

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

24 CORSO PRATICO COL COMPUTER

421768

diretto da **GIANNI DEGLI ANTONI**

è una iniziativa
FABBRI EDITORI

in collaborazione con
BANCO DI ROMA

e **OLIVETTI**

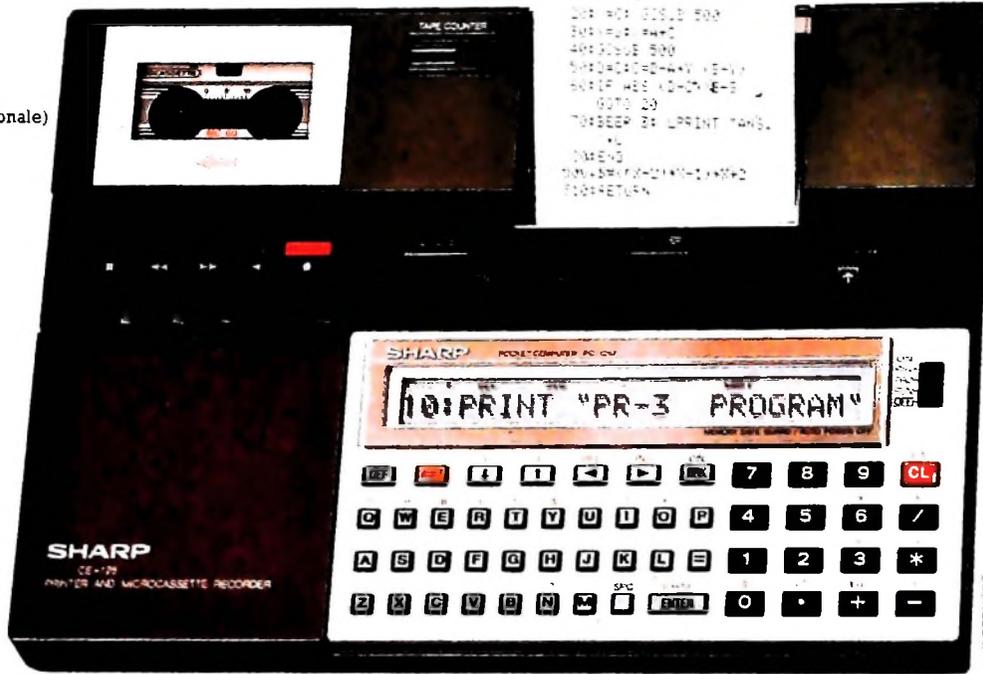


**FABBRI
EDITORI**

Un nuovo Computer tascabile da Sharp.

La giusta combinazione tra semplicità d'uso e disponibilità di programmi, tra superiorità del sistema e reale portatilità

Il PC-1251 costa solo
L. 299.500 + IVA
(Il sistema
stampante/registratore
a microcassette è opzionale)



SHARP
PC-1251

Basic esteso con comandi ed istruzioni estremamente versatili

Il Basic realizzato dalla Sharp comprende dei nuovi comandi che vi consentono le più vaste applicazioni. Il comando DIM, ad esempio vi permette di trattare matrici bidimensionali di variabili. Si possono effettuare operazioni con le stringhe. Avete a disposizione il comando IN-KEYS per effettuare inserimenti da tastiera senza fermare il programma. Grazie al BASIC esteso questo Computer, il più tascabile di tutti, appartiene alla categoria dei Personal Computer: ideale per neofiti, versatile per gli esperti.

C-MOS e CPU a 8-bit per rapide elaborazioni

Per mezzo del CPU a 8 bit si ottiene una veloce elaborazione dei dati, mentre l'utilizzazione dei C-MOS permette un funzionamento con basso consumo di corrente per una più lunga durata delle batterie.

24 KBytes di ROM, 4,2 KBytes di RAM

Nonostante le sue dimensioni il PC-

1251 ha 24 KBytes di ROM e 4,2 di RAM, di cui 3,7 a disposizione dell'utente. Con questa grande capacità di memoria potete scrivere o caricare da nastro anche lunghi programmi.

Protezione della memoria

Le pile mantengono l'alimentazione alle memorie RAM proteggendole anche a macchina spenta. Grazie a questa caratteristica i programmi che avete in memoria sono sempre pronti per essere utilizzati.

Funzione Pass

Potete assegnare un codice segreto ad ogni programma che limita l'accesso alle sole persone che lo conoscono.

Visore a 24 caratteri con matrice a punti 5x7

Il visore è in grado di visualizzare contemporaneamente fino a 24 caratteri, ognuno dei quali chiaramente rappresentato per mezzo di una matrice da 5x7 punti. La luminosità del display è regolabile.

Stampante/Registratore a microcassette CE-125 (opzionale)

La CE-125 è una unità speciale comprendente sia la stampante che il registratore a microcassette, realizzata per funzionare con il PC-1251. Vi consente sia una stampa veloce che il trasferimento su nastro dei programmi. Le ridotte dimensioni del sistema consentono d'avere la potenza del PC-1251 e la versatilità della CE-125 sempre e ovunque con voi.

Registratore a microcassette

Non è necessario avere un registratore esterno quando dovete "salvare" o ricaricare un programma. Usate semplicemente una microcassetta e tutto è risolto. Questo sistema elimina gli errori di trasferimento.

I programmi sono la chiave per il successo di un Computer. Col PC-1251 siete già sulla giusta strada. Perché? Perché potete utilizzare sia i già collaudati programmi del PC-1211 che i nuovi programmi realizzati per il PC-1251 o quelli che voi stessi realizzerete grazie alla semplicità del BASIC.

Elenco dei programmi inseriti nel manuale del PC-1251

- Radici di un'equazione secondo il metodo di Newton
- Percentuale, varianza, deviazione standard
- Intersezioni tra cerchi e rette
- Calcolo del numero dei giorni
- Pratica con la tastiera
- Atterraggio morbido
- Prova di memoria
- Caccia alla pulce
- Rotazione doppia

Elenco dei programmi inseriti nel manuale della CE-125

- Prodotto di matrici
- Integrazione numerica secondo il metodo di Simpson
- Coefficiente di correlazione e regressione lineare
- Istogrammi
- Conversione Δ - Y e Y - Δ
- Curva clotoidale
- Incroci
- Ordinamento
- Calcolo delle rate

- Bioritmo (semigrafico)
- Ritratto stampato

Programmi su microcassette CE-12A/B/C

Le microcassette sono disponibili opzionalmente
CE-12A - 19 programmi di statistica e giochi
CE-12B - 20 programmi di ingegneria e giochi
CE-12C - 20 programmi di matematica e giochi

Direttore dell'opera
GIANNI DEGLI ANTONI

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

UMBERTO ECO
Ordinario di Semiotica presso l'Università di Bologna

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

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

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

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

Testi
Etnoteam (ADRIANA BICEGO), Eidos (TIZIANO BRUGUETTI), GOFFREDO HAUS, GIANCARLO MAURI, CLAUDIO PARMELLI

Tavole
Logical Studio Communication
Il Corso di Programmazione e BASIC è stato realizzato da Etnoteam S.p.A., Milano
Computergrafia 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
RENATA FRIGOLI
LUCIA MONTANARI

Corso Pratico col Computer - Copyright © sul fascicolo 1984 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Einaudi S.p.A., Milano - Copyright © sull'opera 1984 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Einaudi 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 Forzezza 27 - tel. 2526 - Milano - Pubblicazione periodica settimanale - Anno I - n. 24 - 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 - 20138 MILANO - Via P. Colonna, 37

SHARP

MESSAGGI E TRASMISSIONE

I messaggi "spezzati" vengono trasmessi in tre modi: simplex, half-duplex, full-duplex.

Formati di messaggio

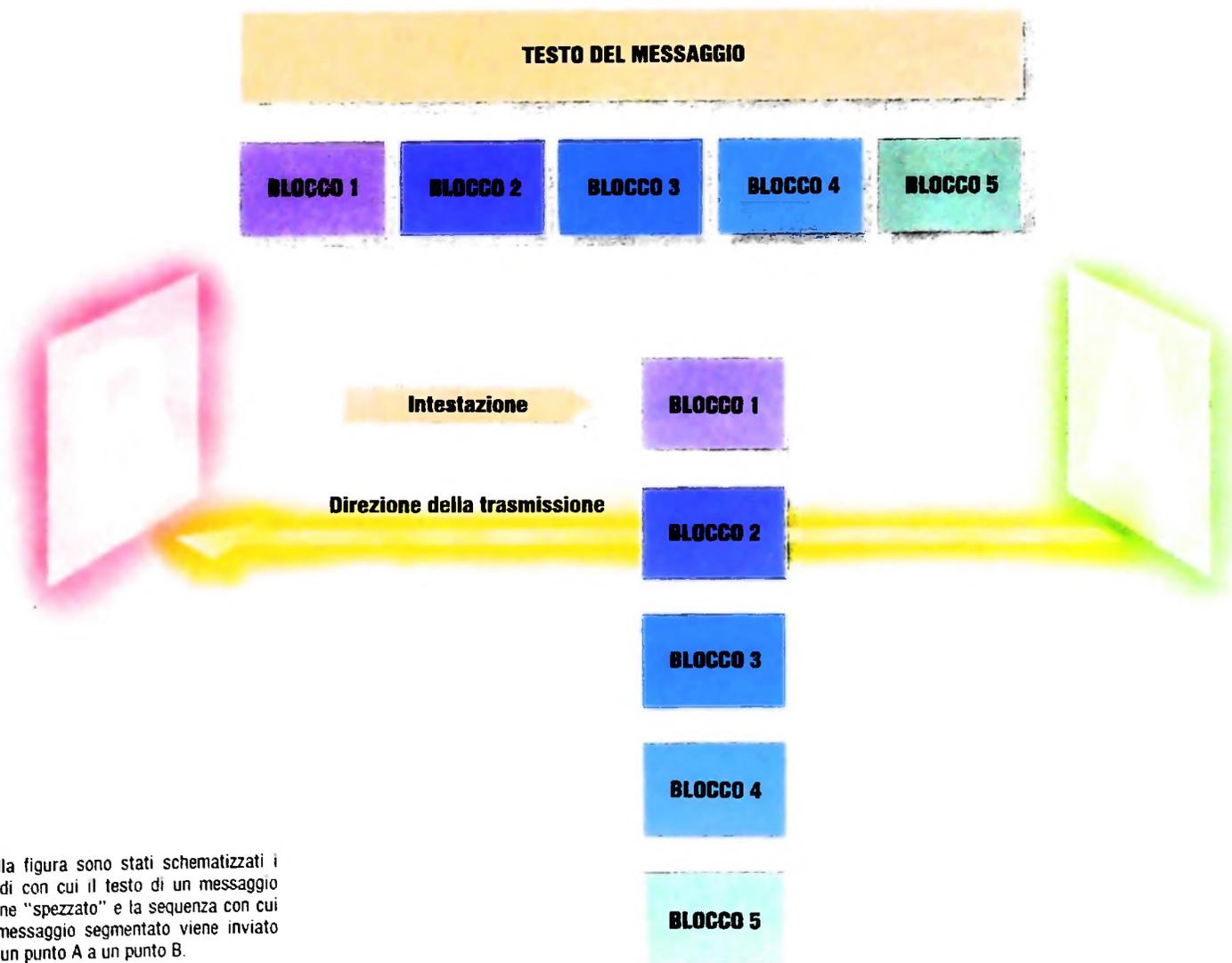
Il contenuto di un messaggio viene denominato *testo*. Quando i messaggi sono lunghi, si spezza il messaggio stesso in un certo numero di blocchi di testo (si veda l'illustrazione qui sotto). Questi blocchi vengono trasmessi uno alla volta lungo la linea di trasmissione e, in funzione del sistema utilizzato, possono essere preceduti da qualche informazione di intestazione (*message header*) contenente informazioni relative all'indirizzamento e/o informazioni amministrative relative al testo del messaggio.

Ad esempio, il message header può identificare l'indirizzo

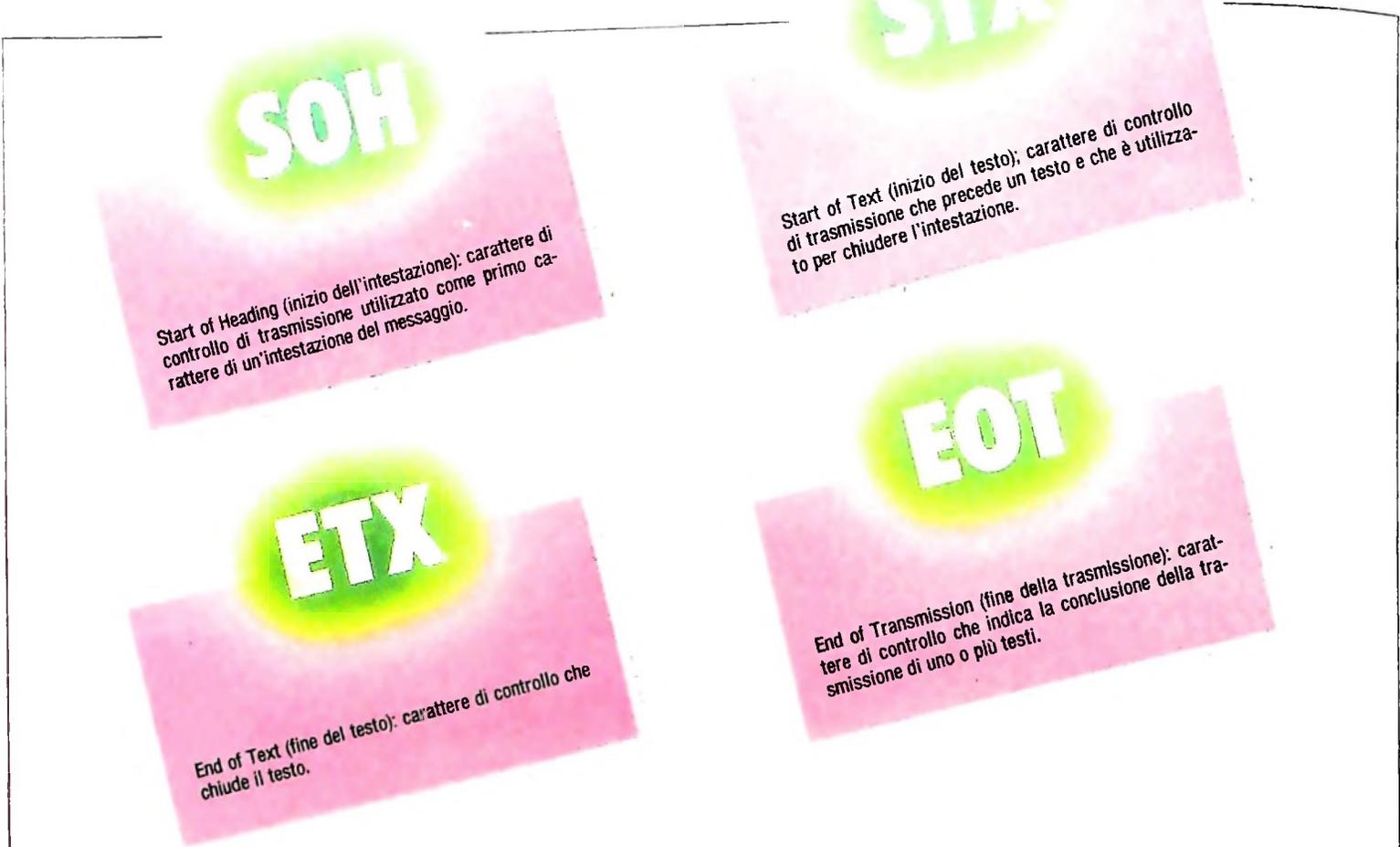
del terminale d'origine, l'autore del messaggio, le priorità, la data e l'ora di spedizione del messaggio e così via.

Nella figura di questa pagina, in basso, è visualizzata una tipica sequenza di messaggio, costituita da cinque blocchi di testo: il primo blocco viene trasmesso con intestazione che descrive in qualche modo il messaggio, mentre i blocchi successivi sono inviati senza intestazione, supponendo quindi che il terminale destinatario sia in grado di collegare fra di loro e nel giusto ordine i vari blocchi di testo.

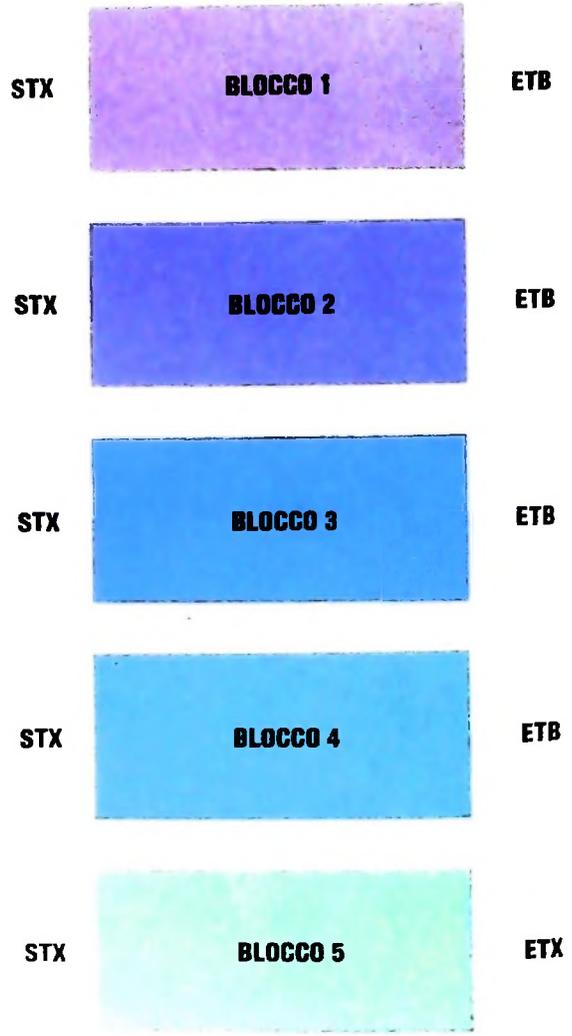
Vediamo nei riquadri delle pagine seguenti i caratteri di controllo di trasmissione, così come sono schematizzati nella tabella dei codici ASCII che è stata riportata a pagina 358.



Nella figura sono stati schematizzati i modi con cui il testo di un messaggio viene "spezzato" e la sequenza con cui il messaggio segmentato viene inviato da un punto A a un punto B.



Schema dell'impiego dei caratteri di controllo quando il messaggio viene trasmesso a segmenti: la sigla ETB avvisa il ricevente che il segmento del messaggio è terminato, mentre ETX indica la fine dell'intero messaggio. A sua volta SOH informa che ha inizio un messaggio, mentre STX avvisa dell'inizio dei diversi segmenti.



Descriviamo ora l'impiego di alcuni controlli di trasmissione per identificare le parti che costituiscono un messaggio multisegmento; facciamo riferimento alla figura qui a lato.

Il carattere SOH è collocato davanti all'intestazione del messaggio e questo segnala al destinatario che l'informazione che segue il carattere SOH deve essere interpretata come informazione appartenente all'intestazione. In modo analogo il carattere STX, situato all'inizio del testo, informa il destinatario che tutto ciò che segue STX è il testo del messaggio; in questo esempio lo STX è anche utilizzato per chiudere l'intestazione.

Poiché si sta inviando un messaggio multisegmento, il carattere ETB chiuderà il primo blocco di dati e questo avvertirà il destinatario che il segmento di messaggio è completo, ma anche che ci sono altri blocchi.

Con la stessa modalità verranno trattati i blocchi 2, 3 e 4. Infine, il blocco 5 inizierà con un carattere STX, ma verrà chiuso da ETX, che informa il destinatario dell'avvenuta trasmissione dell'intero messaggio.

ENQ

Enquiry (interrogazione): carattere di controllo della trasmissione con cui si richiede una risposta da una stazione remota. Tale risposta può essere costituita dall'identificazione della stazione e/o dallo stato della stessa.

ACK

Acknowledgement (riconoscimento): carattere di controllo inviato dal destinatario come risposta affermativa a un'interrogazione del mittente.

DLE

Data Link Escape (modifica del collegamento dati): carattere di controllo che muta il significato di un limitato numero di successivi caratteri contigui. Viene impiegato per fornire funzioni supplementari di controllo della trasmissione dati. Nella sequenza DLE possono essere usati solo caratteri grafici e di controllo della trasmissione.

NAK

Negative Acknowledgement (riconoscimento negativo): carattere di controllo trasmesso dal destinatario come risposta negativa al mittente.

SYN

Synchronous Idle (sincronizzazione in condizioni di inattività): carattere di controllo usato dai sistemi di trasmissione sincroni in assenza di qualsiasi altro carattere (in stato *idle*, cioè di inattività) per offrire un segnale dal quale si possa ottenere o mantenere il sincronismo tra stazioni terminali.

ETB

End of Transmission Block (fine del blocco in trasmissione): carattere di controllo impiegato per indicare la fine della trasmissione di un blocco di dati quando i dati stessi siano stati divisi in blocchi ai fini della trasmissione.

Dato che non tutti i terminali consentono di chiudere i blocchi segmentati, con ETB, quando si invia un messaggio multisegmentato, si devono inserire in ogni blocco di testo informazioni che permettano al destinatario di poter collegare correttamente i blocchi di testo per poter ricomporre il messaggio completo.

Tipi di trasmissione

La *linea di trasmissione* è il mezzo che porta i messaggi in un sistema di comunicazione dei dati: solitamente tale linea proviene dalla rete telefonica e prende quindi il nome di *linea telefonica* (o *linea* o *linea dati*).

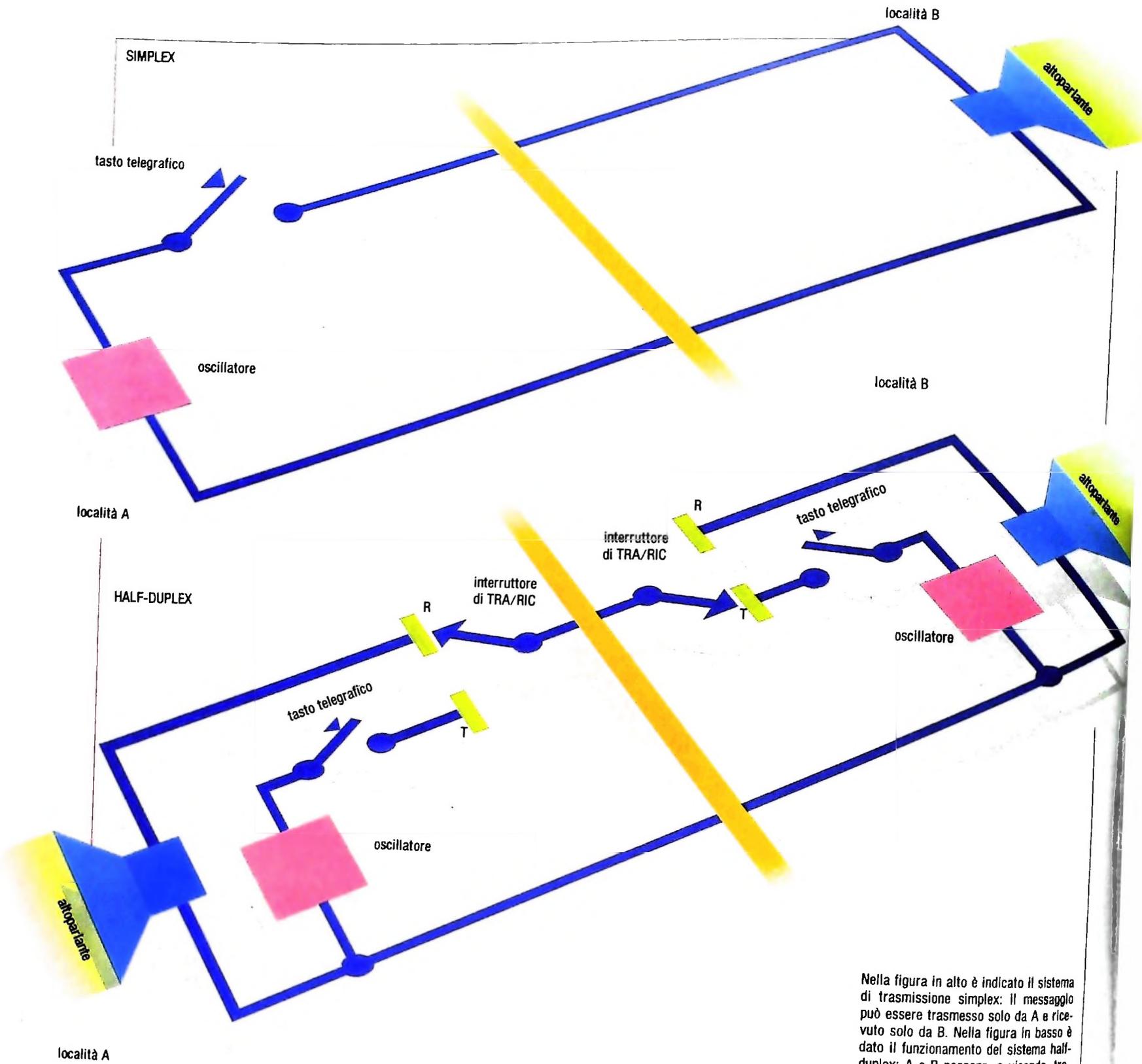
I tipi di trasmissione sono tre:

- a) SIMPLEX (a una via)
- b) HALF-DUPLEX (a via alternata)
- c) FULL-DUPLEX (a due vie)

a) SIMPLEX

Nella trasmissione a una via, l'informazione è trasmessa in una sola direzione e tale direzione è determinata dalle caratteristiche dei dispositivi che si trovano ad ogni capo del canale; i ruoli del trasmettitore e del ricevente sono fissi.

La radio e la televisione sono esempi di trasmissione simplex, i segnali viaggiano dalla stazione trasmittente fino a quando vengono raccolti dagli appositi apparecchi riceventi i



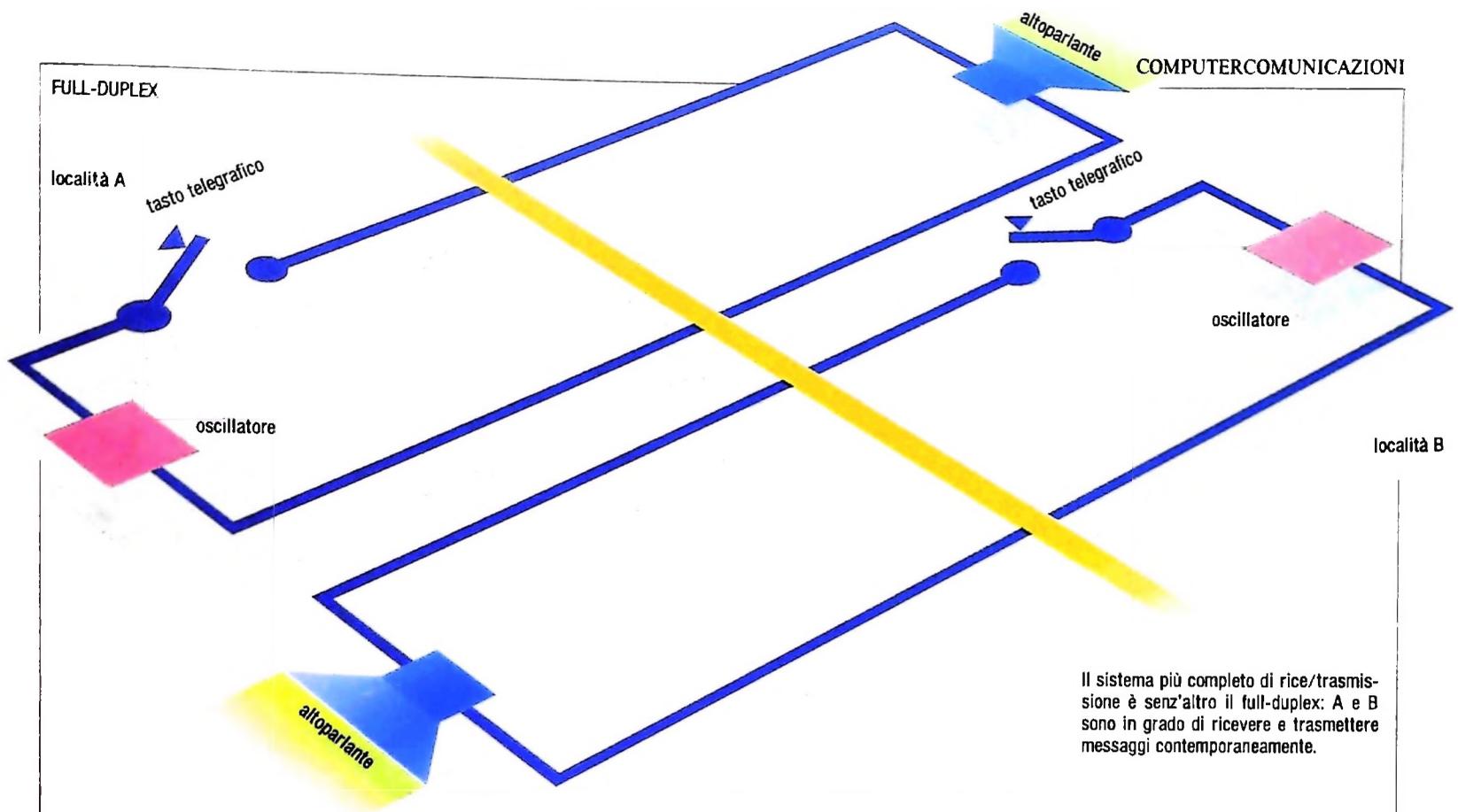
Nella figura in alto è indicato il sistema di trasmissione simplex: il messaggio può essere trasmesso solo da A e ricevuto solo da B. Nella figura in basso è dato il funzionamento del sistema half-duplex: A e B possono, a vicenda, trasmettere e ricevere un messaggio.

quali non possono inviare, a loro volta, informazioni perché non sono stati progettati e costruiti per trasmettere così come l'emittente non è stata progettata e costruita per ricevere informazioni.

Se tentassimo di inviare dati lungo il canale in entrambe le direzioni nello stesso momento otterremmo messaggi alterati ed irriconoscibili. Proviamo a pensare ad un tubo (canale) collegato ad un rubinetto dell'acqua (trasmettente). Aprendo il rubinetto il flusso dell'acqua (informazione) scorrerà in una precisa direzione: si provi ora a collegare l'altro capo del tu-

bo (ricevente) ad un altro rubinetto (trasmettente) e lo si apra: cosa succederà?

La figura in alto in questa pagina schematizza un semplice sistema di comunicazione a una via. Qui si ha una persona nella località A e un'altra nella località B: la persona in A sia fornita di un oscillatore e di un tasto telegrafico mentre quella in B sia fornita di un altoparlante. Colleghiamo le due località con due fili che consentano ad A di inviare informazioni a B premendo il tasto telegrafico, il quale, premuto, chiude il circuito consentendo al segnale dell'oscillatore di attra-



Il sistema più completo di rice/trasmis-sione è senz'altro il full-duplex: A e B sono in grado di ricevere e trasmettere messaggi contemporaneamente.

versare il filo fino all'altoparlante. Quando si rilascia il tasto il segnale viene interrotto.

b) HALF-DUPLEX

Supponiamo di migliorare gli impianti posti alle estremità del canale in modo che questi siano in grado di trasmettere o di ricevere: si potrà quindi inviare informazioni da A verso B e, alla fine del messaggio, trasformare il sistema in modo che la trasmissione avvenga da B verso A.

In questo modo otteniamo flussi di dati alternativi nelle due direzioni e tale sistema prende il nome di *half-duplex* (HDX). Una ricetrasmittente è un sistema HDX; quando si preme il pulsante del microfono si può parlare ma non ascoltare, mentre se si rilascia il pulsante avviene il contrario. Il gioco del ping-pong, ad esempio, è un sistema HDX.

Torniamo allo schema della figura precedente e cerchiamo di trasformarlo in un sistema HDX aggiungendo in A un altoparlante e un interruttore di trasmissione/ricezione e in B un oscillatore, un tasto telegrafico e l'interruttore di trasmissione/ricezione. Collegiamo le parti come indicato nella figura in basso della pagina a fronte, dove B è in modo trasmissione e A in modo ricezione: la persona posta nella località B può comunicare con quella posta nella località A premendo il tasto telegrafico e alla fine del messaggio entrambi possono commutare il proprio interruttore trasmissione/ricezione portandolo in posizione opposta per permettere a A (previa pressione del tasto telegrafico) di comunicare con B.

Osserviamo, ora, che l'operazione di invertire il flusso dei dati richiede una quantità finita di tempo, quella richiesta per il riconoscimento di fine trasmissione e per la commutazione dell'interruttore.

Il tempo di *turn around* (di attesa) del sistema è la somma del

tempo di reazione (che è il tempo richiesto dall'operatore per riconoscere la fine della trasmissione) e del tempo di *turn around* della linea (che è il tempo richiesto per premere fisicamente gli interruttori e prepararsi a trasmettere nella direzione opposta).

c) FULL-DUPLEX

Se il dispositivo terminale ad ogni estremità della linea è capace di trasmettere e di ricevere dati simultaneamente, allora l'intero sistema è in grado di sopportare un flusso dati simultaneo nei due sensi. Tale sistema è conosciuto come sistema *a due vie* o *full-duplex* (FDX).

Ad esempio, una comunicazione telefonica è un sistema FDX, la maggior parte delle strade sono sistemi FDX.

Il modo più semplice per trasformare gli schemi analizzati in FDX è quello di duplicare il circuito *simplex* della pagina a fronte, in modo da ottenere un sistema simile a quello della figura qui sopra. Infatti A può premere il tasto telegrafico per inviare informazioni nello stesso momento in cui B sta inviando il suo messaggio e viceversa.

Notiamo, tuttavia, che nonostante i canali di comunicazione e i dispositivi terminali siano progettati e sistemati per supportare una trasmissione in *full-duplex*, potremmo non essere in grado di usare il sistema in FDX: questo per limiti imposti dagli operatori A e B.

Infatti solo un operatore molto esperto può essere in grado di interpretare il messaggio in arrivo e trasmettere al tempo stesso il messaggio in partenza: la maggior parte di essi sarebbe in grado di fare alternativamente i due lavori, ma non contemporaneamente.

Ritroveremo ancora il ritardo dovuto ai tempi di reazione, ma il tempo totale di inversione del sistema sarà più breve.

ALGORITMI E MACCHINE DI TURING

Macchine ideali per capire le possibilità e i limiti delle macchine reali.

Nel 1936, gli elaboratori elettronici non esistevano ancora (il primo, ENIAC, sarebbe stato messo in cantiere sette anni dopo e realizzato dieci anni dopo). Eppure, proprio in quell'anno Alan Mathison Turing pubblicò un lavoro che già conteneva, anche se allora nessuno poteva rendersene conto, gli strumenti concettuali fondamentali della nuova scienza che intorno all'uso dei calcolatori si sarebbe sviluppata: l'informatica. In questo lavoro, non solo si forniva una definizione estremamente precisa di una nozione, quella di algoritmo, destinata a diventare centrale nell'informatica, ma si dava già una risposta ad una questione che, di fronte alla enorme velocità di calcolo ed alle eccezionali capacità di memoria degli elaboratori, non ci si può non porre: esiste qualche problema formulabile in termini matematici rigorosi che non solo gli elaboratori attuali, ma nessun elaboratore potrà mai risolvere?

Il lavoro di Turing era un tentativo di risposta ad alcune questioni che proprio negli anni Trenta erano diventate centrali nella logica matematica.

Per millenni, i matematici hanno ideato algoritmi per risolvere problemi concreti, considerando questa una attività del tutto normale: nel 1800 a.C. i matematici babilonesi già conoscevano un eccellente algoritmo per l'estrazione della radice quadrata e nel III secolo a.C. Euclide non avrebbe certo immaginato che oltre duemila anni dopo il metodo da lui ideato per il calcolo del massimo comun divisore tra due numeri sarebbe stato usato più per illustrare il concetto di algoritmo che per il suo scopo originario.

Soprattutto, per millenni i matematici sono stati convinti che per ogni problema matematico è possibile trovare un algoritmo di soluzione, a condizione di possedere abbastanza conoscenze e abbastanza "creatività". Nel 1900, Hilbert affermò, in una allocuzione diventata famosa: "Come esempio del modo in cui si possono trattare questioni fondamentali, sceglierò la tesi che ogni problema matematico possa essere risolto. Noi tutti siamo convinti di ciò."

E, forte di questa certezza, ipotizzò che fosse possibile ridurre la soluzione di un problema matematico alla esecuzione di una procedura "effettiva" o meccanizzabile per verificare se una particolare formula logica sia vera o falsa. Il problema principale con cui cimentarsi nel secolo che allora si apriva

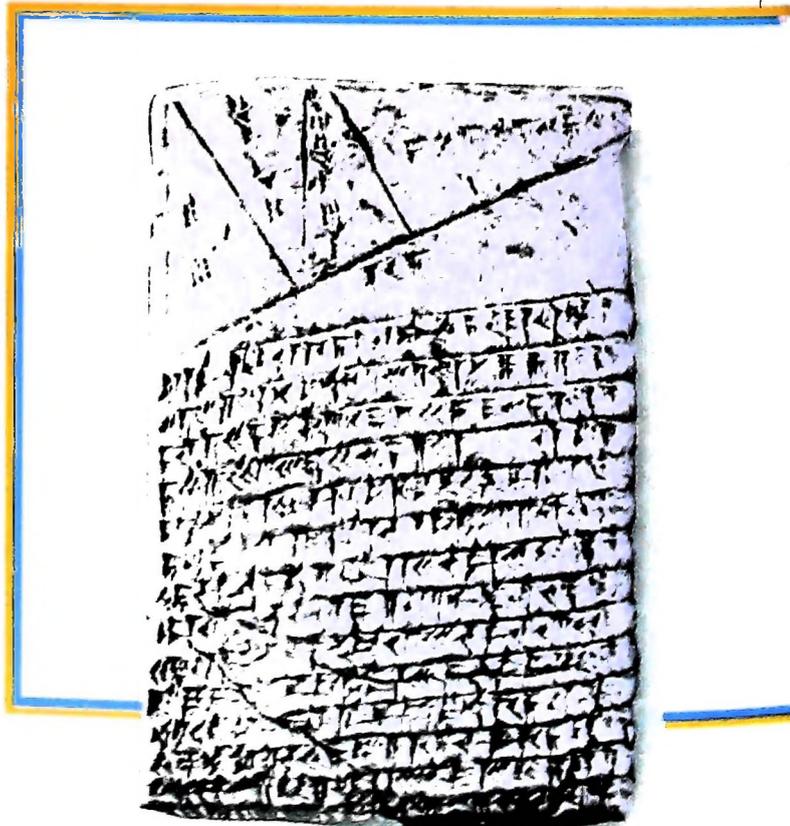
era quindi quello di trovare questa procedura, che avrebbe consentito di ridurre l'intera matematica a una computazione meccanica e, in particolare, di risolvere una lista di problemi ancora aperti che Hilbert elencava.

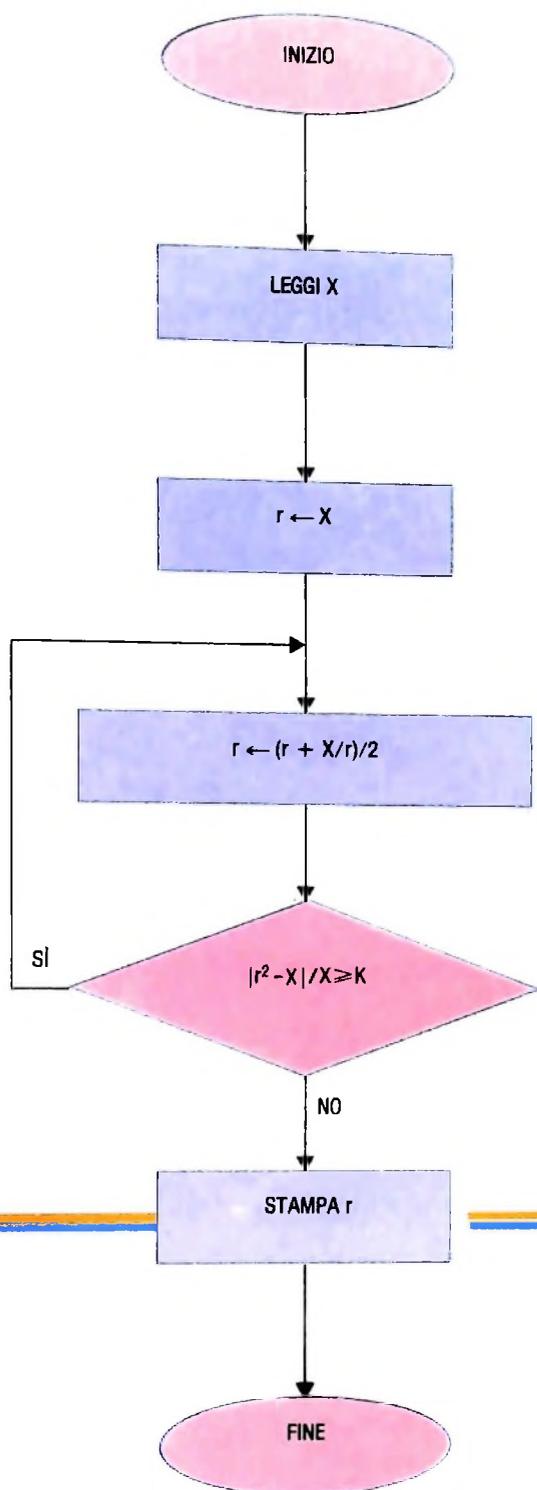
Il nuovo secolo avrebbe invece portato duri colpi alle certezze di Hilbert. Il sospetto che non sempre fosse possibile stabilire la verità o falsità di una formula logica, fatto nascere dalla inutilità dei tentativi di risolvere alcuni dei problemi di Hilbert, divenne infatti certezza nel 1931, quando Gödel dimostrò l'esistenza di formule logiche che esprimono particolari proprietà dei numeri naturali di cui non si può dimostrare in modo effettivo né che sono vere né che sono false.

Il problema della "computabilità effettiva"

Riflettiamo un attimo sul significato di affermazioni negative del tipo di quella sostenuta da Gödel.

Cosa sia un algoritmo da un punto di vista intuitivo, a que-





L' algoritmo ha più di tremila anni

La tavoletta a caratteri cuneiformi documenta l'esistenza di approfonditi studi matematici compiuti dagli astronomi-sacerdoti dell'antica Babilonia. Si tratta di un algoritmo per il calcolo della radice quadrata (1800 a.C. circa), il cui procedimento logico di soluzione è schematizzato qui sopra: K rappresenta l'errore relativo che si ammette nel calcolo. L'uso degli algoritmi per la soluzione di problemi matematici fu ripreso da Euclide e introdotto poi in Occidente nel IX sec. d.C.

sto punto dell'opera, dovrebbe essere già chiaro: esso è un insieme finito di istruzioni che, eseguite nell'ordine indicato, permettono di risolvere un determinato problema. Le istruzioni devono essere abbastanza semplici e ben definite perché un esecutore umano o meccanico possa comprenderle senza ambiguità, eseguirle effettivamente e stabilirne in modo deterministico la successione.

Se noi disponiamo della descrizione di un algoritmo (in senso intuitivo) o di un procedimento di soluzione per un dato problema sulla cui "effettività" non abbiamo dubbi perché siamo in grado di verificarla sperimentalmente, utilizzandola in pratica o direttamente o attraverso una macchina, possiamo chiaramente asserire che il problema è risolubile in modo effettivo pur senza avere una definizione precisa del concetto generale di "metodo effettivo". Ma se vogliamo dimostrare in modo rigoroso che un problema non è risolubile con nessun metodo effettivo, nemmeno con quelli che si possono eventualmente scoprire in futuro, ci occorre ben di più: occorre fornire una definizione rigorosa dei termini "algoritmo" e "procedimento effettivo" o "meccanizzabile" che copra tutte le possibili accezioni particolari che ad essi si possono dare. Il contributo di Gödel, Turing e altri logici riguarda proprio la definizione precisa di questi concetti intuitivi.

La macchina di Turing

La principale intuizione di Gödel è stata senza alcun dubbio quella di ridurre, attraverso una codifica opportuna, la soluzione di un qualsiasi problema matematico al calcolo di una funzione dall'insieme N dei numeri interi non negativi in se stesso, $f: N \rightarrow N$.

Con questo, il campo di studio veniva delimitato con precisione all'insieme delle funzioni da N in N : si trattava di definire la nozione di funzione effettivamente computabile. In seguito, parleremo quindi indifferentemente di "computazione di funzioni" o di "soluzione di problemi".

Le domande che ci poniamo a questo punto sono allora le seguenti:

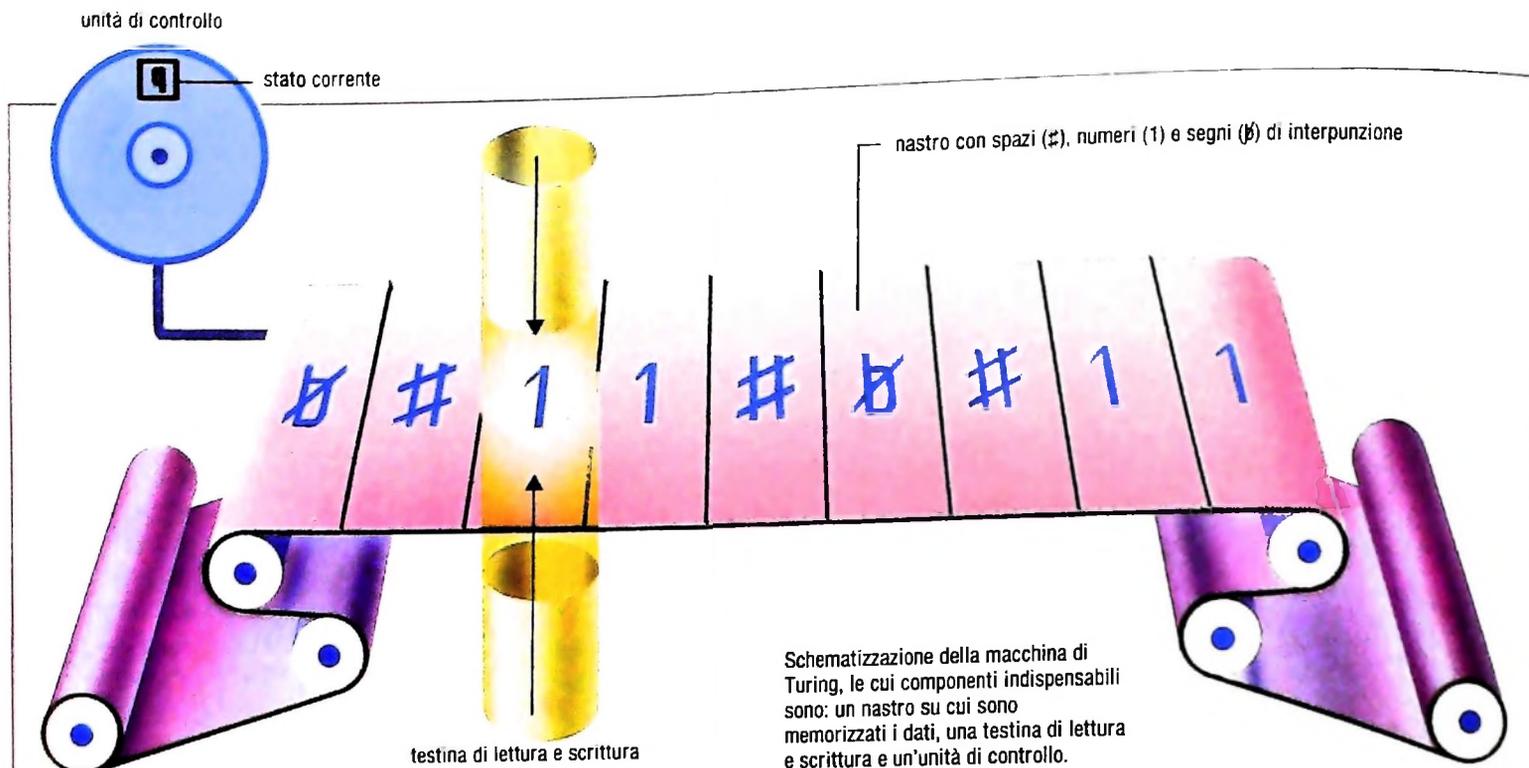
— quando una funzione $f: N \rightarrow N$ può dirsi effettivamente computabile?

— esistono delle funzioni che non sono effettivamente computabili?

Per formalizzare la nozione di effettivamente computabile, Turing ha analizzato il comportamento di un uomo che esegue un calcolo identificandone le azioni elementari.

"Per fare calcoli si scrivono simboli sulla carta...". Occorre quindi anzitutto un foglio di carta, che possiamo pensare suddiviso in riquadri in ognuno dei quali si può scrivere un solo simbolo. Anzi, per semplificare ulteriormente, possiamo pensare che i riquadri siano allineati uno dopo l'altro su un nastro di carta. Inoltre, i simboli che si possono scrivere vengono scelti in un insieme finito (ad esempio, le dieci cifre arabe, gli usuali simboli che indicano le operazioni aritmetiche e l'uguaglianza e così via).

"Il comportamento del calcolatore umano è determinato in ogni momento dai simboli che legge e dallo stato della sua



Schematizzazione della macchina di Turing, le cui componenti indispensabili sono: un nastro su cui sono memorizzati i dati, una testina di lettura e scrittura e un'unità di controllo.

Cosa sono gli algoritmi?

Il persiano Abu Ja'far Mohammed ibn Musa al-Khwarizmi scrisse, circa nell'825 d.C., il trattato *Kitab al jabr w'almuqabala*, sulle regole per trasportare i termini da un membro all'altro di una equazione e per semplificarli.

Dal titolo di questo trattato è derivato il termine "algebra" (che indicava originariamente quella parte della matematica che riguardava appunto i procedimenti di soluzione delle equazioni, ed ha poi man mano assunto un significato più ampio e generale), mentre dal nome dell'autore, al-Khwarizmi, in latino *Algoritmus*, è venuto il termine "algoritmo", che dal significato iniziale di "regola per il calcolo con le cifre decimali" è gradualmente passato ad indicare qualunque procedura effettiva per la soluzione di un problema matematico attraverso un insieme di regole che consentono di trasformare, in un numero finito di passi, i dati iniziali nel risultato voluto (o non voluto, se l'algoritmo è scorretto...).

Negli anni Trenta, la nozione di "effettivamente calcolabile", e quindi di algoritmo, è stata uno dei principali oggetti di studio dei logici matematici e Turing ne ha data una definizione formale attraverso un modello ideale di macchina che è tuttora uno strumento concettuale indispensabile per lo studio teorico della capacità dei calcolatori reali di risolvere problemi. Oltre alla definizione di Turing, altre definizioni formali di "computazione effettiva" sono state date da altri logici nello stesso periodo. Il fatto che le diverse formulazioni si siano dimostrate equivalenti (nel senso che definiscono tutte lo stesso insieme di funzioni effettivamente computabili) è il dato essenziale che giustifica la tesi di Church, secondo cui la nozione rigorosa di macchina di Turing costituisce una formalizzazione adeguata del concetto informale di algoritmo o procedimento effettivo.

mente". Anche il calcolatore meccanico potrà allora assumere un certo numero (finito) di stati distinti, e ogni sua mossa dipenderà dallo stato in cui si trova e dal simbolo letto.

"Le operazioni effettuate dal calcolatore possono essere decomposte in operazioni semplici così elementari che non è facile pensare di poterle ulteriormente dividere".

Turing identifica le seguenti operazioni elementari: leggere un simbolo, cambiarlo, passare ad osservare il quadrato successivo o il quadrato precedente, cambiare lo stato della mente.

La macchina di Turing è così sostanzialmente definita; riassumendo e precisando meglio alcuni dettagli, possiamo vederla come un meccanismo composto essenzialmente da tre organi:

- un nastro potenzialmente illimitato diviso in caselle, in ognuna delle quali si può scrivere un simbolo, scelto in un alfabeto finito Σ . Il nastro funge da memoria della macchina. Per semplicità, possiamo assumere che l'alfabeto contenga solo i simboli 1 e #. Inoltre, useremo il simbolo \emptyset per indicare le caselle vuote.
- una testina di lettura e scrittura, che può spostarsi sul nastro, leggere il simbolo contenuto in una cella, sostituirlo con un altro simbolo. Questo dispositivo corrisponde essenzialmente all'unità aritmetico-logica di un computer reale.
- una unità di controllo, che può assumere un numero finito di stati distinti.

La figura in questa pagina visualizza una ipotetica realizzazione fisica di una macchina di Turing. È però importante osservare che la descrizione delle macchine di Turing in termini meccanici non è essenziale, poiché esse sono oggetti matematici, e possono quindi essere descritte in termini del tutto astratti. La metafora meccanica è tuttavia estremamente utile soprattutto per capire il funzionamento delle macchine di Turing, di cui parleremo in seguito.

Lezione 23

Tipi di dati: strutture di strutture

Abbiamo visto esempi di strutture di dati costruite a partire da variabili semplici aggregate secondo le regole di un costruttore di tipo. È possibile però costruire strutture più complesse, nelle quali i componenti siano a loro volta strutture complesse. Abbiamo già trattato una struttura di questo tipo: infatti il programma per costruire un listino prezzi per un bar utilizzava due array, uno contenente i nomi delle consumazioni, l'altro i corrispondenti prezzi. Poiché, in realtà, le informazioni contenute nei due array sono correlate, la descrizione più congruente al problema sarebbe stata la seguente:

```
VAR LISTINO: ARRAY [1..30] OF
  RECORD
    CONSUMAZIONE: [(GIN,COGNAC,WHISKY,VODKA,APERITIVI)];
    PREZZO: 1000..5000
  END
```

Abbiamo così costruito una struttura array, i cui elementi sono record. I campi di ciascun record sono variabili di tipo semplice.

Possiamo anche immaginare di costruire una struttura dati che indichi, per tutti gli allievi di una classe, i voti che hanno ottenuto in ciascuna materia nella pagella ricevuta a una certa data. In PASCAL la descriveremmo così:

```
VAR PAGELLE: RECORD
  CLASSE: STRINGA;
  DATA: RECORD
    GG: 1..31;
    MM: 1..12;
    AA: 80..90
  END;
  ALUNNI: ARRAY [1..20] OF
    RECORD
      NOME: STRINGA;
```

Si noti la definizione di tipi utente per l'indice e i componenti dell'array VOTO.

PAGELLE

		CLASSE	2D	IPPOLITO NIEVO		10 - 2 - 84				
NOME ALUNNI	1	ROSSI	×	5	▲	6	●	7	■	6
	2	BIANCHI		8		7		8		8
		●						●		
		●						●		
		●						●		
		●						●		
		●						●		
		●						●		
		●						●		
		●						●		
		●						●		
		●						●		
		●						●		
		●						●		
	20	VERDI		4		2		5		5

DATA (GG,MM,AA)
 VOTO
 × VOTO (ITALIANO)
 ▲ VOTO (LATINO)
 ● VOTO (MATEMATICA)
 ■ VOTO (INGLESE)

```
VOTO: ARRAY [(ITALIANO, LATINO, MATEMATICA, INGLESE)] OF  
0..10;  
END
```

END
che corrisponde al modulo della pagina precedente che prevede:

- l'indicazione della classe;
- la data a cui le pagelle si riferiscono;
- l'elenco di tutti gli allievi, per ciascuno dei quali si indicano:
 - il nome;
 - i voti riportati, rispettivamente, in italiano, latino, matematica e inglese.

Si noti che la struttura costruita è un record i cui componenti sono sia di tipo semplice (stringa) sia di tipo complesso. Infatti il campo DATA è a sua volta un record costituito di elementi semplici, mentre il campo ALUNNI è un array. Tale array ancora è costituito di record comprendenti un campo di tipo semplice (NOME) e un campo che è a sua volta un array (VOTO).

Realizzazione in BASIC di strutture di strutture

Abbiamo già visto la realizzazione BASIC del listino dei prezzi. Poiché il BASIC non consente di definire strutture complesse e in particolare non dispone del costruttore record, siamo stati costretti a utilizzare due array, uno per le consumazioni e uno per i prezzi. La correlazione tra ogni consumazione e il suo prezzo è data, nel nostro esempio, dalla posizione nell'array, per cui lo stesso indice individua la consumazione e il relativo prezzo.

Più complesso è il caso del modulo dei voti, che è stato ottenuto con applicazioni successive dei due costruttori ARRAY e RECORD. In BASIC non abbiamo altra scelta che utilizzare variabili indipendenti dei tipi previsti dal linguaggio. In questo caso sarebbe necessaria:

- una variabile stringa per la classe;
- tre variabili intere per i tre campi del record data e cioè rispettivamente per il giorno, il mese e l'anno;
- un array di stringhe per i nomi degli allievi;
- un array di quattro interi per ciascuno degli allievi; si noti che ciascuno di tali array dovrebbe avere un proprio nome per un totale così di venti differenti array.

Modalità di riferimento a componenti di strutture complesse

Definite le modalità di riferimento ai componenti delle strutture array e record, nel caso di strutture complesse non occorre fare altro che applicare in successione tali regole di riferimento, fino a individuare il componente in questione. Così, poiché la struttura PAGELLE è un record, il riferimento a ciascuno dei campi avverrà nel modo seguente:

```
PAGELLE. CLASSE  
PAGELLE. DATA  
PAGELLE. ALUNNI
```

Se poi vogliamo individuare il campo MM del record PAGELLE. DATA non dobbiamo fare altro che applicare nuovamente la regola di riferimento propria del costruttore record:

La possibilità di costruire strutture di strutture è, come si vede, uno strumento molto potente, che consente in particolare di descrivere in modo conciso un insieme anche numeroso e variegato di variabili; si noti, inoltre, quanto più espressiva risulta la descrizione PASCAL anche grazie all'uso di tipi definiti dall'utente, rispetto alla sola disponibilità dei tipi predefiniti.

PAGELLE. DATA. MM

Poiché, inoltre, il campo alunni è in realtà un array i cui componenti sono record, se vogliamo fare riferimento all'i-esimo alunno e alla sua votazione diremo:

PAGELLE. ALUNNI [I]. NOME

PAGELLE. ALUNNI [I]. VOTO

e, se ci interessa specificamente il voto d'italiano di tale i-esimo alunno, diremo:

PAGELLE. ALUNNI [I]. VOTO [ITALIANO]

Come si vede, non si tratta d'altro che di successive applicazioni delle regole di riferimento dei due costruttori.

Strutture di strutture: un esempio

Immaginiamo di descrivere il catalogo di una biblioteca. Tale catalogo conterrà evidentemente un insieme di informazioni che identificano ciascuno dei libri disponibili. Si tratta di informazioni quali il nome degli autori, il titolo, l'editore, l'anno di pubblicazione e un insieme di dati che consentono di catalogarlo sulla base del genere letterario e di individuarne la collocazione fisica nella biblioteca. Possiamo inoltre immaginare che esista un elenco di tutti coloro che ne hanno una copia in prestito. Tutte queste informazioni possono essere sinteticamente descritte così:

```

TYPE LIBRO = RECORD
  AUTORI: ARRAY [1..5] OF STRINGA;
  EDITORE: STRINGA;
  TITOLO: STRINGA;
  ANNO PUBBLICAZIONE: 1520..2000;
  TIPO: (GIALLO, ROMANZO, SAGGIO);
  COLLOCAZIONE: RECORD
    REPARTO: 1..10;
    SCAFFALE: A..Z;
    POSIZIONE: 1..100
  END;
  PRESTITI: ARRAY [1..NCOPIE] OF
    RECORD
      NOME: STRINGA;
      INDIRIZZO: STRINGA;
      SCADENZA: DATA
    END
END
END

```

Si noti, in particolare, che nella definizione sono stati usati tipi utente che si suppongono già definiti (per esempio il tipo DATA).

A questo punto, non è difficile costruire un programma che costruisce il suddetto catalogo e effettua quindi ricerche di un libro particolare sulla base per esempio dell'autore. Lo schema del programma risulterebbe così:

```

VAR CATALOGO: ARRAY [1..NLIBRI] OF LIBRO;
FOR I: = 1 TO NLIBRI DO
  BEGIN
    ACQUISISCI CATALOGO [I]. TITOLO, CATALOGO [I]. EDITORE, CATALOGO [I], ANNO PUBBLICAZIONE, CATALOGO [I]. TIPO;
    FOR J: = 1 TO 5 DO
      ACQUISISCI CATALOGO [I]. AUTORI [J];
    DETERMINA CATALOGO. COLLOCAZIONE
  END
END

```

L'equivalente programma BASIC risulterebbe così:

```
10 INPUT "numero libri";N
20 DIM A$(N,5),E$(N),T$(N),A(N),G$(N),R(N),S$(N)
,P(N)
30 INPUT "Massimo numero copie disponibili";M
40 DIM P$(N,M),I1(N,M),S1$(N,M)
50 FOR I=1 TO N
60 INPUT "Editore,titolo";E$(I),T$(I)
63 FOR J=1 TO 5
65 INPUT "Autore";A$(I,J)
67 NEXT J
70 INPUT "Anno pubblicazione,genere";A(I),G$(I)
80 GOSUB 200 ' Stabilisce collocazione
90 NEXT I
100 GOTO 250
200 ' Stabilisce collocazione fisica
210 RETURN
250 ' Fine programma
```

Si noti che anche se il linguaggio di programmazione disponibile non prevede l'utilizzo di tipi utente sia semplici che complessi, è comunque importante utilizzare il PASCAL come linguaggio per modellare le strutture di dati e gli algoritmi. La principale difficoltà infatti nella costruzione di programmi consiste non tanto nella conoscenza delle specifiche caratteristiche tecniche di un linguaggio, bensì nella comprensione del problema e nell'individuazione di un opportuno modello di soluzione. I linguaggi come il PASCAL sono, da questo punto di vista, all'avanguardia e rappresentano lo stato dell'arte. È pertanto buona norma, anche per programmatori esperti, non cedere mai alla tentazione di costruire direttamente il programma nel linguaggio di programmazione disponibile, ma passare sempre attraverso una fase di disegno dell'algoritmo e delle strutture di dati. Fatto ciò, la costruzione del programma altro non è che una traduzione con le istruzioni disponibili dell'algoritmo già individuato. Si tratta qui di "simulare" le strutture sia di dati che di controllo usate nella descrizione PASCAL. Se, inoltre, tale "traduzione" sarà opportunamente spiegata da commenti, si otterrà un programma che ha maggiori garanzie di aderenza al problema e quindi maggiori probabilità di correttezza e inoltre, non ultimo vantaggio, sarà più leggibile.

Cosa abbiamo imparato

In questa lezione abbiamo visto:

- L'uso dei costruttori di tipo array e record per realizzare strutture complesse in PASCAL;
- La realizzazione BASIC di strutture complesse;
- le modalità di riferimento ai componenti di strutture complesse.

LE FORME MUSICALI

Le forme musicali rappresentano le strutture più generali nella descrizione di un brano musicale.

Motivo, frase, periodo

Per dare l'idea della musica, può essere utile paragonarla a un libro. Un libro è organizzato in capitoli, costituiti da paragrafi, a loro volta costituiti da periodi, a loro volta costituiti da frasi, a loro volta costituite da parole, a loro volta costituite da lettere.

La musica ha una strutturazione proprio di questo tipo: la *nota* corrisponde alla *lettera*, il *motivo* corrisponde alla *parola*, la *frase* musicale alla *frase*, il *periodo* musicale al *periodo*, il *movimento* al *paragrafo* o al *capitolo* e l'intera *composizione* musicale al *libro*.

Della nota abbiamo già parlato; vediamo invece, partendo dalle altre strutture musicali elementari, come si organizza il materiale musicale.

Il *motivo* è l'unità minima che contraddistingue un brano: è costituito da almeno due note e possiede un'identità ritmica che lo rende riconoscibile. Come esempio, consideriamo il motivo che apre la *Quinta sinfonia* di Beethoven:



La *frase* è costituita da uno o più motivi e viene generalmente conclusa con una *cadenza* (di cui abbiamo parlato in precedenza).

Il *periodo* è formato da una o più frasi; generalmente, ha una prima parte che porta ad una condizione di attesa, seguita da una seconda parte che porta alla soddisfazione dell'attesa creata; ad esempio, nel tema del quarto movimento della *Nona sinfonia* di Beethoven (che abbiamo già avuto modo di discutere precedentemente), le prime quattro battute creano un'attesa, poi soddisfatta dalle successive quattro battute. Si tratta di una struttura del tipo *Frase1...: Frase2...*

Il periodo ha in genere un'estensione di otto battute come nell'esempio considerato, ma esistono naturalmente svariati contro-esempi di periodi più brevi (contratti) o più lunghi (dilatati).

Forme binarie e forme ternarie

L'organizzazione dei periodi in strutture più complesse avviene mediante l'applicazione di funzioni musicali (quali quelle di cui abbiamo discusso nei capitoli precedenti) ai periodi iniziali della partitura.

Ma la struttura ad alto livello di queste trasformazioni del materiale musicale iniziale risponde generalmente a modelli formali molto semplici che vengono chiamati *forma binaria* e *forma ternaria*.

La forma binaria è costituita da due parti, una in contrapposizione all'altra:

A B

dove *A* è una struttura che parte dalla tonica e approda alla dominante, *B* è una struttura che parte dalla dominante e ritorna alla tonica.

La forma ternaria ha invece una struttura di tipo:

A1 B A2

in cui *A1* e *A2* sono molto simili tra loro (se non addirittura uguali) mentre *B* è una struttura molto differenziata rispetto alle altre due, contenente tonalità diverse da quella di *A1* (ed eventualmente anche lontane).

Sulla base di queste semplici nozioni, possiamo definire dei *modelli* di brani musicali e costruire programmi che generano ed eseguono testi musicali con le caratteristiche strutturali da noi designate; è molto semplice, infatti, tradurre in termini di programmazione i concetti sovraesposti; ad esempio una forma ternaria in cui *A1 = A2* è traducibile in una struttura:

```
100 gosub 1000      REM struttura A1
110 gosub 2000      REM struttura B
120 gosub 1000      REM struttura A2 = A1
```

L'uso opportuno delle strutture di controllo e delle strutture di dati ci consente di codificare con semplicità e chiarezza l'informazione musicale non solo come flusso sequenziale di note, ma anche come struttura intima della partitura.

Soluzione dell'Esercizio 3

Eccoci a definire un modello del frammento della partitura della *Sagra della Primavera* (ai numeri 91-92); le ipotesi formulate nel numero precedente erano:

- tenere fissa la struttura ritmica;
- giustapporre la sequenza *A B C D E* alla ripetizione per quattordici volte della struttura *F*;
- descrivere il materiale melodico del frammento lasciando la possibilità di alterarlo.

Vedremo ora una realizzazione che soddisfa queste tre ipotesi e che, inoltre, contiene la descrizione dei rapporti melodici tra la struttura *A* e le altre di modo che, se vogliamo alterare la melodia del brano, è sufficiente alterare *A*; le altre strutture saranno quindi calcolate automaticamente mantenendo i legami armonici dello schema strutturale originale di Stravinsky.

Usiamo due matrici a due dimensioni (*d* per le durate e *a* per le altezze) per memorizzare il testo; il secondo indice corrisponde alla struttura: 0 per la *A*, 1 per la *B* ecc.; il primo indice definisce il numero d'ordine della nota a cui si riferisce il parametro: 0 per la prima nota di ogni struttura, 1 per la seconda ecc.

Nella parte iniziale del programma memorizziamo la struttura ritmica (070-290) e la struttura melodica del tema originale (300-380) e

le relazioni delle altre strutture melodiche rispetto a quella del tema della struttura *A* (mediante la subroutine 1380-1570).

Segue la parte interattiva, in cui:

- scegliamo il tempo (dal lento 0 al velocissimo 5; istruzioni 570-585);
- scegliamo se alterare il tema melodico o no (istruzioni 600-610);
- eseguiamo le strutture *A*, *B*, *C*, *D*, *E* (istruzioni 620-670) e la ripetizione per quattordici volte di *F* (istruzioni 680-720);
- decidiamo se riprendere dal passo a) o terminare l'esecuzione del programma (istruzioni 725-750).

Quando scegliamo di modificare il tema melodico originale (al punto b)) chiamiamo la subroutine 1000-1100 in cui abbiamo la possibilità di specificare una nuova sequenza di sette altezze che vanno a sostituire l'originale struttura melodica *A*; i valori delle altezze devono essere codificati secondo la tabella dell'istruzione **SOUND**.

Oltre a visualizzare la nuova sequenza di altezze (1050-1080), eseguiamo nuovamente la subroutine 1380-1570 per calcolare tutte le altre strutture melodiche in funzione del nuovo frammento melodico *A* che abbiamo definito.

Proviamo allora a sperimentare nuovi temi per questa struttura di Stravinsky; sarà meglio l'originale o una nuova stesura?



Sopra, una pagina della partitura della "Sagra della primavera", nella stesura originale di Igor Stravinsky.

A lato, una caricatura del musicista eseguita dal pittore futurista Fortunato Depèro.



```

010 clear
020 REM esercizio 3
030 REM N. 91-92 della partitura
040 REM della "Sagra della
    primavera"
050 REM di Igor Stravinsky
060 dim d(7,5),a(7,5)
070 REM struttura di base: ritmo
080 d(0,0)=3
090 d(1,0)=1
100 d(2,0)=2
110 d(3,0)=1
120 d(4,0)=1
130 d(5,0)=2
140 d(6,0)=2
150 d(7,0)=0
160 for i=0 to 7
170 d(i,1)=d(i,0)
180 d(i,2)=d(i,0)
190 d(i,3)=d(i,0)
200 next i
210 d(7,1)=2:d(7,3)=2
220 d(0,2)=0:d(1,2)=0
230 d(0,3)=0:d(1,3)=0
240 d(0,4)=1:d(1,4)=1:d(2,4)
    =4:d(3,4)=6
250 for i=0 to 3
260 d(i+4,4)=0
270 d(i,5)=1
280 d(i+4,5)=0
290 next i
300 REM sequenza di base: melodia
    originale
310 a(0,0)=1660
320 a(1,0)=1864
330 a(2,0)=1660
340 a(3,0)=2216
350 a(4,0)=2484
360 a(5,0)=1864
370 a(6,0)=2216
380 a(7,0)=0
390 gosub 1380
560 ri$=""
570 REM impostazione del tempo
580 input "tempo (da 0 a 5)";tmp
585 t=32/2^tmp;if t>255 then t=255
590 REM introduzione di nuovo
    frammento melodico
600 input "vuoi cambiare il tema
    (SI/NO)";ri$
610 if ri$="SI" then gosub 1000
620 REM esecuzione
630 for j=0 to 4
640 for i=0 to 7
645 if a(i,j)>16383 then
        a(i,j)=16383
650 sound a(i,j),d(i,j)*t
660 next i
670 next j
680 for k=0 to 13
690 for i=0 to 3
695 if a(i,5)>16383 then
        a(i,5)=16383
700 sound a(i,5),d(i,5)*t
710 next i
720 next k
725 print ""
730 input "vuoi continuare (SI/NO)
    ";ri$
740 if ri$="SI" goto 560
750 end

1000 REM modifiche
1005 ri$=""
1010 for i=0 to 6
1020 print "altezza";i;":"
1030 input a(i,0)
1040 next i
1050 print "nuovo frammento
    melodico"
1060 for i=0 to 6
1070 print a(i,0);
1080 next i
1085 REM modifica degli altri
    frammenti
1090 gosub 1380
1100 return

1380 REM relazioni melodiche tra i
    frammenti
1390 for i=0 to 7
1400 a(i,1)=a(i,0)
1410 a(i,2)=a(i,0)
1420 a(i,3)=a(i,0)
1430 next i
1440 a(6,1)=a(5,0)/1.059463^2
1450 a(7,1)=a(6,0)
1460 a(5,3)=a(6,1)
1470 a(6,3)=a(5,1)
1480 a(7,3)=a(7,1)
1490 a(0,4)=a(4,0)
1500 a(1,4)=a(3,0)
1510 a(2,4)=a(3,0)/1.059463^2
1520 a(3,4)=a(2,4)
1530 a(0,5)=a(1,0)*2
1540 a(1,5)=a(2,0)*2^2
1550 a(2,5)=a(3,0)*2
1560 a(3,5)=a(4,0)*2^2
1570 return

```



ADRIANO ABBADO

Moderne congas e tamburo rullante: questi strumenti a percussione derivano dall'evoluzione del primitivo tamburo a fessura dell'isola di Pentecoste (Oceania).

La fuga

Una particolare forma musicale è la *fuga*.

La sua struttura è generalmente la seguente:

P P P R contrappunto libero

S C contrappunto libero

P R C contrappunto libero

P P S C contrappunto libero

In questo modello, la prima riga rappresenta la parte di *soprano*, la seconda la parte di *contralto*, la terza la parte di *tenore*, la quarta la parte di *basso*; si tratta, quindi, di una fuga a quattro voci (il caso più comune); *S* è il *soggetto*, cioè la frase che caratterizza la fuga; *R* è la *risposta*, cioè "il richiamo" del soggetto per trasposizione di una quinta verso il grave o verso l'acuto; *C* è il *controsoggetto*, cioè una linea melodica che si oppone a quella del soggetto e che viene a sovrapporsi ora al soggetto ed altrimenti alla risposta; *P* è una pausa della durata del soggetto.

Dopo questa fase, che viene chiamata *esposizione*, viene una fase *centrale* ricca di modulazioni, trasposizioni, inversioni speculari ecc.; in questa fase il numero delle voci "attive" può diminuire anche ad una sola voce.

Infine, la *conclusione* della fuga riprende la struttura principale dell'*esposizione*.

Il rondò-sonata

Per concludere questa rapida carrellata sulla nozione di forma musicale, vediamo un interessante esempio di struttura complessa, che combina le caratteristiche della *sonata* con quelle del *rondò*.

Il rondò ha la seguente struttura:

A1 B A2 C A3

in cui *A1*, *A2* e *A3* sono strutture basate sul tema nella tonalità della composizione, *B* e *C* sono invece strutture "diversive" (come *B* nella forma ternaria) in tonalità diverse dalla tonica.

La sonata è una forma ternaria in cui *A1* è l'*esposizione*, *B* è lo *sviluppo* e *A2* è la *ripresa*.

Il rondò-sonata ha quindi la seguente struttura:

• *Esposizione*:

A1: primo tema, in tonica;

B1: secondo tema, in dominante o altra tonalità (anche molto lontana dalla tonica);

A2: primo tema, in tonica;

• *Sviluppo*:

C: corrisponde allo sviluppo della normale sonata ed è quindi costituito da trasformazioni anche molto elaborate del primo tema, eventualmente in diverse tonalità;

• *Ripresa*:

A3: primo tema, in tonica;

B2: secondo tema, questa volta in tonica;

A4: primo tema, in tonica, con funzioni di coda.

Un utile quanto divertente esercizio può essere senz'altro quello di cimentarsi nella composizione di un semplice rondò-sonata, descrivendo esplicitamente solo i due temi e realizzando tutto il resto della composizione mediante strutture di controllo e funzioni musicali programmate.

BUSINESS GRAPHICS: I DIAGRAMMI A "TORTA"

Un programma per M10 che realizza diagrammi a torta, un immediato e diretto strumento informativo che permette di presentare dati in forma grafica.

Abbiamo visto in precedenza come la business graphics computerizzata, detta anche Computer-Aided Business Graphics, sia diventata un efficace e ormai indispensabile strumento di controllo manageriale, grazie alla sua enorme capacità di comunicare informazioni in modo visivo.

Presentare i dati in maniera grafica consente infatti di ottenere alcuni grossi vantaggi, quali:

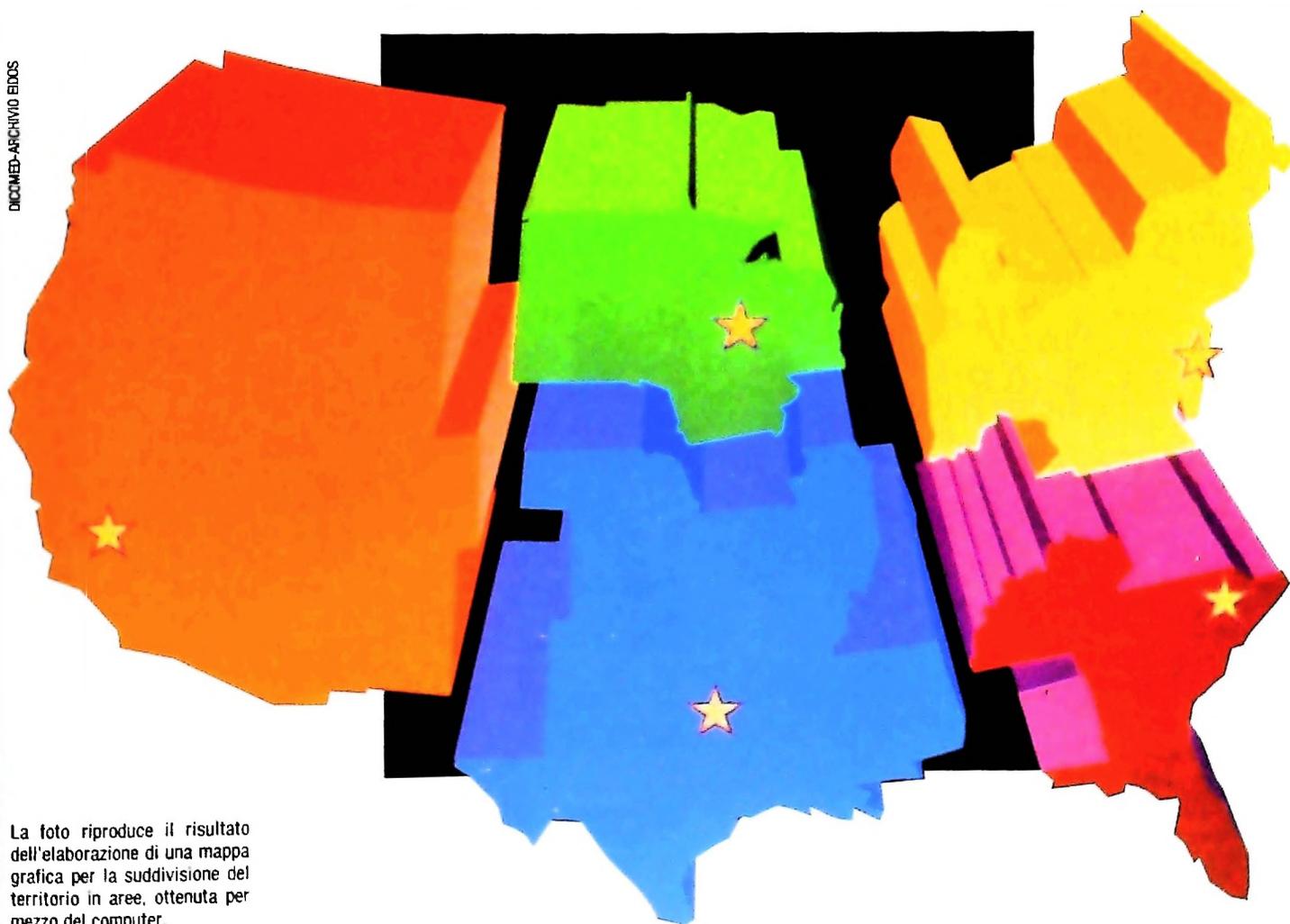
— *determinazione degli errori*, ossia è immediatamente rile-

vabile quando alcuni valori sono troppo alti o troppo bassi;
— *valutazione degli scostamenti*, da quelli che sono i risultati previsti o attesi;

— *riduzione della quantità di dati*, essendo gli stessi sintetizzati in un grafico;

— *comparazione dei valori*: il significato di un grafico ottenuto da un insieme di dati può essere stabilito confrontandolo con il grafico di un secondo insieme. I diagrammi a barre e a

DICOMED-ARCHIVO EIDOS



La foto riproduce il risultato dell'elaborazione di una mappa grafica per la suddivisione del territorio in aree, ottenuta per mezzo del computer.

Come realizzare i diagrammi a torta

```

10 CLS
15 PIGRECA=3.1415
30 INPUT "QUANTE FETTE";N
40 DIM A(N),X(N),Y(N),T(N),P(N),C(N)
50 FOR I=1 TO N
60 PRINT "Q.TA' FETTA";I;
70 INPUT " : ";T(I)
100 T=T+T(I)
110 NEXT I
120 FOR I=1 TO N
130 P(I)=T(I)*100/T
140 B=B+360*P(I)/100
150 A(I)=B/180
160 NEXT I
170 INPUT "PLOTTER O VIDEO (P/V)"; A$
180 IF A$="P" THEN 600 ELSE IF A$="V"
THEN 205
190 PRINT "ERRORE DI INPUT"
200 GOTO 170
201 REM
202 REM
205 REM ROUTINE PER USCITA SU VIDEO
206 REM
207 REM
210 REM ASSEGNAZIONE CENTRO E RAGGIO
220 REM DELLA TORTA.
230 XC=40 : YC=31 : R=27
250 GOSUB 2000
280 CLS
285 REM STAMPA DEI RISULTATI NUMERICI
290 PRINT TAB(14);"          Q.TA'          % "
300 FOR I=1 TO N
310 C=FIX(P(I)) : D=P(I)-C : D=FIX(D*100) :
C(I)=C+D/100
320 PRINT TAB(14);"T"I;"          ";;PRINT USING
"EEEE";T(I);:PRINT "          ";;PRINT USING
"EEE.EE";C(I)
330 NEXT I
340 GOSUB 2500
370 PRINT TAB(14);"TOT.;" "          ";;PRINT USING
"EEEE";T;:PRINT "          ";;PRINT USING
"EEE.EE";CT
380 REM DISEGNO DELLA TORTA
400 FOR ANG=0 TO 2*PIGRECA STEP PIGRECA/50
410 X1=XC+R*COS(ANG)
420 Y1=YC+R*SIN(ANG)
430 PRESET (X1,Y1,0)
440 NEXT ANG
450 FOR I=1 TO N
460 LINE (40,31)-(X(I),Y(I)),1
470 NEXT I
500 GOTO 4000
600 REM ROUTINE PER USCITA SU PLOTTER
605 REM
610 REM ASSEGNAZIONE CENTRO E RAGGIO
611 REM DELLA TORTA
620 XC=130 : YC=0 : R=85
630 GOSUB 2000
665 REM DISEGNO DELLA TORTA
670 LPRINT CHR$(18)
675 LPRINT "C2"
680 LPRINT "M215,0"
690 X0=215 : Y0=0
700 FOR ANG=0 TO 2*PIGRECA STEP PIGRECA/50
710 X1=XC+R*COS(ANG)
720 Y1=YC+R*SIN(ANG)
730 LPRINT "D"+STR$(X0)+", "+STR$(Y0)+", "
+STR$(X1)+", "+STR$(Y1)
740 X0=X1 : Y0=Y1
742 NEXT ANG
745 LPRINT "C3"
750 DIM Z(N+1)
752 Z(1)=0
754 FOR I=2 TO N
756 Z(I)=A(I-1)*PIGRECA
758 NEXT I
760 FOR I=1 TO N
763 IF I<4 THEN K=I ELSE K=I-3
765 LPRINT "C"+STR$(K)
768 FOR J=Z(I) TO A(I)*PIGRECA STEP A(I)/10
770 LPRINT "M"+STR$(XC)+", "+STR$(YC)
775 X=XC+R*COS(J) : Y=YC+R*SIN(J)
780 LPRINT "D"+STR$(XC)+", "+STR$(YC)+", "
+STR$(X)+", "+STR$(Y)
790 NEXT J
810 NEXT I
815 REM STAMPA DEI RISULTATI NUMERICI
820 LPRINT "MO,85"
830 LPRINT "CO"
910 LPRINT CHR$(17)
920 LPRINT TAB(25)"Q.TA'          % "
930 FOR I=1 TO N
940 C=FIX(P(I)) : D=P(I)-C : D=FIX(D*100)
: C(I)=C+D/100
950 LPRINT TAB(19)"T"I"          ";;LPRINT USING
"EEEE";T(I);:LPRINT "          ";;LPRINT USING
"EEE.EE";C(I)
960 NEXT I
970 GOSUB 2500
980 LPRINT TAB(19)"TOT.;" "          ";;LPRINT USING
"EEEE";T;:LPRINT "          ";;LPRINT USING
"EEE.EE";CT
990 GOTO 4000
2000 REM CALCOLO COORDINATE VERTICI DEI
2010 REM BORDI DELLE FETTE.
2020 FOR I=1 TO N
2030 X(I)=XC+R*COS(A(I)*PIGRECA)
2040 Y(I)=YC+R*SIN(A(I)*PIGRECA)
2050 NEXT I
2060 RETURN
2500 REM SUBROUTINE PER LA SOMMA
2501 REM DELLE PERCENTUALI
2510 FOR I=1 TO N
2520 CT=CT+C(I)
2530 NEXT I
2540 RETURN
4000 END

```

Questo programma realizza diagrammi a torta su M10 e fornisce i relativi dati numerici. I risultati sono visualizzabili sul video o sul plotter.

10-110 Si forniscono in input interattivamente i dati, dei quali si calcola anche il totale (istr. 100), e si dimensionano dinamicamente, ossia in base al numero dei dati, gli array di utilizzo nel programma.

120-160 Si calcolano le percentuali sul totale dei singoli dati, $P(I)$, e l'angolo che la fetta della torta rappresentante i singoli dati formerà con l'asse delle x , $A(I)$.

170-200 Istruzioni di selezione per la periferica di uscita e controlli sulla correttezza della richiesta.

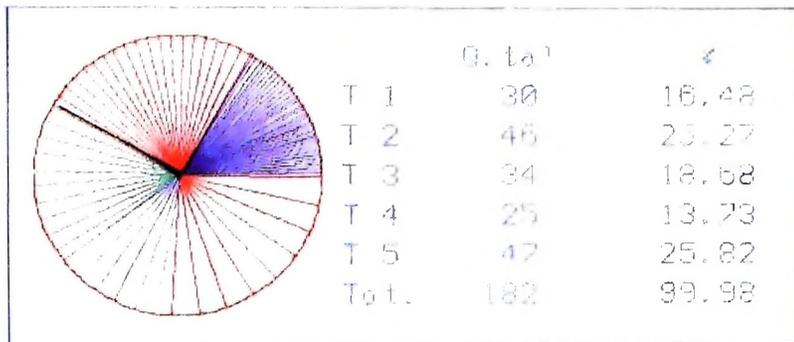
Routine per uscita su video

Nella subroutine 2000 si calcolano le coordinate dei vertici dei lati delle fette utilizzando gli angoli $A(I)$ calcolati prima.

285-370 Queste istruzioni servono per stampare sulla destra dello schermo i dati e i risultati dell'elaborazione. La linea 310, costituita da quattro istruzioni, serve per troncare i valori percentuali $P(I)$ alla seconda cifra decimale. Nella linea 320, costituita anch'essa da quattro istruzioni, compaiono due comandi PRINT USING, che servono per definire un campo e un formato all'interno del quale verranno stampati dei valori numerici. Ossia, per esempio, significa: "stampa usando il formato £££.££" in cui le cifre intere andranno allineate alla sinistra del punto e le decimali a destra. Nel caso "££££" si possono inserire soltanto cifre intere. La subroutine 2500 serve per calcolare i dati totali da stampare nella 370.

380-470 Blocco di istruzioni per disegnare la torta sul video. Dalla 400 alla 440 si sviluppa il già noto ciclo di FOR per il disegno del cerchio con incrementi di angolo pari a $PIGRECA/50$. Il ciclo dalla 450 alla 470 serve per tracciare le fette sulla torta, utilizzando i vertici calcolati nella subroutine 2000. Esse vengono disegnate procedendo in senso orario.

Va segnalato che, date le dimensioni del video, non possono essere visualizzati i risultati relativi a più di 4 dati. Inoltre, per via delle approssimazioni alla seconda cifra decimale sulle percentuali parziali, può risultare un totale inferiore di qualche centesimo al 100%. Sarebbe comunque possibile, con qualche artificio, arrotondare qualcuno dei dati in modo da far tornare il totale; qui non lo si fa per maggior chiarezza. Va anche rilevato che le approssimazioni introdotte dal calcolo delle funzioni trigonometriche (seno e coseno) e la risoluzione grafica propria dello schermo dell'M10 generano delle linee segmentate.



Routine per uscita su plotter

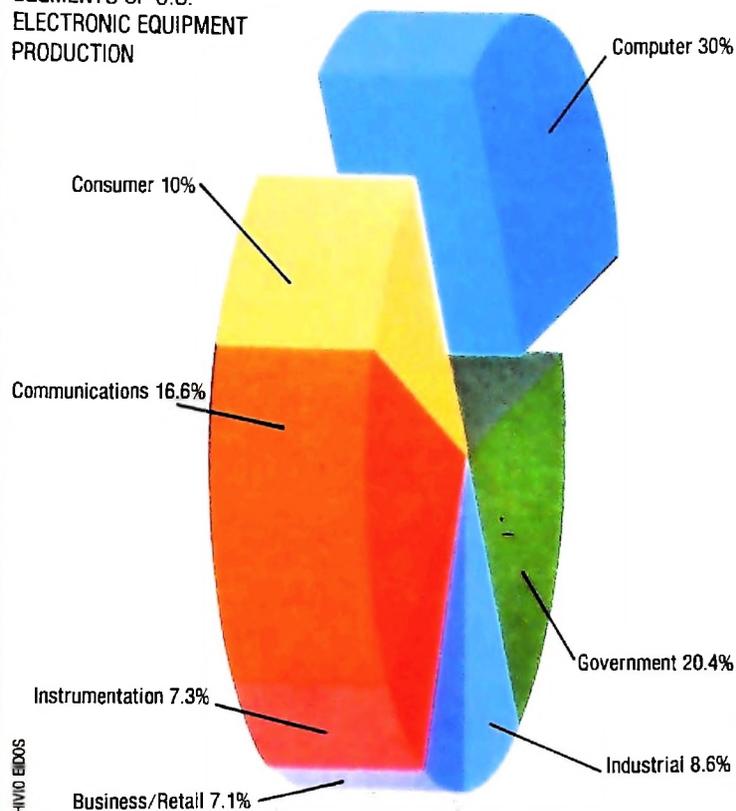
Anche qui si assegnano le coordinate del centro e del raggio del cerchio e quindi si richiama la subroutine 2000.

665-742 Qui si disegna il cerchio con le modalità che abbiamo già visto in altre occasioni.

750-810 Si definisce un nuovo array (istr. 750) al quale vengono assegnati i valori degli angoli $A(I)$ moltiplicati per $PIGRECA$ e in modo tale che $Z(I)=A(I-1)$. Ciò permette (ciclo do FOR 768-790) di disegnare le singole fette della torta e di differenziarle con una tratteggiatura colorata. Sul plotter la disposizione avviene in senso antiorario, ossia in maniera contraria a quella del video.

815-990 Queste istruzioni servono per scrivere alla destra della torta i risultati numerici. Esse sono costituite in modo del tutto analogo a quanto specificato nel caso dell'uscita su video, con le dovute modifiche relative ai comandi per il plotter.

SEGMENTS OF U.S. ELECTRONIC EQUIPMENT PRODUCTION



DICO MED-ARCHIVIO EIDOS

Diagramma a torta tridimensionale, sul quale è stata effettuata una rotazione spaziale.

torte sono tecniche grafiche atte a queste comparazioni; — *correlazioni*, per determinare il grado di similarità o di differenza fra due o più insiemi di dati. Per esempio, il grado di istruzione e lo stipendio, oppure l'età e la salute.

Applicazioni della Business Graphics

Il "marketing" e la "finanza" sono due settori nei quali la business graphics ha avuto e continuerà ad avere un rapido sviluppo; ma anche in tutte le applicazioni in cui si fa ricorso alla statistica o alla matematica applicata essa rappresenta un valido strumento, come in medicina, in biologia e in tutti i settori scientifici.

Nel "marketing", ossia nello studio del mercato, la business graphics è usata per quantificare la capacità di penetrazione di un'azienda o di un prodotto, per studiare e prevedere le possibili espansioni del mercato e anche, utilizzando delle mappe disegnate sul video, per selezionare le zone all'interno di un'area in cui va concentrato lo sforzo di vendita.

Nel mondo finanziario, la business graphics costituisce un immediato e diretto strumento informativo. I dati trovano infatti un efficace sommario in un grafico, dal quale è già possibile dedurre alcune informazioni generali sul buon o cattivo esito di una manovra finanziaria. Inoltre la possibili-

tà di confrontare grafici omogenei, ossia riferiti ad un unico insieme di valori, ma ottenuti in periodi di tempo diversi, permette di individuare eventuali cicli periodici dei valori studiati. Informazione, questa, di fondamentale importanza nel mondo finanziario, dove per esempio è essenziale sapere se e perché esiste una ciclicità nel fenomeno della svalutazione e della rivalutazione monetaria.

Come presentare i dati

Come in tutte le attività informative, anche in business graphics esiste il problema di definire alcune regole base nella presentazione dei dati. Due qualità che comunque dovrebbero sempre caratterizzare un grafico o un diagramma sono: consistenza e semplicità.

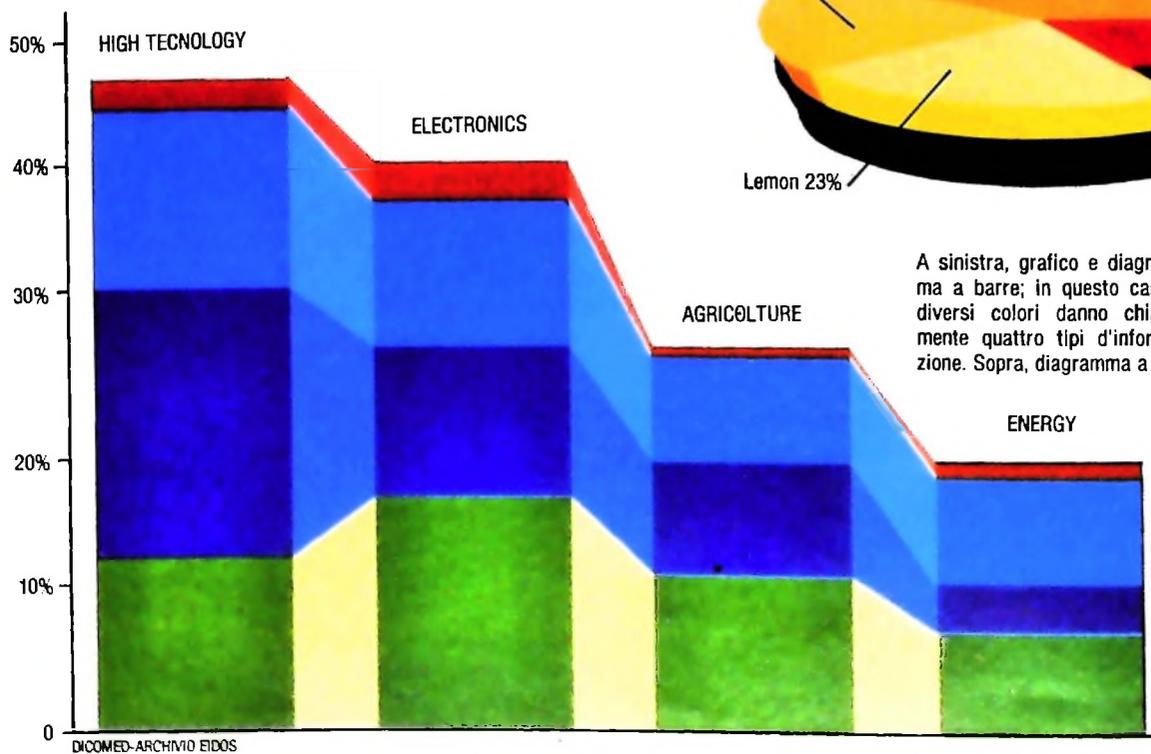
I simboli grafici comunicano informazioni e costituiscono il vocabolario del linguaggio grafico. Come in ogni linguaggio, deve esserci consistenza, ossia coerenza, nel significato degli elementi base e nelle loro combinazioni. La consistenza nel disegno, nei formati, nei simboli e nei colori è essenziale in business graphics; altrimenti, potrebbe essere difficile se non impossibile comparare due o più grafici fra loro, ossia le differenze presenti in una tale comparazione devono essere attribuibili a differenze nel contesto dell'informazione e non a variazioni della struttura grafica o dei simboli del vocabolario. Proprio per queste ragioni, si sta studiando, anche per la

business graphics, uno standard che consenta di evitare errori di interpretazione su grafici realizzati da persone diverse. La business graphics deve essere anche semplice nel disegno, per consentire al lettore di recepire appieno l'informazione. La percezione dell'informazione di un grafico è infatti massimizzata quando esso mette a fuoco una singola idea e quando la confusione visiva è ridotta al minimo. Usare un solo grafico per riassumere più informazioni contemporaneamente può essere utile in alcune circostanze, ma è per lo più fonte di confusione visiva e di conseguenza anche di cattiva informazione.

Una delle componenti della business graphics è il colore. Il suo utilizzo, però, solleva alcuni problemi, legati al fatto che ogni persona reagisce emozionalmente in maniera diversa ai vari colori. Molti utenti però preferiscono i grafici colorati per le loro qualità estetiche, ma quando si utilizza un grafico per evidenziare dati qualitativi, si è verificato che l'uso di colori diversi per lo stesso grafico induce a diverse interpretazioni, a seconda della reazione individuale ai colori prescelti: bisogna quindi affermare che in certe situazioni i colori vanno utilizzati con molta accortezza e forse sarebbe meglio non farne uso. È per questa ragione che nel settore finanziario si tende sempre più ad utilizzare grafici in bianco e nero.

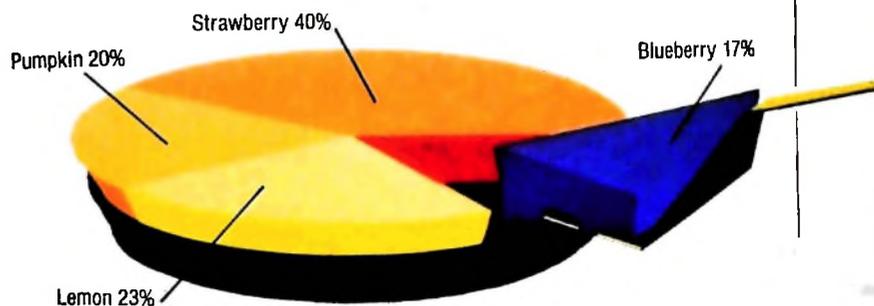
Il colore ha, nonostante quanto detto, un vantaggio: catturare l'attenzione di chi guarda e porre l'enfasi su un certo insieme di dati. Usato con oculatazza, quindi, il colore diventa anch'esso un serio strumento informativo.

PROJECTED TWO YEAR INDUSTRIAL GROWTH BY COUNTRY



DICOMED-ARCHIVIO EIDOS

AUGUST PIE SALES



A sinistra, grafico e diagramma a barre; in questo caso i diversi colori danno chiaramente quattro tipi d'informazione. Sopra, diagramma a tor-

ta tridimensionale; il computer ha lavorato qui anche esteticamente, con l'inserimento di una sorgente luminosa che crea luci e ombre.

UNITED STATES
WEST GERMANY
GREAT BRITAIN
FRANCE

— UN NUOVO MODO DI USARE LA BANCA. —

Conto corrente più

TANTI PENSIERI
IN MENO CON IL CONTO
CORRENTE "PIÙ"
DEL BANCO DI ROMA.

Essere cliente del Banco di Roma vuol dire anche essere titolari del conto corrente "più". Un conto corrente più rapido: perché già nella maggior parte delle nostre filiali trovate gli operatori di sportello che vi evitano le doppie file.

Più comodo, perché potete delegare a noi tutti i vostri pagamenti ricorrenti: dai mutui all'affitto, dalle utenze alle imposte.

Più pratico, perché consente l'utilizzo del sistema di prelievo automatico Bancomat e l'ottenimento della carta di credito.

Più esclusivo, perché potete usufruire del servizio Voxintesi, attraverso il quale chiedere direttamente al nostro elaboratore il saldo del vostro conto corrente con una semplice telefonata: in qualsiasi ora come in qualsiasi giorno, anche festivo.

Più sicuro, perché con una minima spesa potrete assicurarvi contro furti e scippi mentre vi recate in banca o ne uscite.

Veniteci a trovare, ci conosceremo meglio.

 **BANCO DI ROMA**
CONSCIAMOCI MEGLIO.



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 comunicare via telefono per spedire e ricevere informazioni. In grado di funzionare a batteria oppure collegato all'impianto elettrico, M10 mette ovunque a disposizione la sua potenza di memoria, il suo display orientabile a cristalli liquidi capace anche di elaborazioni grafiche, la sua tastiera professionale arricchita da 16 tasti funzione.



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

PERSONAL COMPUTER OLIVETTI M10

L'UFFICIO DA VIAGGIO



Anche in leasing con Olivetti Leasing.

olivetti

OLIVETTI S.p.A. - Via Broletto, 15 - 20121 Milano
 Olivetti Personal Computer Division - Via Mesago, 12
 20070 Mesago (MI) - Italia
 TELEFONO