il lavoro individuale diventa più semplice e produttivo. Infatti anche i personal computer sono espressione di quel primato nel design e nell'ergonomia che è ormai parte consolidata del successo Olivetti negli uffici di tutto il mondo.

Anche in leasing con Olivetti Leasing S.p.A.

MS-DOS, marchio registrato Microsoft Inc
CP/M-86, marchio registrato Digital Research
UCSD-P System, marchio registrato Regent
of the University of California

olivetti
l'universo della comunicazione

Banco di Roma Aperta

LE NUOVE RISPOSTE DEL BANCO DI ROMA.

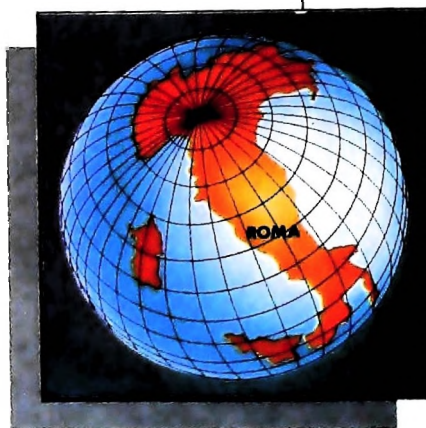


*Vorrei avere
un rapporto più diretto
con la mia banca...*

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.