

43 CORSO PRATICO COL COMPUTER

421958

è una iniziativa
FABBRI EDITORI

in collaborazione con
BANCO DI ROMA

e **OLIVETTI**

F4 F5 F6 F7 F8

diretto da **GIANNI DEGLI ANTONI**

BATTERY LOW

**IN EDICOLA
DAL
28 GENNAIO**

**I GRANDI TEMI
DELLA
MEDICINA**
PER CAPIRE, PREVENIRE, STAR MEGLIO

**IL SISTEMA
NERVOSO**

**IL DIZIONARIO
DELLA
MEDICINA**
n.2
NUOVO ATLAS

**IL DIZIONARIO
DELLA
MEDICINA**
n.1
PER CAPIRE, PREVENIRE, STAR MEGLIO

I primi due fascicoli del Dizionario
e il primo volume de I Grandi Temi
a sole lire **2.200**

**FABBRI
EDITORI**

IL BANCO DI ROMA FINANZIA IL VOSTRO ACQUISTO DI M 10 e M 20

Acquisto per contanti

È la formula di acquisto tradizionale. Non vi sono particolari commenti da fare, se non sottolineare che troverete ampia disponibilità presso i punti di vendita Olivetti, poiché, grazie al "Corso pratico col computer", godrete di un rapporto di privilegio.

Il servizio di finanziamento bancario

Le seguenti norme descrivono dettagliatamente il servizio di finanziamento offerto dal Banco di Roma e dagli Istituti bancari a esso collegati:

Banca Centro Sud
Banco di Perugia

Le agenzie e/o sportelli di questi istituti sono presenti in 216 località italiane.

Come si accede al credito e come si entra in possesso del computer

- 1) Il Banco di Roma produce una modulistica che è stata distribuita a tutti i punti di vendita dei computer M 10 e M 20 caratterizzati dalla vetrofania M 10.
- 2) L'accesso al servizio bancario è limitato solo a coloro che si presenteranno al punto di vendita Olivetti.
- 3) Il punto di vendita Olivetti provvederà a istruire la pratica con la più vicina agenzia del Banco di Roma, a comunicare al cliente entro pochi giorni l'avvenuta concessione del credito e a consegnare il computer.

I valori del credito

Le convenzioni messe a punto con il Banco di Roma, valide anche per le banche collegate, prevedono:

- 1) Il credito non ha un limite minimo, purché tra le parti acquistate vi sia l'unità computer base.
- 2) Il valore massimo unitario per il credito è fissato nei seguenti termini:
 - valore massimo unitario per M 10 = L. 3.000.000
 - valore massimo unitario per M 20 = L. 15.000.000
- 3) Il tasso passivo applicato al cliente è pari

- al "prime rate ABI (Associazione Bancaria Italiana) + 1,5 punti percentuali".
- 4) La convenzione prevede anche l'adeguamento del tasso passivo applicato al cliente a ogni variazione del "prime rate ABI"; tale adeguamento avverrà fin dal mese successivo a quello a cui è avvenuta la variazione.
 - 5) La capitalizzazione degli interessi è annuale con rate di rimborso costanti, mensili, posticipate; il periodo del prestito è fissato in 18 mesi.
 - 6) Al cliente è richiesto, a titolo di impegno, un deposito cauzionale pari al 10% del valore del prodotto acquistato, IVA inclusa; di tale 10% L. 50.000 saranno trattenute dal Banco di Roma a titolo di rimborso spese per l'istruttoria, il rimanente valore sarà vincolato come deposito fruttifero a un tasso annuo pari all'11%, per tutta la durata del prestito e verrà utilizzato quale rimborso delle ultime rate.
 - 7) Nel caso in cui il cliente acquisti in un momento successivo altre parti del computer (esempio, stampante) con la formula del finanziamento bancario, tale nuovo prestito attiverà un nuovo contratto con gli stessi termini temporali e finanziari del precedente.

Le diverse forme di pagamento del finanziamento bancario

Il pagamento potrà avvenire:

- presso l'agenzia del Banco di Roma, o Istituti bancari a esso collegati, più vicina al punto di vendita Olivetti;
- presso qualsiasi altra agenzia del Banco di Roma, o Istituto a esso collegati;
- presso qualsiasi sportello di qualsiasi Istituto bancario, tramite ordine di bonifico (che potrà essere fatto una volta e avrà valore per tutte le rate);
- presso qualsiasi Ufficio Postale, tramite vaglia o conto corrente postale. Il numero di conto corrente postale sul quale effettuare il versamento verrà fornito dall'agenzia del Banco di Roma, o da Istituti a esso collegati.

 **BANCO DI ROMA**
CONOSCIAMOCI MEGLIO.

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
Ricerca universitaria presso l'Istituto di Cibernetica dell'Università degli Studi di Milano

Curatori di rubriche
MARCO ANELLI, DIEGO BIASI, ANDREA GRANELLI, ALDO GRASSO, MARCO MAIocchi, DANIELE MARINI, GIANCARLO MAURI, CLAUDIO PARNELLI

Testi
DOMENICO CAVALLOTTI, DANIELE MARINI, Eidos (TIZIANO BRUNETTI)
Etnoteam (ADRIANA BICEGO)

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

Direttore Editoriale
ORSOLA FENGLI

Redazione
CARLA VERGANI
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 1985 Gruppo Editoriale Fabbri, Bompiani, Sorzogno, Etas S.p.A., Milano - Copyright © sull'opera 1984 Gruppo Editoriale Fabbri, Bompiani, Sorzogno, Etas S.p.A., Milano - Prima Edizione 1984 - Direttore responsabile GIOVANNI GIOVANNINI - Registrazione presso il Tribunale di Milano n. 136 del 10 marzo 1984. - Iscrizione al Registro Nazionale della Stampa n. 00262, vol. 3, Foglio 489 del 20.9.1982 - Stampato presso lo Stabilimento Grafico del Gruppo Editoriale Fabbri S.p.A., Milano - Diffusione - Distribuzione per l'Italia Gruppo Editoriale Fabbri S.p.A., via Mecenate, 91 - Milano - tel. 02/50951 - Pubblicazione periodica settimanale - Anno II - n. 43 - 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.

CORSO PRATICO COL COMPUTER

Volume Quarto

CORSO PRATICO

Diretto da GIANNI DEGLI ANTONI

Q

W

R

Y

A

S

F

G

H

C

B



COL COMPUTER

FABBRI EDITORI

U 4

I 5

O 6

P

=
,
|

ENTE

J 1

Finger

L 3

Finger

%
ù

Finger

?
,
0

/
:[

Finger

NUM

Direttore dell'opera
GIANNI DEGLI ANTONI

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

UMBERTO ECO
Ordinario di Semiotica presso l'Università di Bologna

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

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

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

Curatori di rubriche
MARCO ANELLI
DIEGO BIASI
ANDREA GRANELLI
ALDO GRASSO
MARCO MAIOCCHI
DANIELE MARINI
GIANCARLO MAURI
CLAUDIO PARMELLI

Direttore Editoriale
ORSOLA FENGHI

Redazione
CARLA VERGANI
LOGICAL STUDIO COMMUNICATION

Art Director
CESARE BARONI

Impaginazione
BRUNO DE CHECCHI
PAOLA ROZZA

Programmazione Editoriale
ROSANNA ZERBARINI
GIOVANNA BREGGÉ

Segretarie di Redazione
RENATA FRIGOLI
LUCIA MONTANARI

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

EVOLUZIONE DEI DATA-BASE

Una crescita determinata dall'esigenza di gestire in maniera integrata vasti sistemi di dati.

Gli elaboratori sono oggi parte essenziale della nostra vita. Questa realtà è immediatamente constatabile recandosi in banca, in comune o allo sportello ferroviario: in pochi minuti, o secondi, possiamo avere un estratto conto, un certificato di residenza o una prenotazione per un treno.

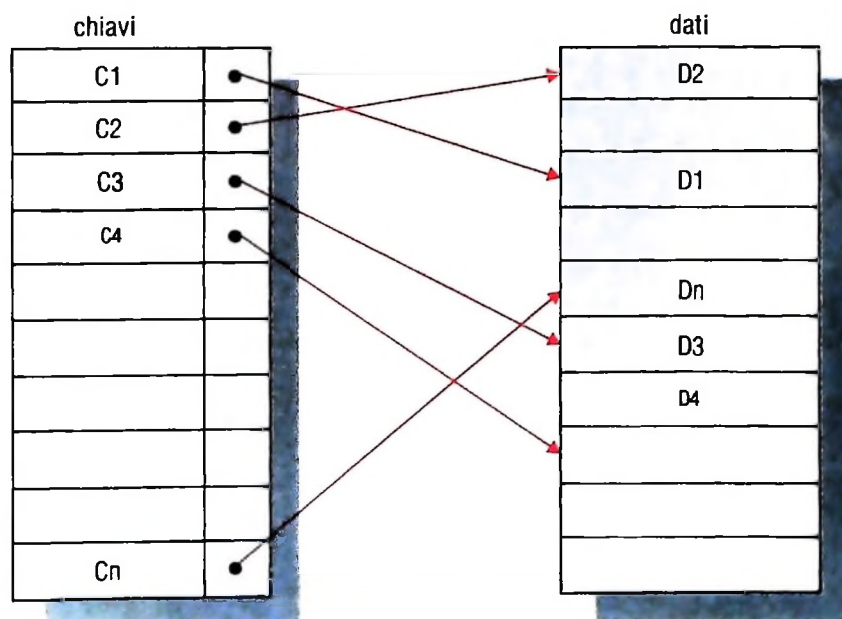
Il problema principale che viene risolto con queste applicazioni è quello di archiviare, reperire ed elaborare in modo efficiente masse di dati enormi; anzi si può dire che quando la quantità di informazione su cui si opera è molto elevata, il problema della elaborazione passa in secondo piano, mentre quello della organizzazione dei dati diventa decisamente di maggior rilievo.

Per elaborare dei dati, bisogna prima trovarli: se questi fanno parte di un archivio di dimensioni limitate, diciamo 100 elementi, è accettabile una loro scansione sequenziale fino a trovare quello che ci serve, ma se si ha a che fare con archivi

di grandi dimensioni (si parla di milioni di elementi) adottare una strategia come la precedente sarebbe proprio come cercare un ago in un pagliaio. Per risolvere i problemi di reperimento dell'informazione sono state studiate fino dagli albori della computer science delle tecniche per memorizzare l'informazione in modo tale da poterla recuperare efficientemente.

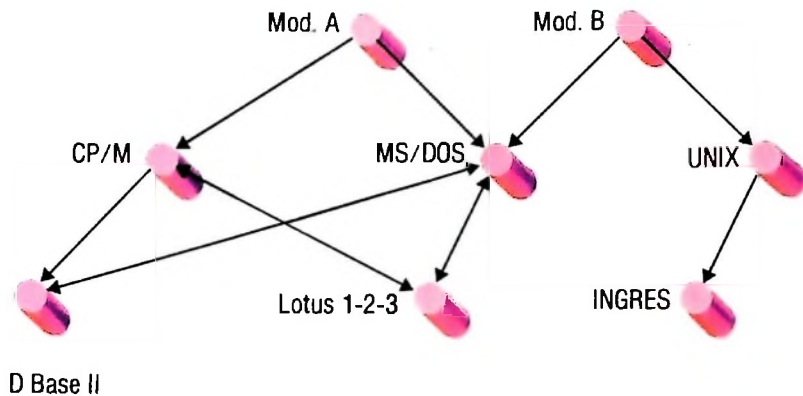
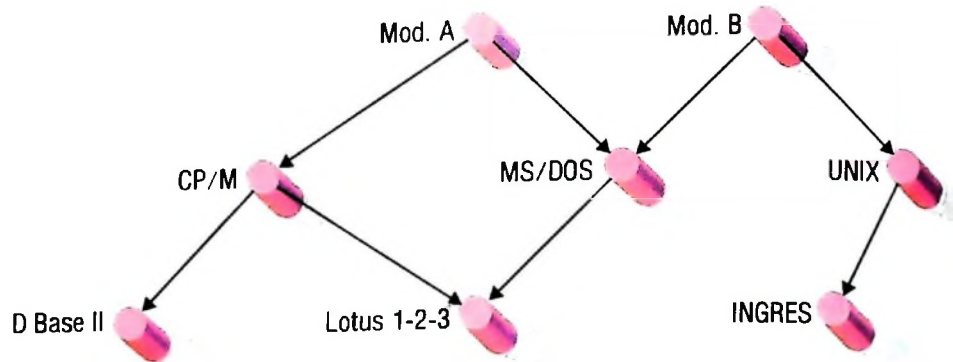
In generale le entità che dobbiamo archiviare e manipolare sono informazioni complesse che risultano dalla aggregazione di informazioni semplici. Per esempio in una rubrica telefonica l'entità trattata è costituita dalla aggregazione di un nome, un indirizzo e un numero di telefono; queste tre informazioni semplici vengono viste come un'entità unica, detta "record", strutturata in campi.

Per l'uso che si fa della rubrica, una di queste informazioni (il nome) riveste un ruolo particolare, in quanto siamo inte-



Nell'organizzazione a indice l'informazione viene suddivisa in una sezione dati e in una sezione chiavi, a cui viene assegnata la posizione in memoria dei dati relativi.

Un esempio di data-base gerarchico (a destra) che contiene informazioni su due modelli di computer (modello A e modello B), sui relativi sistemi operativi (CP/M, MS/DOS e UNIX) e sui prodotti software (Dbase II, Lotus 1-2-3, Ingres). In un data-base gerarchico le relazioni tra i dati sono predefinite. Il cammino di ricerca parte dagli "strati" superiori (modello B per esempio) e arriva a quelli inferiori (Dbase II e Ingres).



In un data-base reticolare (a sinistra) esiste più flessibilità che in uno gerarchico, in quanto gli strati inferiori possono collegarsi a quelli superiori. Questo significa che è possibile accedere al data-base anche dai livelli inferiori. Sia i data-base gerarchici che quelli reticolari presentano il problema che, quando si deve inserire una nuova relazione (per esempio un nuovo prodotto software) o modificarne una già esistente, si deve ridefinire l'intero archivio.

ressati, conoscendo il nome, a ritrovare l'indirizzo oppure il numero di telefono e non viceversa. Il nome diventa quindi una "chiave" per poter accedere alle informazioni a esso associate.

Vediamo ora quale organizzazione dei dati è possibile utilizzare. Una prima idea ci viene suggerita dalle normali rubriche telefoniche: immagazzinare l'informazione suddividendola in base alla prima lettera del campo chiave. Il recupero di questa avverrà effettuando una ricerca sequenziale nella "pagina" corrispondente alla iniziale. Questo tipo di organizzazione è però poco flessibile e comporta un notevole spreco di spazio: infatti se si stabilisce a priori lo spazio dedicato a ogni lettera, buona parte di questo spazio rimane inutilizzato e se malauguratamente uno conosce troppe persone il cui nome inizia per esempio con la lettera "s" dovrà cambiare rubrica. Un altro metodo è quello normalmente usato negli schedari, ovvero immagazzinare l'informazione in modo rigidamente ordinato rispetto al campo chiave, per poi avere un accesso rapido in fase di ricerca. Questo tipo di soluzione richiede il riordinamento di tutte le informazioni ogni volta che viene modificato un campo chiave o inserita una nuova informazione.

Per evitare lo spostamento di grandi masse di dati si è quindi pensato di separare i campi chiave dall'informazione a essi associata e di associare a ogni chiave l'indicazione di dove

trovare il resto dell'informazione; questo è in qualche modo simile all'indice di un libro dove in poche pagine sono concentrate le informazioni che permettono di rintracciare un argomento in base al numero di pagina. È possibile raffigurare questo tipo di organizzazione come nella figura alla pagina precedente. Ogni record ha una chiave di accesso e una parte dati, rispettivamente C e D, l'indice è costituito dall'insieme delle chiavi a ognuna delle quali è associato un puntatore alla parte dati.

Queste non sono le sole organizzazioni dei dati possibili; sono state infatti studiate tecniche estremamente sofisticate e specializzate per la gestione dell'informazione, ognuna delle quali ha i propri pregi e difetti a seconda dell'uso che si intende fare della base di dati: se si intende privilegiare l'occupazione di memoria, la velocità di inserimento di nuovi dati o il tempo necessario per la loro ricerca.

Guardando al passato, si può dire che fino alla metà degli anni Sessanta l'informazione veniva organizzata prevalentemente in maniera seriale e il software di sistema demandava al programmatore il compito di amministrare i dati sul supporto di memorizzazione.

Chiaramente se occorre apportare una modifica alla struttura dei dati era necessario riscrivere il programma applicativo che li utilizzava e ricompilarlo.

Normalmente ogni programma applicativo utilizzava una

**Tabella
dei computer**

Numero di serie	Modello	Sistema operativo
100	A	CP/M
200	B	MS/DOS
350	C	MS/DOS

**Tabella dei
sistemi operativi**

Numero di serie	Nome	Programma
27	CP/M	DBase II
42	MS/DOS	Lotus 1-2-3
111	MS/DOS	DBase II

**Tabella
dei programmi**

Numero di serie	Programma	Prezzo
87	DBase II	1200
22	Lotus 1-2-3	1400
43	Ingres	6000

Graficamente un data-base relazionale può essere visto come un insieme di tabelle collegate da una o più relazioni. Questa struttura offre il vantaggio di poter definire nuove relazioni senza dover modificare la struttura dell'archivio.

propria organizzazione dell'informazione necessaria all'esecuzione. Ciò implicava che diversi programmi utilizzando gli stessi dati, o dati in comune, richiedessero copie multiple degli stessi. Questa ridondanza dell'informazione crea non pochi problemi poiché richiede una maggiore occupazione della memoria di massa e, in caso di aggiornamento di dati memorizzati in copie multiple, più operazioni di quanto non sarebbe logicamente richiesto. Il problema più grave che deriva dalla ridondanza dei dati è però legato alla "consistenza" dell'informazione. Infatti è evidente la possibilità che in un preciso istante il valore di un dato sia stato aggiornato in una copia e non ancora in un'altra; fatto gravissimo in un'organizzazione, in quanto può portare a decisioni non coerenti.

Proprio per superare i problemi di inconsistenza dei dati sono stati realizzati i data-base: ovvero un sistema che si interpone tra i programmi applicativi e un'unica base di dati. Il software che permette a una o più persone di usare e/o modificare i dati viene detto DBMS (Data Base Management System). Il compito principale del DBMS è quello di permettere all'utente o al programma di trattare i dati in termini di astrazioni, senza preoccuparsi di come sono immagazzinati nella memoria fisica.

Una volta stabilita l'esigenza di questo strumento di base, verso la fine degli anni Sessanta i fornitori di sistemi hanno offerto sul mercato una serie di realizzazioni tra di loro sostanzialmente differenti. Pur rifacendosi all'esigenza comune

di uno strumento in grado di consentire un'integrazione dei dati garantendo anche la separazione tra aspetti logici e fisici, i maggiori fornitori hanno adottato architetture logiche differenti.

In particolare, la IBM ha introdotto l'IMS (la sigla significa Information Management System), un sistema di gestione di basi di dati imperniato su una organizzazione di tipo gerarchico. Questo tipo di modello organizza i dati secondo una struttura ad albero in cui i nodi rappresentano insiemi di entità, e i figli di ogni nodo sono a questo associati da qualche relazione. Riprendendo l'esempio della rubrica telefonica, a un nodo verrà associata una lettera dell'alfabeto, i nodi figli di questo nodo saranno tutti i nomi con quella iniziale e ogni nome avrà come figli gli indirizzi e i numeri di telefono a esso associati.

Parallelamente all'adozione da parte della IBM del modello gerarchico sono apparsi sul mercato dei sistemi alternativi, caratterizzati da una maggior flessibilità e basati su un modello reticolare o anulare. Questo modello prevede l'organizzazione dei dati secondo una filosofia di "catene" o "anelli" di record, seguendo i quali l'utilizzatore può "navigare" all'interno del data-base seguendo una certa "rotta" logica.

I fautori di questi due modelli si sono dati battaglia negli anni Sessanta e Settanta per imporre il proprio come standard. Da questo confronto si può dire che non è emerso alcun vincitore; piuttosto si è creata nell'area dei data-base una confusione che ha influenzato negativamente tutti i contendenti sul mercato.

Nel 1970 E.F. Codd, un ricercatore IBM, pubblicava su una rivista scientifica un articolo intitolato *A Relational Model of Data for Large Shared Data Banks*, destinato a divenire una delle pietre miliari nella storia dell'informatica. In questo articolo Codd presentava un nuovo modello di data-base, definito "relazionale", la cui semplicità, flessibilità e potenza logica rappresentavano un netto salto di qualità rispetto ai modelli gerarchici e reticolari.

Sia sul modello gerarchico che su quello reticolare nell'ultimo decennio è gravato lo "spettro" dei data-base relazionali, che sono comparsi sul mercato da un paio di anni e che solo ora stanno concretamente cominciando a far sentire la propria presenza.

Il modello relazionale si basa sul concetto di "relazione tra insiemi", cioè un sottoinsieme del prodotto cartesiano di una lista di domini. Un dominio è semplicemente un insieme di valori. Il prodotto cartesiano dei domini $D_1, D_2 \dots D_n$, scritto $D_1 \times D_2 \times \dots \times D_n$, è l'insieme di tutte le ennuple (v_1, v_2, \dots, v_n) tali che v_1 appartiene a D_1 , v_2 appartiene a D_2 e così via. Per esempio dati gli insiemi $D_1 = \{0,1\}$ e $D_2 = \{a,b,c\}$, il loro prodotto cartesiano $D_1 \times D_2$ sarà $\{(0,a),(0,b),(0,c),(1,a),(1,b),(1,c)\}$ e una relazione su questi due insiemi potrebbe essere $\{(0,a),(1,b),(1,c)\}$. Un modo molto semplice per rappresentare una relazione è la forma tabellare: ogni riga è una ennupla e ogni colonna corrisponde a una componente.

La forma più semplice di relazione tra dati è quella chiamata "relazione binaria" di cui abbiamo riportato un esempio nella relazione "età":

età	
Mario	24 anni
Gianni	36 anni

In ogni colonna compaiono elementi appartenenti allo stesso dominio e su ogni riga sono poste le ennuple che costituiscono la relazione.

Spesso ad ogni colonna della relazione viene associato un nome, corrispondente al dominio. Chiaramente esiste anche un vasto insieme di "relazioni multiple", come ad esempio la relazione "impiegato":

impiegato	nome	età	coniuge	attività
	Gianni	24 anni	Maria	contabile
	Mario	36 anni	Paola	capogruppo

In termini operativi, come avremo modo di vedere in seguito, l'utilizzatore di un data-base relazionale si trova generalmente ad agire su queste tabelle. Operando su queste semplici tabelle è possibile attivare funzionalità di notevole complessità logica, senza doversi preoccupare delle implementazioni fisiche sottostanti.

La semplicità e naturalezza delle forme tabellari, con le quali l'utente è abituato ad agire quotidianamente, rappresenta il principale vantaggio di un data-base relazionale in termini di interfaccia uomo-macchina.

Dietro alla semplicità operativa di un data-base relazionale si nasconde però un rigoroso modello matematico, caratterizzato da alcune regole ben precise.

Questo modello teorico è proprio l'elemento che permette la corretta applicazione di strumenti logici evoluti, come per esempio il calcolo dei predicati.

In questo senso, dietro a un vero data-base relazionale, sta una rigorosa metodologia di definizione delle relazioni, tendente a garantire la futura flessibilità e semplicità operativa. Anche se l'utente finale di un data-base relazionale non avrà a che fare con questo formalismo, esso rappresenta il nucleo fondamentale del modello relazionale.

In un prossimo articolo vedremo come sia possibile ricondurre un insieme di relazioni in quella che viene chiamata "terza forma normale" in modo da poter applicare correttamente gli operatori logici normalmente connessi a un data-base relazionale.

La stampa di stringhe con la grafica del microplotter

Abbiamo finora visto come sia possibile porre il microplotter in modo grafico, e come in questa situazione sia necessario inviare a esso comandi costituiti da una stringa di cui il primo carattere specifica il comando. In particolare abbiamo visto il comando **P** per visualizzare stringhe.

Operando in modo grafico, è possibile definire la dimensione del carattere con un altro comando, in modo tale da ottenere scritte di dimensioni diversissime. Si tratta del comando **S**: l'istruzione

```
10 LPRINT "S0"
```

specifica che si vuole usare un carattere di dimensione tale che su una riga del microplotter ne siano contenuti 80; l'istruzione

```
10 LPRINT "S63"
```

specifica invece una dimensione tale che per ogni riga sia contenuto un solo carattere. In sostanza, il comando **S** ha la forma

$$S_n$$

essendo n un valore compreso tra 0 e 63. La dimensione del carattere, espressa in punti di stampa, è:

$$\begin{aligned} & (n+1) * 4 \text{ la base} \\ & (n+1) * 6 \text{ l'altezza.} \end{aligned}$$

così che se, per esempio, n è uguale a 0 la dimensione del carattere sarà 4 punti di stampa la base e 6 l'altezza.

In assenza di tale comando il microplotter stampa caratteri in modo tale che ne risultino 40 per riga, il che corrisponde a un valore di n pari a 1.

Supponiamo quindi di voler usare tale comando per verificare le possibilità di stampa dei caratteri in formati diversi, e quindi di organizzare un programma nella forma:

- passa in "graphic mode"
- facendo variare n da 0 a 60
- seleziona la dimensione 1
- scrivi una stringa

Purtroppo il comando **S** inviato come stringa in **LPRINT** non è parametrico: esso deve indicare con precisione il valore di n , e non è possibile inserire una variabile per definire tale valore.

Possiamo aggirare l'ostacolo con il seguente accorgimento:

- costruiamo in una variabile stringa il comando Sn;
- usiamo LPRINT per mandare non una stringa direttamente espressa tra virgolette, ma come contenuto della variabile costruita.

In tal modo avremo reso parametrico il nostro comando, potendo modificarlo e costruirlo con ogni possibilità offertaci dalle istruzioni di trattamento delle stringhe in BASIC.

In particolare useremo la funzione STR\$(1), che trasforma il valore numerico della variabile 1 nella corrispondente rappresentazione di caratteri.

Così, supponendo che la variabile 1 contenga il valore 17, l'istruzione

```
50 LET S$="S"+STR$(1)
```

concatena, mediante l'operazione "+" (che su stringhe ha proprio l'effetto di concatenarle l'una all'altra) la lettera "S" alla sequenza di caratteri "17", fornendo come risultato finale la sequenza "S17", che corrisponde al comando di dimensionamento dei caratteri che noi vorremmo usare.

A questo punto possiamo considerare il seguente programma:

```
10 LPRINT CHR$(18) 'Graphic mode
30 FOR I=0 TO 63
40 REM Costruisce il comando
50 LET S$="S"+STR$(I)
60 LPRINT S$
65 LPRINT "PBASIC"
70 NEXT I
```

In esso

- la linea 10 permette di passare al modo grafico;
- la linea 50 costruisce i comandi di dimensione dei caratteri, che al variare del valore dell'indice 1 del ciclo, passa da S0 a S1, fino a S63;
- la linea 60 invia il comando opportunamente costruito;
- la linea 65 usa il comando P per richiedere la stampa della stringa BASIC, che verrà quindi riprodotta 64 volte in 64 formati differenti.

Il risultato che otteniamo dal microplotter è il seguente:

BASICBASICBASICBASICBASICB

Come si vede

- le scritte sono in nero, in quanto tale è il colore su cui il microplotter si posiziona all'accensione;
- la dimensione della parola cresce rapidamente;
- l'istruzione LPRINT comanda l'uso grafico, e ogni comando viene eseguito a partire dalla posizione in cui il pennino si trovava precedentemente; quindi le scritte sono consecutive, e non si va mai a capo;
- dopo aver esaurito la riga sembra che il programma si arresti, in quanto continua a inviare i comandi al plotter, che però non è più in grado di muoversi; dopo un adeguato lasso di tempo, equivalente a quello necessario per tracciare realmente tutte le scritte richieste, il programma termina.

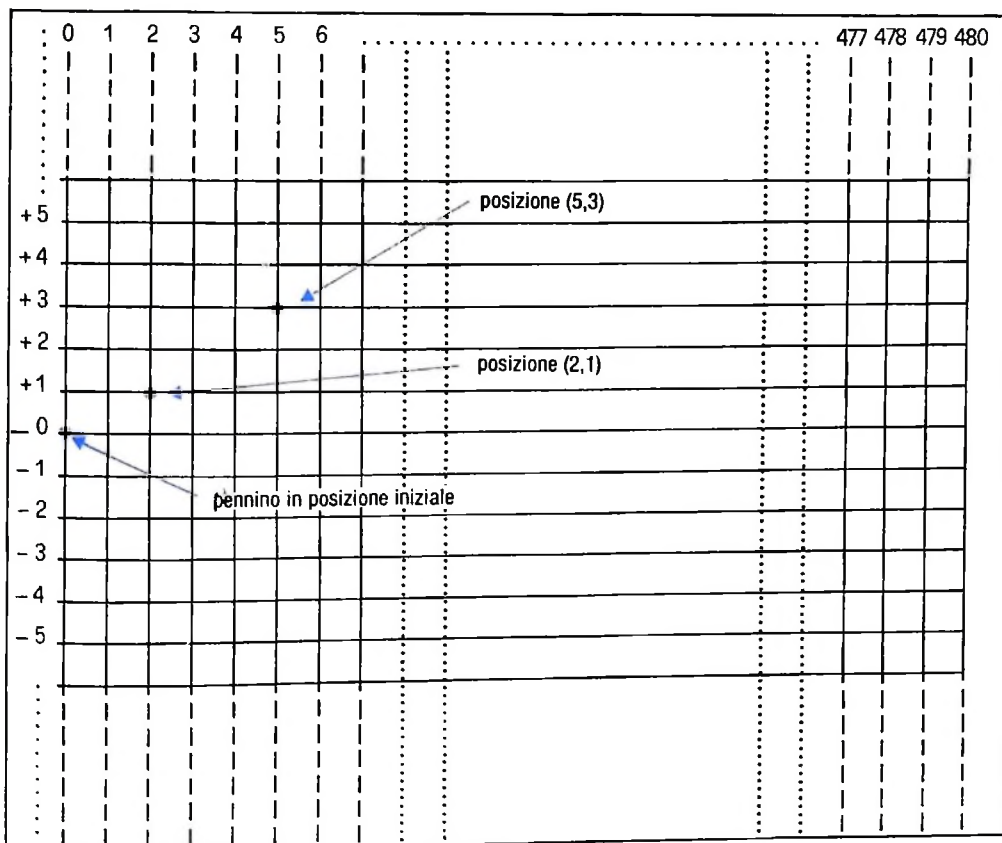
Per fare in modo che il programma "vada a capo" dopo ogni stampa, dovremo dirgli di spostare il pennino alla riga successiva.

Ciò è possibile in diversi modi:

- è possibile infatti richiedere uno spostamento del pennino rispetto al punto in cui si trova, con il comando R;
- oppure è possibile richiedere uno spostamento del pennino in una specifica posizione del foglio, non riferita alla sua posizione attuale, con il comando M.

Prima di analizzare con dettaglio i due comandi è necessario quindi capire cosa intendiamo per "posizione" sul foglio.

La figura illustra una schematizzazione del rullo di carta così come esso è interpretato dal microplotter:



Esso è quindi immaginato come un piano in cui righe e colonne sono numerate: il punto di partenza del pennino all'accensione del microplotter (ovvero all'inizio dell'esecuzione di una parte grafica di un programma, dopo il comando CHR\$(18), è considerato l'origine dei riferimenti, cioè riga 0 e colonna 0; le colonne disponibili arrivano fino alla 480, mentre le righe sono numerate con un valore positivo crescente verso l'alto e negativo verso il basso.

Così, chiedendo al microplotter di spostarsi alla posizione (5,3), vedremo il pennino avanzare di 5 colonne e il rullo arretrare di 3 righe; da qui potremo per esempio, sempre con il comando R, chiedere di spostarci in giù di 2 righe e a sinistra di 3 colonne, raggiungendo la posizione (2,1); lo stesso risultato sarebbe stato ottenuto con il comando M, chiedendo direttamente di posizionarci al punto (2,1).

Le figure seguenti mostrano un esempio di uso dei comandi per spostare il pennino, rispettivamente in modo assoluto e relativo. È stato scelto di stampare.

```
10 LPRINT CHR$(18)
15 LPRINT "M0,0"
20 LPRINT "P1"
30 LPRINT "R50,30"
40 LPRINT "P2"
50 LPRINT "R-40,0"
60 LPRINT "P3"
70 LPRINT "M200,-20"
80 LPRINT "P4"
100 END
```

L'esecuzione:

```
      3 2
     1
           4
```

Vedremo nella prossima lezione l'uso di comandi come quelli illustrati.

Cosa abbiamo imparato

In questa lezione abbiamo visto:

- il comando S per specificare la dimensione dei caratteri quando si opera in modo grafico;
- la funzione STR\$ per ottenere la rappresentazione in caratteri di un valore numerico;
- l'operatore "+" per la concatenazione delle stringhe;
- come il microplotter interpreta il suo foglio di carta.

LISP: UN LINGUAGGIO PER MANIPOLARE LE LISTE

Manipolare simboli anziché valori numerici.

Negli esempi relativi ai vari linguaggi di programmazione visti, come il Pascal, il COBOL, il BASIC, i programmi avevano a che fare per lo più con la trattazione di valori numerici, su cui effettuare calcoli.

Così, in BASIC

```
10 LET A=10
20 LET B=30
30 LET C=A+B
40 PRINT C
```

l'istruzione 30 assegna a C il valore ottenuto dalla somma dei valori precedentemente assegnati alle variabili A e B, mentre l'istruzione successiva stampa tale valore.

Sarebbe però utile avere a disposizione linguaggi di programmazione che permettano di effettuare un calcolo "simbolico", un calcolo cioè, in cui, più che i valori effettivi delle variabili A, B, C contano le loro relazioni.

Che cosa significa "Calcolo simbolico"?

Facciamo un esempio. A scuola, dopo averci insegnato a calcolare espressioni aritmetiche del tipo:

$$(3*2+5)*(2*(5-1)-6/(2+1))$$

il cui valore è $11*(8-2)$, pari a 66, ci insegnano a effettuare calcoli "senza numeri", come:

$a+a$	che vale	$2*a$
a^2-b^2	che vale	$(a+b)*(a-b)$
$(a^2-b^2)/(a-b)$	che vale	$a+b$

e così via.

Astraendo cioè dal fatto che un simbolo, come la lettera "a" rappresenti il valore numerico 5, piuttosto che 10, 50 o 1000, quando incontriamo $5+5$, ovvero $10+10$, o $50+50$, o in generale $a+a$, possiamo trasformare l'espressione in $2*5$ o $2*10$, o in $2*50$ o, in generale, in $2*a$.

Un linguaggio di programmazione che permetta tali operazioni deve avere la capacità di manipolare facilmente i simboli, piuttosto che non i valori numerici. Ciò non è impossibile con un linguaggio di programmazione tradizionale.

Per esempio, in Basic, il seguente programma effettua il calcolo simbolico su un'espressione di due soli operandi, legati da un operatore di somma, sottrazione, divisione o moltiplicazione:

```
10 INPUT "Espressione";A$,O$,B$
   'A$ e B$ operandi;O$ operat.
20 IF A$<>B$ THEN 100
   'Nessuna semplificazione
   possibile
30 IF O$="+" THEN PRINT "2*";A$
40 IF O$="-" THEN PRINT "0"
50 IF O$="*" THEN PRINT A$;"^2"
60 IF O$="/" THEN PRINT "1"
100 END
```

Fornendo al programma l'espressione:

A*A

si ottiene il risultato:

A^2

Di fatto, però, la manipolazione delle stringhe in un linguaggio come il Basic, per calcolare espressioni simboliche, risulta alquanto macchinosa, qualora si vogliano manipolare espressioni che risultino molto complesse.

Abbiamo invece a disposizione linguaggi di programmazione che, con un approccio completamente differente nella rappresentazione dei dati e nelle istruzioni, sono in grado di facilitare in maniera molto evidente il problema.

Uno di questi, il più famoso, è il Lisp.

Il linguaggio LISP

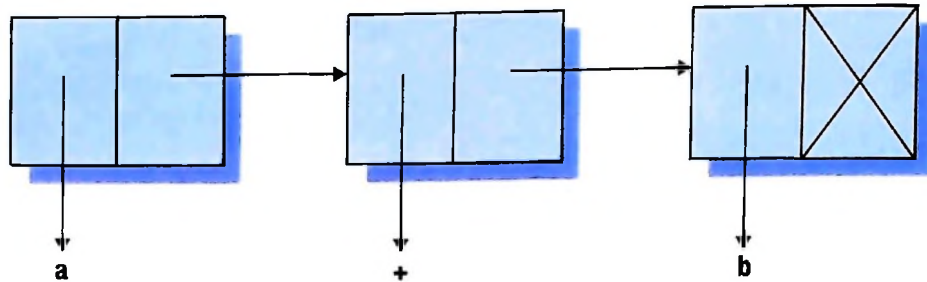
Il Lisp è un linguaggio in grado di elaborare strutture di dati dette LISTE (in effetti il suo nome deriva dalla contrazione List Processor).

Una struttura a lista può essere considerata come una successione di coppie di cellette concatenate tra loro. La prima celletta contiene un riferimento a un simbolo, mentre la seconda indica la seguente coppia da prendere in considerazione.

Così, per esempio, la scrittura:

$(a+b)$

costituisce una struttura del linguaggio Lisp che viene organizzata, nella memoria del calcolatore, nel modo seguente:



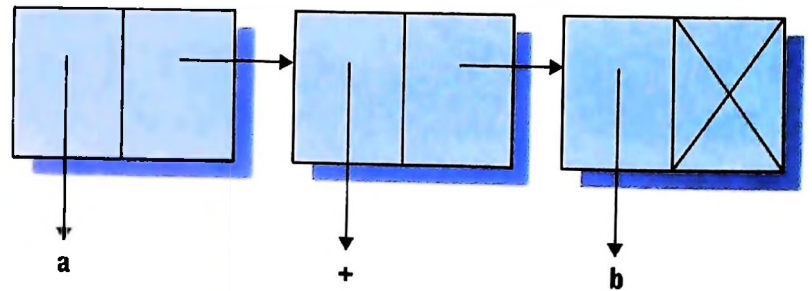
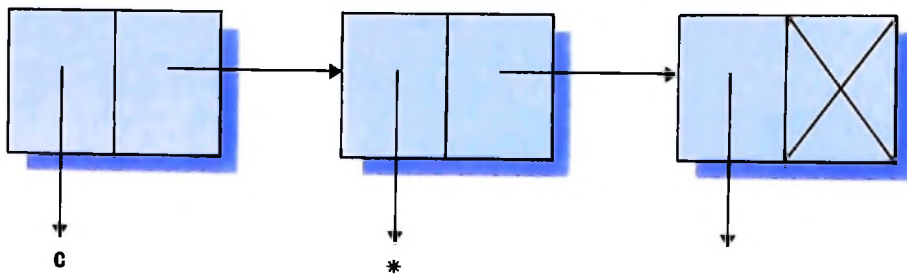
La "croce" che riempie la seconda cella dell'ultima coppia indica che la lista termina: spesso viene indicata anche con il simbolo NIL.

caso seguente:

$(c*(a+b))$

Una lista può avere come suoi elementi anche liste, come nel

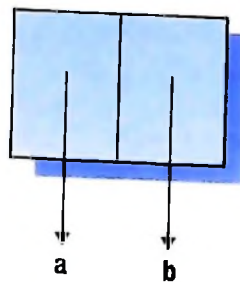
che viene rappresentato in questo modo:



Ancora, il Lisp permette di rappresentare la singola coppia di celle. Vediamo, anche in simile caso, un esempio.

La scrittura: è rappresentata come:

$(a.b)$



La lista (a) equivale quindi a (a.NIL).

Un elemento simbolico base, come i vari "a", "b", "+" degli

esempi, non decomponibile ulteriormente, viene definito "atomo".

Su strutture di dati di tale tipo, il Lisp può effettuare trasformazioni con un certo insieme di operatori, tra cui i più elementari sono riportati in fondo alla pagina; per esempio, l'operatore CAR permette di estrarre il primo elemento di una lista e

(CAR '(a + b))

vale

a

(Si noti l'apice che precede la lista (a + b): esso ha lo scopo di indicare che ciò che segue è una lista di simboli senza alcuna specifica interpretazione associata, e quindi da considerare come una pura costante; vedremo meglio in seguito il perché di una tale convenzione).

Si osservi che il modo con cui abbiamo effettuato la trasformazione ha una forma che è a sua volta una lista: in Lisp un programma (perché, a tutti gli effetti, l'espressione che abbiamo scritto è un programma Lisp) è una lista, né più né meno come un dato.

Un'espressione Lisp da eseguire ha in sostanza la forma di

una lista, il cui primo elemento è il nome dell'operatore da applicare, mentre i restanti elementi sono gli argomenti su cui applicare l'operatore.

È quindi chiara l'esigenza di quell'apice dell'esempio precedente: è assolutamente necessario che la lista (a + b) non venga interpretata, che altrimenti l'esecutore Lisp cercherebbe, prima ancora di applicare l'operazione CAR, di applicare l'operazione "a", primo elemento della lista argomento, che non corrisponde ad alcun operatore noto.

Un programma Lisp è quindi costituito da una funzione, il cui nome è il primo elemento di una lista e i cui argomenti sono gli elementi successivi; la sua esecuzione presuppone, prima, la valutazione di tutti gli argomenti, che a loro volta possono essere espressioni Lisp; poiché un dato e un programma, in Lisp, hanno la stessa forma, nulla vieta che un programma fornisca come risultato della sua esecuzione una lista che possa essere interpretata a sua volta come programma, e quindi eseguita.

Queste potenzialità, che rendono eccezionalmente flessibile (ma anche di una certa difficoltà!) questo linguaggio, verranno illustrate successivamente con qualche esempio evocativo.

Operatori Lisp

CAR

Isola il primo elemento di una lista; così:
(CAR '(A B C)) vale A

CDR

Toglie dalla lista il primo elemento; così:
(CDR '(A B C)) vale (B C)

CONS

Costruisce una coppia di due elementi; così:
(CONS 'A 'B) vale (A B)
(CONS 'A NIL) vale (A.NIL) ovvero (A)

EQ

È un predicato che fornisce un valore di verità T (per vero) o F (per falso) se i due argomenti sono "atomi" uguali; così:
(EQ 'A 'B) vale F
(EQ 'A (CAR '(A B))) vale T
Si noti che, in questo secondo caso, il secondo argomento viene valutato prima che venga effettuato il confronto richiesto da EQ.

COND

Si tratta dell'unica reale struttura di controllo del Lisp, e ha come argomenti una serie di liste, ciascuna delle quali è composta da due elementi, comunque complessi.

La serie viene scandita sequenzialmente e, qualora in una di esse il primo elemento (o il suo valore) sia T (vero), allora viene eseguito il secondo elemento di tale lista. Così:

(COND ((EQ 'A 'B) (CAR '(A + B)))
(T (CAR '(C-D))))

vale C, in quanto il primo argomento ha come primo elemento della lista un'espressione EQ che risulta falsa, mentre il secondo argomento ha addirittura T come primo elemento.

Si osservi che le espressioni Lisp, al crescere della complessità, perdono leggibilità, per la presenza di parentesi che non esaltano la visualizzazione della struttura.

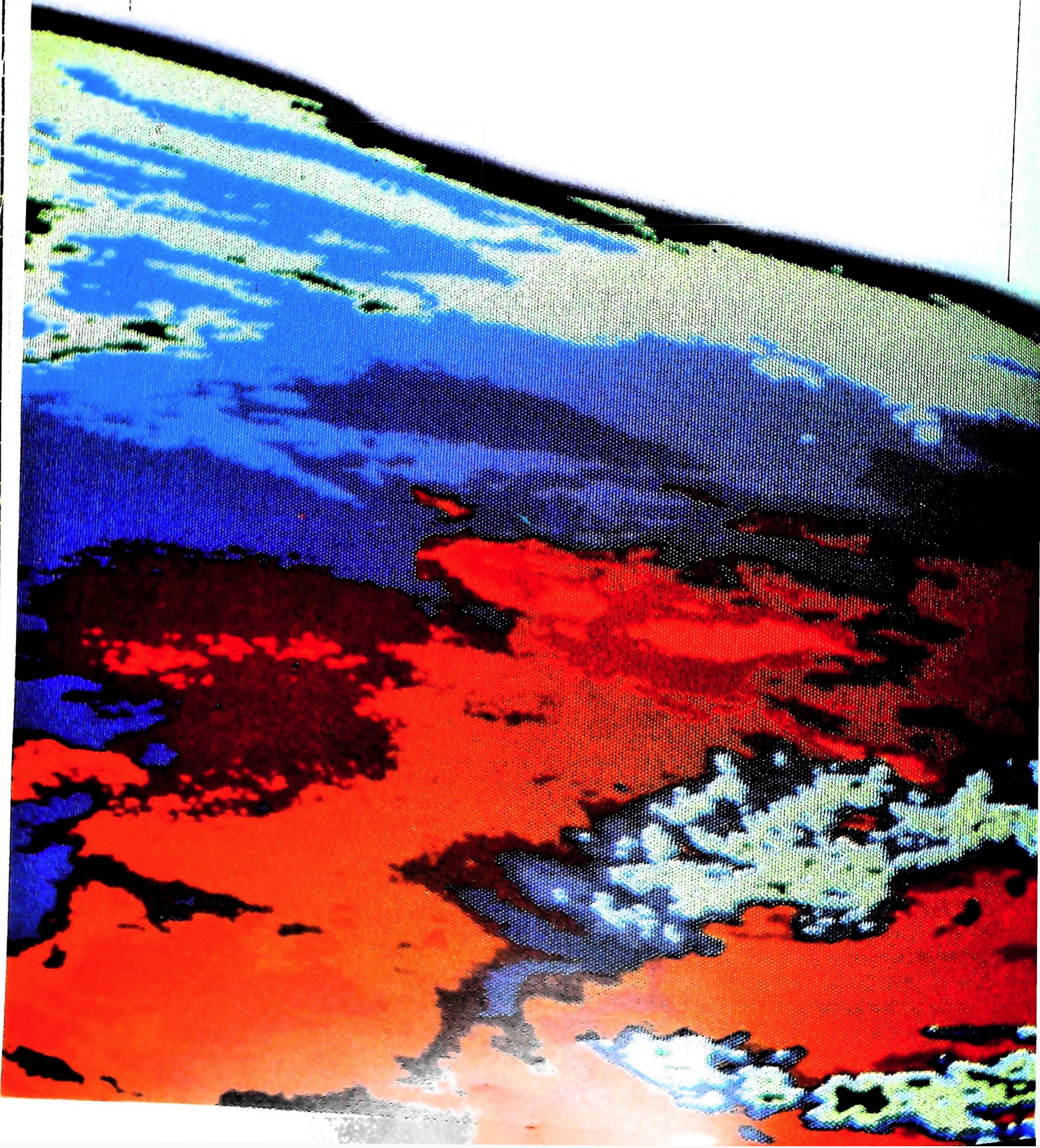
È bene, come è stato fatto nell'esempio, usare pesantemente tecniche di tabulazione per evidenziare tale struttura.

ATOM

È un predicato che vale T (vero), se l'argomento a cui è applicato è un atomo, F (falso) altrimenti; così:
(ATOM 'A) vale T
(ATOM '(A)) vale F
infatti (A) non è un atomo, ma è una lista costituita di un solo elemento.

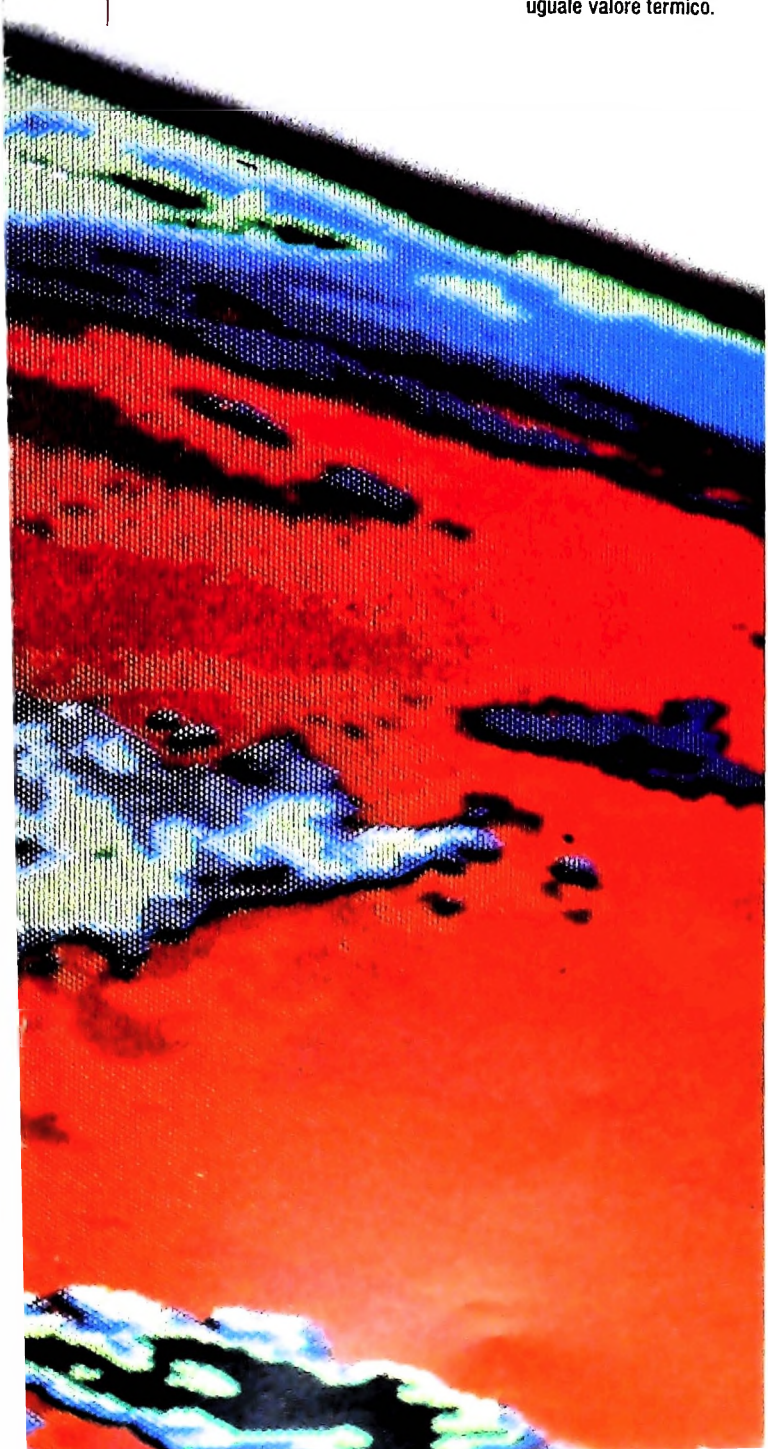
L'ELABORAZIONE DI IMMAGINI

L'informazione luminosa e cromatica dell'immagine originale viene scomposta ed elaborata in forma numerica.



Dai videogame ai grafici, agli istogrammi, dalle animazioni alle immagini di computer-art sino alle informazioni meteorologiche che ogni sera vediamo alla TV attraverso l'occhio del satellite: per il profano sono tutte "immagini elettroniche", accomunate dai tipici colori brillanti e dalle forme nitide. In realtà, in questo vastissimo campo possiamo distinguere molti aspetti. L'immagine che vediamo sul teleschermo può essere il risultato di diverse operazioni, quali quelle di sintesi che abbiamo discusso in precedenza (immagini create sulla base di descrizioni geometriche e di proprietà visive), può essere un'immagine manipolata con apparecchiature video anche molto complesse (gli "squeezezoom" che le reti private usano per arricchire le proprie sigle di effetti cromatici o deformare le figure), infine può essere il risultato di operazioni di elaborazione condotte su dati numerici che rappresentano informazione luminosa e cromatica dell'immagine originale. È in questo ultimo ambito che parliamo di elabo-

Francia, Spagna, Africa nordoccidentale e Atlantico. Rappresentazione ottenuta partendo da una sola immagine spettrale all'infrarosso elaborata al computer. Ciascun valore rappresenta zone di uguale valore termico.



razione di immagini. In questo campo le immagini rilevate da satellite costituiscono una delle più tipiche applicazioni di "image processing".

L'immagine processing si basa su sistemi che utilizzano un'apparecchiatura di visualizzazione, nella quale l'immagine viene memorizzata punto per punto, in modo che la risoluzione dei punti di campionamento corrisponda da una parte alla qualità dell'apparecchiatura di ripresa (in generale una speciale telecamera, sensibile a determinate bande di frequenza) e dall'altra alla definizione, in pixel, del video terminale.

Le apparecchiature che costituiscono un tipico sistema di image processing sono infatti tre: un sistema di acquisizione delle immagini, un sistema di elaborazione e uno di visualizzazione.

L'immagine digitale

Il concetto fondamentale dell'elaborazione di immagini acquisite è quello di immagine digitale. Un semplice esempio di immagine digitale è riprodotto nella figura in alto alla pagina seguente. La scena è costituita da un quadrato nero su sfondo bianco e una sua possibile rappresentazione in forma digitale è data dalla matrice posta al lato della scena stessa. Ogni elemento della matrice corrisponde a un'area all'interno dell'immagine rappresentata, la cui intensità luminosa è data dal valore numerico dell'elemento stesso. In questo caso, il più semplice, al valore 1 corrisponde il bianco e al valore 0 il nero.

Ogni elemento della matrice dell'immagine digitale viene fatto corrispondere a un pixel sul sistema di visualizzazione. Se nel frame buffer del sistema di elaborazione viene dedicato un solo bit a ogni pixel, è possibile ottenere soltanto due gradi di intensità luminosa (bianco e nero), ma se si dispone di un sistema che assegna più bit per pixel è possibile attribuire anche delle sfumature di colore alle singole aree dell'immagine da elaborare. Ossia è possibile disporre di una gamma di grigi la cui ricchezza dipende dalla struttura del frame buffer. Nella figura in basso, invece, si evidenzia la stessa scena con una rappresentazione digitale diversa, in cui a ogni pixel vengono cioè attribuiti 8 bit, che consentono $256 = 2^8$ possibili sfumature di intensità.

Questo concetto di digitalizzazione dell'immagine sta alla base della decodificazione degli impulsi che, per esempio, vengono inviati a terra da un satellite per telerilevamento, ossia per la raccolta di immagini del suolo terrestre effettuate dallo spazio.

L'acquisizione dell'immagine

Per acquisire un'immagine ci sono delle speciali apparecchiature che consentono di convertire una scena reale in una rappresentazione digitale elaborabile da un computer. Tutti i sistemi di acquisizione si possono ricondurre a un unico principio base: convertire l'informazione luminosa proveniente da un oggetto in un numero. Questa conversione può avveni-

re utilizzando strumenti basati su telecamere o su allineamenti di fotodiodi. Ciascuno di questi strumenti ha una propria sensibilità, e genera un segnale elettrico che è proporzionale alla quantità di luce ricevuta in ingresso. A valle dello strumento vengono collegati dei convertitori analogico/digitali, che traducono in un numero il valore di intensità elettrica ricevuto. La gamma di sfumature possibili è quindi legata alla lunghezza del numero digitale che il convertitore è in grado di produrre.

Solitamente i convertitori producono parole di 8 bit, dando luogo a una gamma di 256 valori di intensità luminosa. Per poter arrivare all'acquisizione di immagini a colori è necessario "leggere" l'intensità di luce riflessa nelle componenti cromatiche fondamentali: rosso, verde e blu. Ciò si può fare o mediante telecamere a colori e strumenti sensibili in queste

tre bande di frequenza, oppure interponendo dei filtri opportuni ed effettuando tre separate riprese.

Nel campo del telerilevamento l'acquisizione dell'informazione luminosa non è limitata alle bande di luce visibile. Infatti molti fenomeni fisici e molte proprietà del terreno sono legate a emissione di radiazione elettromagnetica nelle bande dell'infrarosso e in quella dell'ultravioletto. In questi casi i sistemi di digitalizzazione sono in grado di acquisire e digitalizzare anche in queste bande.

Vedremo ora quali operazioni verranno compiute sopra un'immagine acquisita.

L'elaborazione delle immagini

Una volta che un'immagine sia stata convertita in numeri, si tratta di affrontare svariati problemi destinati a raggiungere gli obiettivi tipici dell'elaborazione delle immagini. Infatti si tratta in primo luogo di verificare la qualità della conversione digitale effettuata.

Gli strumenti di acquisizione sono fabbricati sulla base di principi fisici, ma, come ogni strumento, sono sempre in qualche misura approssimazioni di strumenti ideali. Gli effetti di questa approssimazione consistono nella relativa affidabilità, e, in ultima analisi, nell'introduzione di dati imprecisi, errori, che vanno sotto il nome di "rumore". Così avviene per esempio nelle trasmissioni radiofoniche, in cui i disturbi che percepiamo sono dovuti alla varietà di strumenti che intervengono nel processo di trasmissione dell'informazione, ciascuno dei quali interviene come una vera e propria sorgente di rumore. La prima operazione che normalmente si compie su un'immagine acquisita è quindi quella della ricostruzione della rappresentazione numerica più probabile in assenza di errori di codifica. Questa fase è nota come filtraggio digitale, e si basa su tecniche di calcolo numerico che derivano dalle teorie di Wiener, Kalman e altri padri della teoria dell'informazione e della trasmissione. In seguito si può procedere per estrarre dall'immagine restaurata informazioni utili, quali il riconoscimento di forme particolari e la misura di regioni uniformi. Queste operazioni vengono compiute ad esempio per riconoscere in un'immagine rilevata da satellite la presenza di certe coltivazioni in una determinata stagione, e la dimensione delle aree che sono occupate da quella medesima cultura.

Ciascuna di queste operazioni può essere compiuta perché si dispone di una codifica numerica dell'immagine; per esempio il calcolo dell'area di una regione consiste molto semplicemente nel conteggio del numero di pixel che sono all'interno dei confini della regione stessa.

Questi esempi mettono in evidenza le notevoli complessità che si incontrano operando nel campo dell'elaborazione di immagini.

È chiaro che si tratta di una disciplina che richiede elevate competenze matematiche e di programmazione numerica, proprio perché consiste nella traduzione in forma algoritmica delle più importanti nozioni dell'analisi e della teoria dell'informazione.

oggetto

immagine



1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	0	0	1	1	1	1
1	1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

Sopra: immagine digitale in cui ogni elemento della matrice corrisponde a un'area all'interno dell'immagine rappresentata, e a ogni elemento è fatto corrispondere un pixel. Sotto: stessa scena, ottenuta però con una rappresentazione digitale diversa (più bit per pixel).

oggetto

immagine



255	255	255	255	255	255	255	255	255
255	255	255	126	132	255	255	255	255
255	255	119	3	2	120	255	255	255
255	255	123	2	4	119	255	255	255
255	255	255	121	118	255	255	255	255
255	255	255	255	255	255	255	255	255

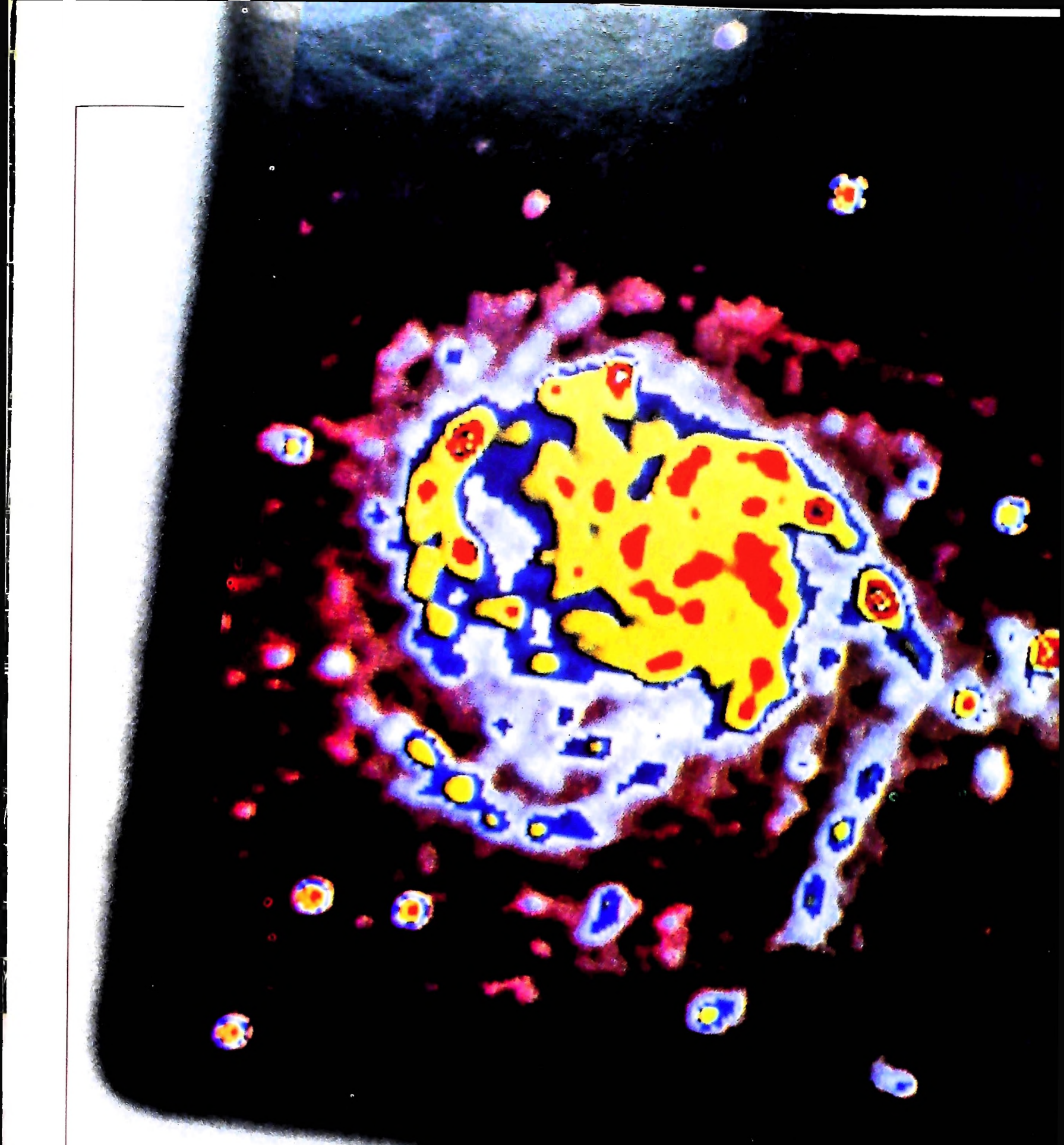


Immagine spettrale della costellazione M101, conosciuta come Orsa Maggiore. Poiché è ricca in stelle calde, questa costellazione risulta luminosissima agli ultravioletti. Le regioni più brillanti sono rosse in mappa, con intensità decrescente dal giallo al verde, al blu, fino al grigio. Le informazioni vengono elaborate dal computer in forma digitale.

La visualizzazione delle immagini

Esistono attualmente molte apparecchiature che consentono di visualizzare l'immagine elaborata.

Dai tradizionali video display delle più disparate dimensioni, alla registrazione su pellicola, dalle hard-copy collegabili direttamente al video sino ai più sofisticati prodotti fotografici per la produzione di microfilm, la tecnologia è in grado di fornire le più adeguate forme di visualizzazione e di archiviazione d'immagini per ogni tipo di necessità.

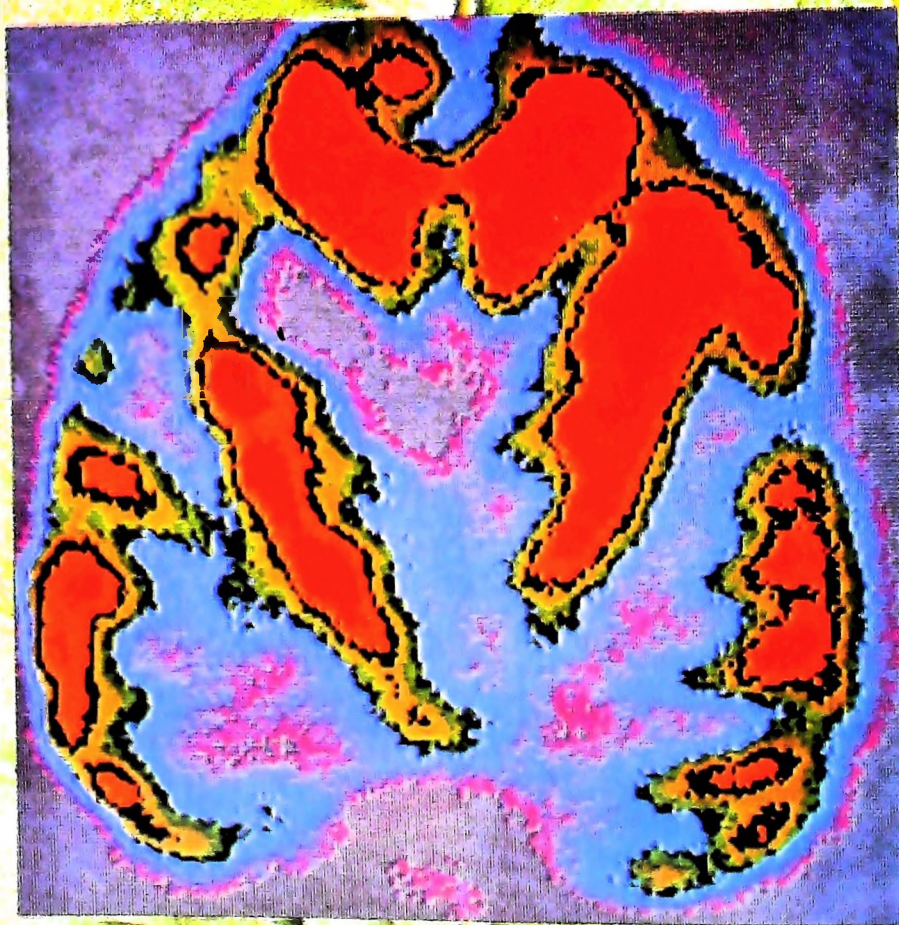
Trattando l'elaborazione di immagini, le più tipiche operazioni di visualizzazione riguardano la "falsa colorazione". In questo caso lo scopo è quello di ottenere una visualizzazione dell'immagine di più facile lettura e interpretazione. Infatti l'immagine acquisita solitamente non comprende una codifica per cui a un valore numerico di un pixel corrisponde il vero colore; si ha piuttosto una codifica che fa corrispondere al numero un valore di intensità di luce riflessa in una certa banda e in certe condizioni di illuminazione. Perciò se vo-

gliamo colorare in blu le zone di mare, in grigio quelle costruite, in verde quelle coltivate, in marrone le aree montane, dovremo far corrispondere ai pixel che registrano i valori di riflessione propri del mare il colore blu, e così via. Queste operazioni si possono compiere utilizzando le "tavole di colore" (look up table) che abbiamo illustrato nella lezione sui frame buffer.

Un'immagine elaborata al video deve poter essere riprodotta, per allegarla a una relazione, o per conservare una documentazione permanente. Si ricorre a tale scopo alla registrazione su pellicola, attuata mediante apparecchiature di "film recording". Il film recorder riconosce in ingresso il valore numerico dell'intensità di ogni pixel dell'immagine digitale e dirige un fascio di elettroni in un tubo catodico su una regione della pellicola affacciata allo schermo.

I colori in questo caso si ottengono interponendo filtri dei tre colori fondamentali ed eseguendo tre scatti in sovrapposizione delle tre immagini corrispondenti alle componenti cromatiche fondamentali.

Anche in questo caso l'informazione luminosa e cromatica dell'immagine originale viene scomposta ed elaborata in forma numerica dal computer.



LA FAMIGLIA DEI PERSONAL COMPUTER OLIVETTI



FRIENDLY & COMPATIBLE

Questa famiglia di personal compatibili tra loro e con i più diffusi standard internazionali, non ha rivali per espandibilità e flessibilità. Prestazioni che su altri diventano opzionali, sui personal computer Olivetti sono di serie. Per esempio M24 offre uno schermo ad alta definizione grafica, ricco di 16 toni o di 16 colori e con una risoluzione di 600x400 pixel; mentre la sua unità base dispone di 7 slots di espansione, fatto questo che gli consente di accettare schede di espansione standard anche se utilizza un microprocessore a 16 bit reali (INTEL 8086). Ma ricchi vantaggi offrono anche tutti gli altri modelli.

Basti pensare che tutte le unità base includono sia l'interfaccia seriale che quella parallela. Oppure basti pensare all'ampia gamma di supporti magnetici: floppy da 360 a 720 KB o un'unità hard disk (incorporata o esterna) da 10 MB. La loro compatibilità, inoltre, fa sì che si possa far uso di una grande varietà di software disponibile sul mercato. Come, ad esempio, la libreria PCOS utilizzabile anche su M24. Come le librerie MS-DOS®, CP/M-86® e UCSD-P System®, utilizzabili sia da M20 che da M21 e M24.

MS-DOS è un marchio Microsoft Corporation
 CP/M-86 è un marchio Digital Research, Inc.
 UCSD-P System è un marchio
 Regents of the University of California

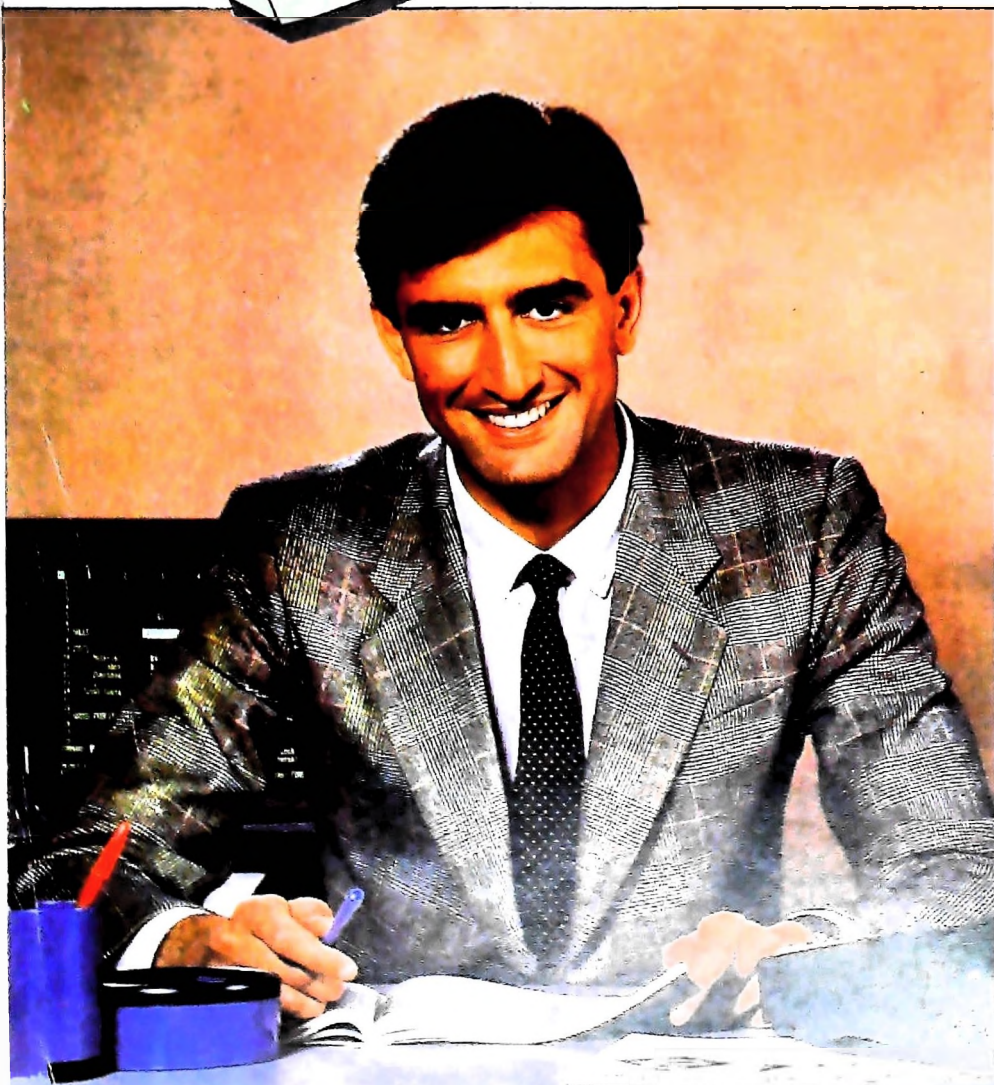
olivetti

Per maggiori informazioni inviare il coupon a Olivetti,
 Divisione Personal Computer Via Meravigli 12, 20125 Milano

NOME _____
 INDIRIZZO _____
 CITTA' _____
 TELEFONO _____

— UN NUOVO MODO DI USARE LA BANCA. —

CONSAVALERIALO



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

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

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

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

Veniteci a trovare, ci conosceremo meglio.

 **BANCO DI ROMA**
CONSCIAMOCI MEGLIO.

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

421958

F4 F5 F6 F7

43 CORSO PRATICO COL COMPUTER

di **GIANNI DEGLI ANTONI**

è una iniziativa
FABBR EDITOR
in collaborazione con
BANCO DI ROMA
e **OLIVETTI**

**IN EDICOLA
DAL GENNAIO
28**

**I GRANDI TEMI
DELLA MEDICINA
PER CURE, PREVENZIONE, MEDO**

**IL SISTEMA
NERVOSO**

per il
libro
di 100
pag.

**IL DIZIONARIO
DELLA
MEDICINA**

**IL DIZIONARIO
DELLA
MEDICINA**

PER IL MEGLIO

