

12 CORSO PRATICO COL COMPUTER

421664

è una iniziativa
FABBRI EDITORI

in collaborazione con
BANCO DI ROMA

e **OLIVETTI**

F4

F5

F6

F7

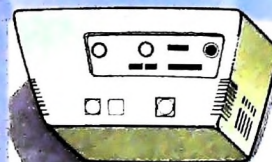
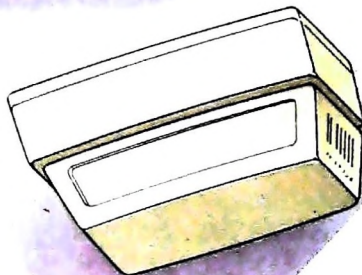
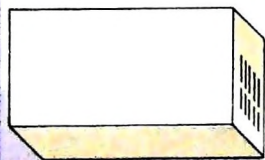
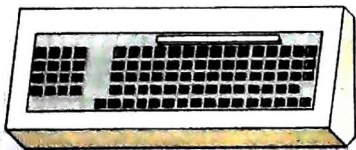
F8

diretto da **GIANNI DEGLI ANTONI**

CA DEL

BATTERY LOW

**I SISTEMI INTELLIGENTI:
I NUOVI CAMPI
DELL'INTELLIGENZA ARTIFICIALE**



**FABBRI
EDITORI**

**È in edicola il quarto numero
di LIBRERIA DI SOFTWARE
dedicato ai
"Programmi di ordinamento e ricerca"**

Ringraziamo per la cortesia e l'attenzione tutti i lettori che hanno scritto o telefonato per farci avere i loro commenti sui precedenti numeri di "Libreria di software".

Tutti noi sappiamo quanto sia difficile costruire un programma e come ogni programma sia passibile di modifiche e miglioramenti. L'attenta partecipazione dei lettori ci è di conforto nel nostro lavoro e di stimolo al conseguimento di un sempre più alto standard di qualità.

Invitiamo tutti a continuare a scriverci o a telefonarci per comunicarci commenti, difficoltà, suggerimenti per superarle.

Pubblicheremo man mano eventuali modifiche suggerite per i programmi che possano facilitarne sia il caricamento sia l'uso.

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

Testi
Eidos (TIZIANO BRUGNETTI), ADRIANO DE LUCA, ENNIO PROVERA, LUCA SPAMPINATO
Etnoteam (ADRIANA BICEGO)

Tavole
Logical Studio Communication
Il Corso di Programmazione e BASIC è stato realizzato da Etnoteam S.p.A., Milano
Computer grafica 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 BREGGÉ

Segretarie di Redazione
RENATA FRIGOLI
LUCIA MONTANARI

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

Corso Pratico col Computer - Copyright © sul fascicolo 1984 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Etas S.p.A., Milano - Copyright © sull'opera 1984 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Etas S.p.A., Milano - Prima Edizione 1984 - Direttore responsabile GIOVANNI GIOVANNINI - Registrazione presso il Tribunale di Milano n. 135 del 10 marzo 1984 - Iscrizione al Registro Nazionale della Stampa n. 00262, vol. 3, Foglio 489 del 20.9.1982 - Stampato presso lo Stabilimento Grafico del Gruppo Editoriale Fabbri S.p.A., Milano - Diffusione Gruppo Editoriale Fabbri S.p.A. via Mecenate, 91 - tel. 50951 - Milano - Distribuzione per l'Italia: A. & G. Marco s.a.s., via Fortezza 27 - tel. 2526 - Milano - Pubblicazione periodica settimanale - Anno I - n. 12 - 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.

LA LEGGE DI DE MORGAN

Come progettare i propri circuiti logici usando componenti reperibili in commercio.

Molto spesso, quando si sta realizzando un progetto con circuiti digitali, ci si trova dinanzi alla necessità di avere a disposizione dei circuiti che però non esistono in commercio. Questa difficoltà è solo apparente, come mostreremo qui di seguito.

Facciamo un esempio: abbiamo due elementi ATTIVI NEGATIVI A e B e vogliamo che quando almeno uno o tutti e due sono ATTIVI possano attivare un terzo elemento C con entrata ATTIVA POSITIVA (figura 1).

Vediamo come si procede a determinare il circuito logico necessario a risolvere il problema. Dall'enunciato risulta immediatamente che il circuito che realizza la funzione richiesta non può essere altro che un circuito OR. Per quanto riguarda la determinazione delle sue entrate e uscite, bisogna sempre rifarsi alle condizioni del problema, nel modo seguente: in entrata abbiamo due segnali ATTIVI NEGATIVI e, di conseguenza, sono necessarie due linee con cerchietti, mentre in

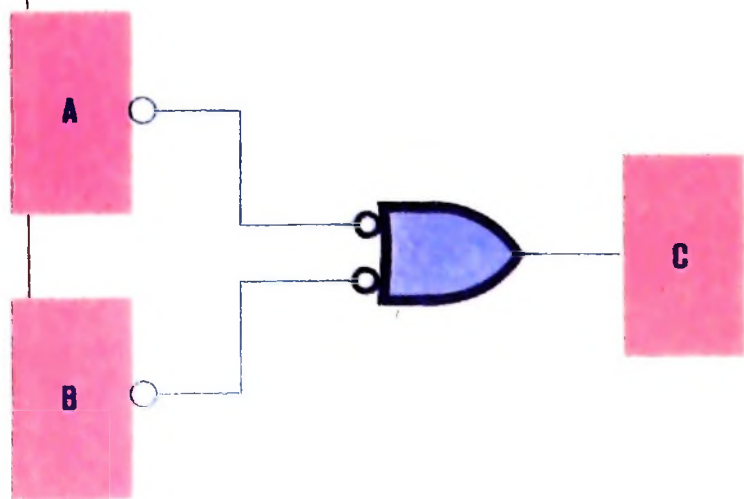
uscita abbiamo un segnale ATTIVO POSITIVO.

Ma un circuito OR del tipo rappresentato in figura non esiste in commercio: i circuiti logici commerciali sono rappresentati in figura 2, naturalmente con diverse entrate e diverse caratteristiche, secondo le famiglie. Il metodo per trasformare un circuito logico non esistente in commercio in uno esistente è quello di applicare la "legge di De Morgan". Questa si enuncia così, in modo semplicissimo:

- trasformare gli OR in AND e viceversa;
- trasformare le linee ATTIVE POSITIVE in linee ATTIVE NEGATIVE e viceversa.

Ciò è rappresentato nella Tabella di trasformazione, la cui colonna di sinistra mostra i simboli dei circuiti logici di partenza, quella di destra il risultato dell'applicazione della legge di De Morgan. Approfondendone l'analisi, notiamo che quando, figura 1, i circuiti logici sono in concordanza con i loro stati ATTIVI si avrà anche una concordanza grafica, per

① un nuovo circuito logico non in commercio



② i circuiti logici in commercio

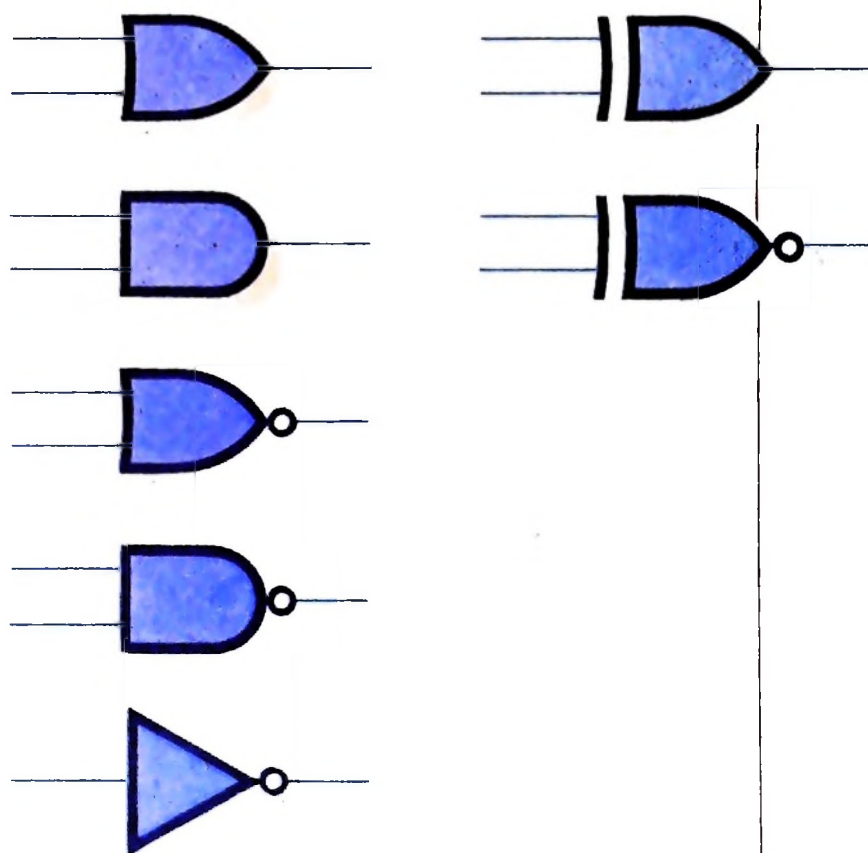

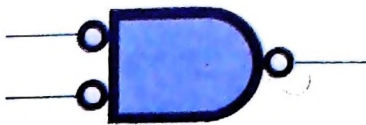


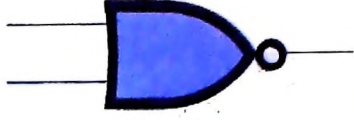
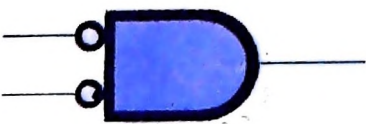
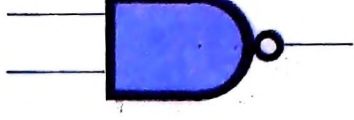

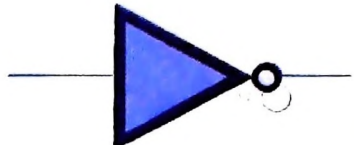
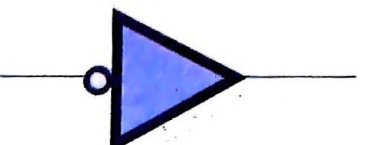


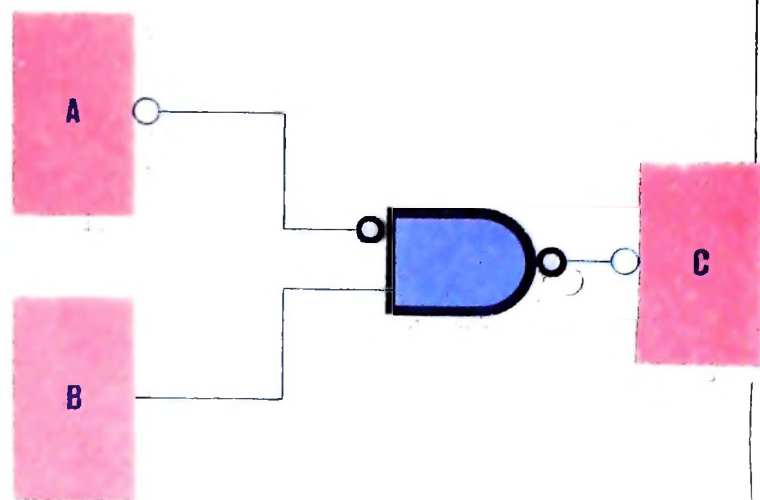
Tabella di trasformazione

OR		
AND		
NOR		
NAND		
INV		

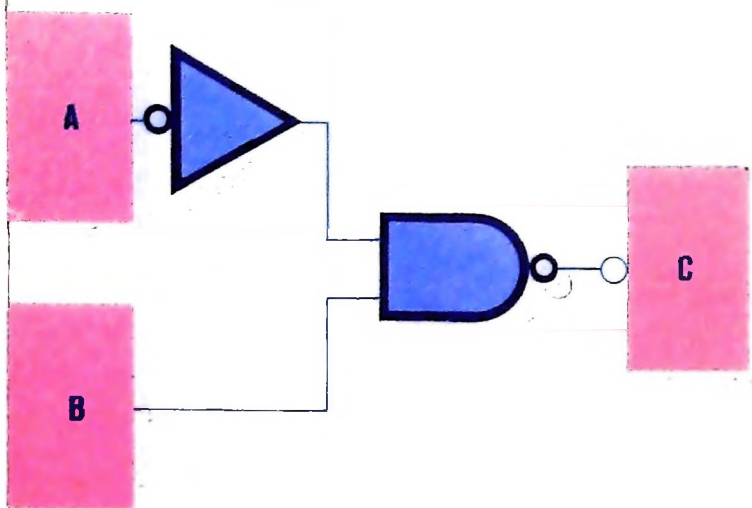
cui a linee ATTIVE NEGATIVE in uscita corrispondono linee ATTIVE NEGATIVE in entrata e viceversa. Ma questa non è la norma, come mostra l'esempio della figura 3, in cui vi sono due elementi A e B, il primo ATTIVO NEGATIVO e l'altro ATTIVO POSITIVO. Imponiamo che, quando tutti e due sono ATTIVI, attivino un terzo elemento C ATTIVO NEGATIVO. È chiaro che questo circuito non può essere altro che un circuito AND; dal momento però che non esiste alcun circuito che abbia entrate ATTIVE diverse, dobbiamo agire sui segnali per trasformarlo in un circuito logico reperibile in commercio. Esistono due soluzioni. La prima è quella di porre un INVERSORE in serie al comando dell'elemento A, per cui il circuito si trasforma come indicato nella figura 4, e il problema è risolto in quanto il circuito logico è un NAND. La seconda soluzione è quella di porre l'inversore sulla linea B (figura 5) e, secondo la legge di De Morgan, il circuito si trasforma in OR (vedi Tabella). In ambedue i casi si ha concordanza grafica fra linee ATTIVE.

Rimane ancora un terzo esempio, molto importante, che continua a provocare errori di interpretazione degli schemi elettrici dei circuiti digitali. Riprendiamo il caso di due circuiti A e B, di cui il primo ATTIVO NEGATIVO e il secondo ATTIVO POSITIVO, e un terzo elemento C, ATTIVO

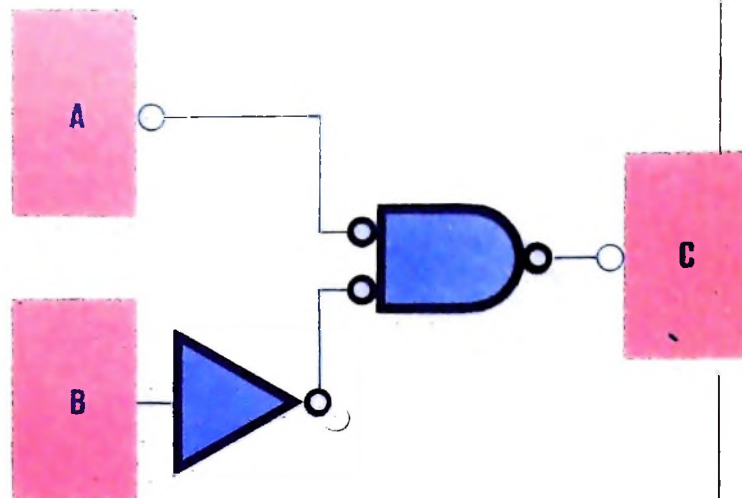
③ circuiti con entrate di tipo diverso



④ una prima realizzazione del circuito di figura 1



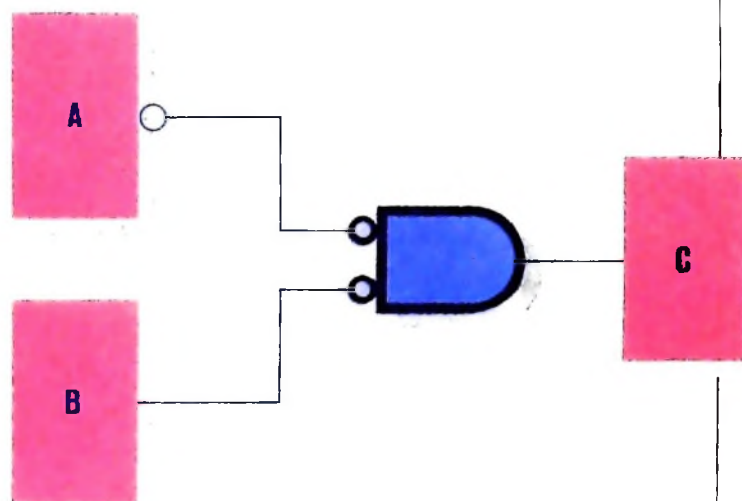
⑤ una seconda realizzazione del circuito di figura 1



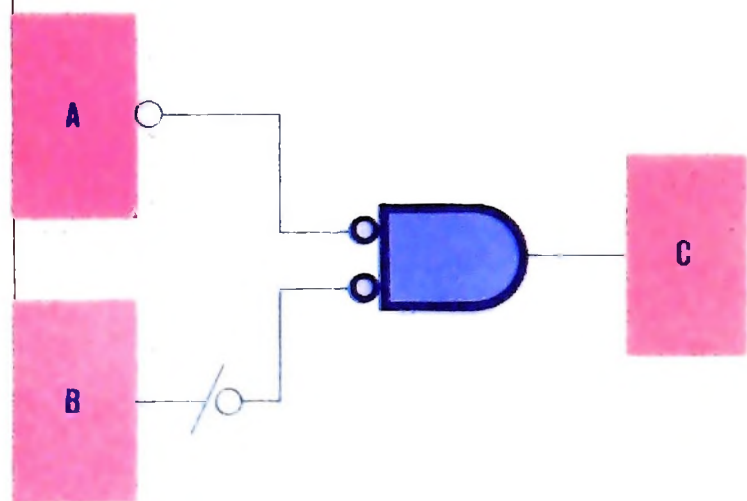
POSITIVO. Imponiamo che quando A sia ATTIVO e B NON ATTIVO si attivi C (figura 6). Naturalmente, data la coincidenza dei segnali, il circuito è un AND, però da parte del circuito A ATTIVO NEGATIVO gli arriva uno zero. Avremo quindi una linea con circoletto (che, ricordiamo, significa ATTIVO NEGATIVO) sia all'uscita di A sia all'entrata di AND, mentre da B, che normalmente è ATTIVO POSITIVO, in questo caso, volendo B NON ATTIVO, dobbiamo porre un circoletto all'entrata di AND.

Abbiamo cioè una discordanza grafica sulla linea B, che è ATTIVA POSITIVA in uscita da B e ATTIVA NEGATIVA in entrata in AND. Per ovviare a questo inconveniente, si è

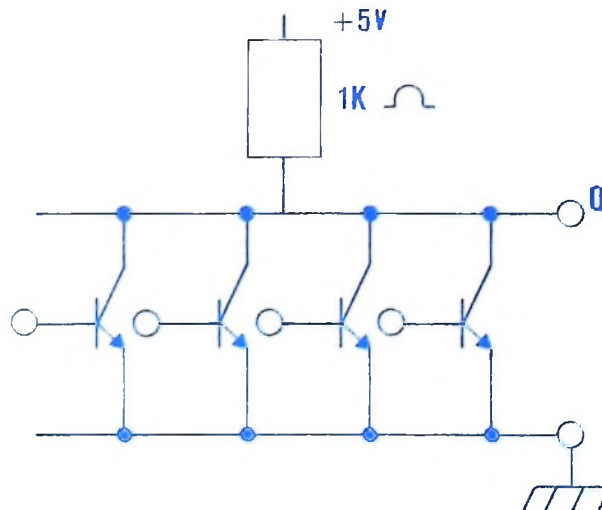
⑥ un nuovo tipo di linea non concordante

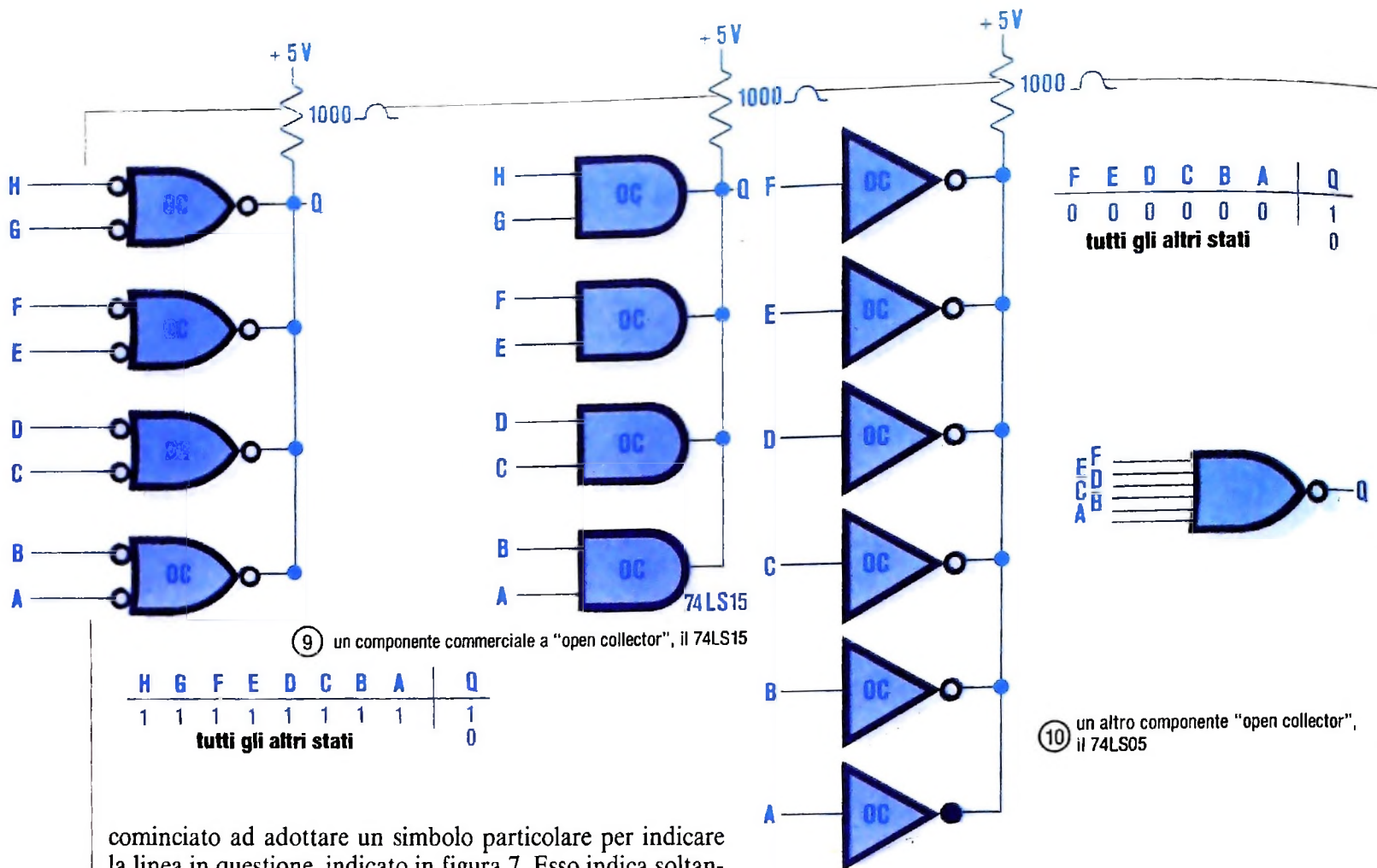


⑦ il simbolo della linea non concordante



⑧ il dispositivo "open collector"



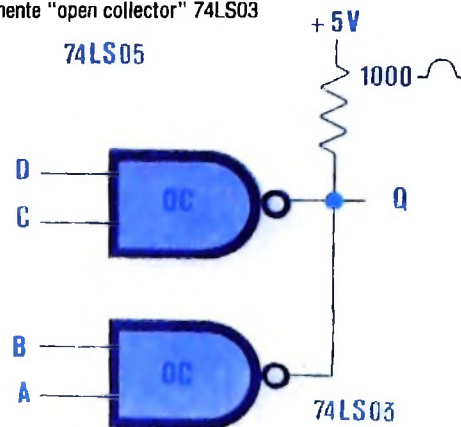


⑨ un componente commerciale a "open collector", il 74LS15

H	G	F	E	D	C	B	A	Q
1	1	1	1	1	1	1	1	0
tutti gli altri stati								1

⑩ un altro componente "open collector", il 74LS05

⑪ il componente "open collector" 74LS03



D	C	B	A	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

cominciato ad adottare un simbolo particolare per indicare la linea in questione, indicato in figura 7. Esso indica soltanto che in questo circuito si usa la linea B, ATTIVA POSITIVA, per attivare il circuito seguente. Per quanto riguarda il circuito commerciale necessario, dalla Tabella di conversione risulta essere un NOR.

Gli elementi a collettore aperto

Come ultima analisi dei circuiti logici esaminiamo ora gli elementi a collettore aperto. Questi circuiti sono molto utili quando ci si trova in circostanze per cui si ha bisogno di un circuito logico con molte entrate. Consideriamo una resistenza, collegata da una parte all'alimentazione e dall'altra a più transistor, come in figura 8. Nel punto di unione Q avremo uno "zero" logico quando almeno uno dei transistor ha un "uno" logico nella base, mentre avremo un "uno" logico quando tutti i transistor hanno uno "zero" logico nella base. Come è facile capire, allo stesso punto Q si possono collegare, in teoria, un numero infinito di transistor e, come risultato finale, avremo un AND a molte entrate, a piacere.

Questi circuiti logici corrispondono ai dispositivi di tipo 74LS15 disponibili in commercio. Sono mostrati in figura 9 assieme alla tabella della verità e all'equivalente di De Morgan (a sinistra). Un esempio di dispositivo 74LS05, anch'esso con il circuito equivalente di De Morgan, un NOR in questo caso, e la sua tabella della verità è mostrato in figura 10.

Bisogna in ogni modo stare attenti nel comporre fra di loro circuiti integrati con collettori aperti perché, a volte, danno risultati strani che pongono problemi di determinazione. La tabella della figura 11, a esempio, mostra nove possibilità di "uno" logico e sette di "zero" logico.

I SISTEMI INTELLIGENTI

**I nuovi campi dell'intelligenza artificiale:
la programmazione esplorativa e i sistemi esperti.**

In precedenza si è visto quali sono le idee che hanno permesso l'aprirsi di un certo numero di campi di ricerca che oggi viene collettivamente indicato con il nome di Intelligenza Artificiale (A.I.).

Si è parlato dell'ambizioso progetto iniziale che prevedeva la realizzazione di sistemi in grado di esibire un comportamento "intelligente", simile cioè a quello delle persone umane e di come il conseguimento di tali obiettivi sia stato ostacolato, se non impedito, dal manifestarsi di problemi di complessità in ogni passo del ciclo di sviluppo. Sono state indicate complessità di analisi, di definizione del modello e di realizzazione dei programmi.

L'effetto di questa situazione fu che, lentamente, la mentalità dei ricercatori dei vari campi dell'A.I. cambiò sensibilmente. Ogni gruppo, alle prese con le difficoltà dei propri studi, restrinse progressivamente il campo di interesse e mantenne relazioni solo con i gruppi dediti a studi più affini. Ciascuno delineò obiettivi più localizzati e specifici per la propria ricerca. Quelli che si configuravano inizialmente come sottoproblemi, erano diventati, già nei primi anni '70, vere e proprie discipline autonome.

Nel corso degli anni '70, emerse sempre più chiaramente un secondo effetto, forse anche più importante per l'evoluzione informatica in generale, degli sforzi fatti dai ricercatori nel

Una stazione di lavoro per la progettazione di nuovi circuiti integrati: sul terminale grafico a colori compaiono gli schemi (o "celle" elementari) contenuti nella memoria del calcolatore cui il terminale è collegato. Tali schemi sono i "mattoni elementari" dei nuovi circuiti.



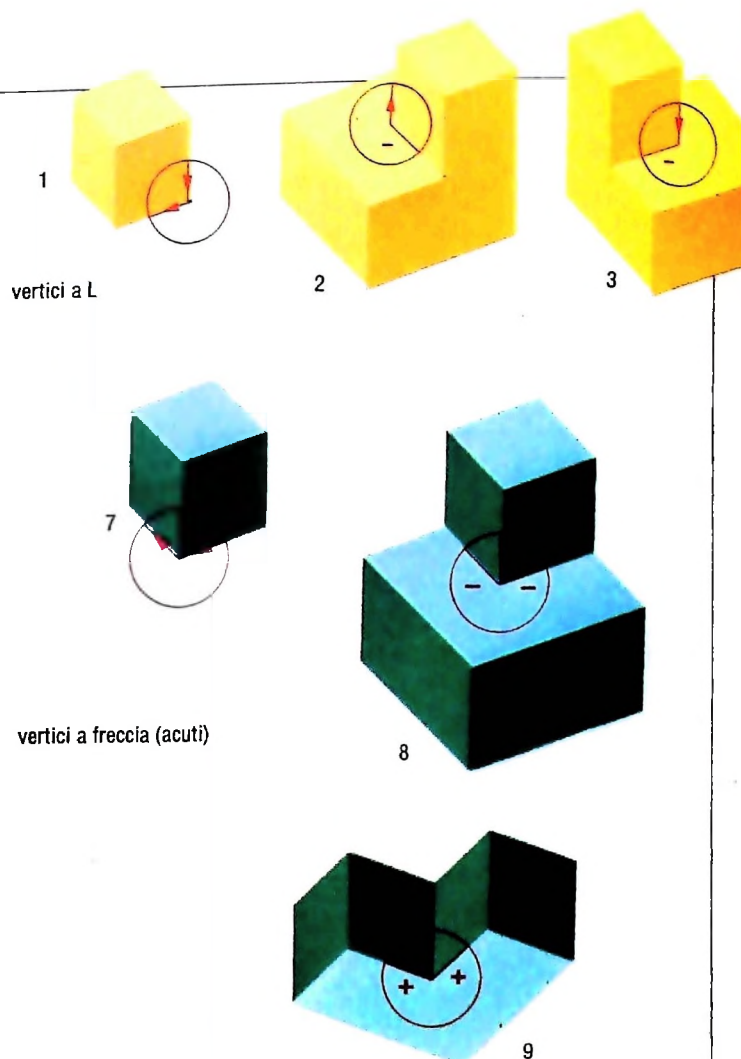
campo dell'A.I. Ci si accorgeva che, nel misurarsi con i problemi che quotidianamente si dovevano affrontare in ogni sottocampo, erano stati messi a punto nuovi metodi, tecniche e linguaggi che potevano avere interesse ben più generale. Il nostro discorso a questo punto si divide in due parti: si parlerà dello stato dell'arte e di alcuni risultati nei principali campi di applicazione e di ricerca in A.I., nonché del modo in cui sono stati sviluppati e vengono utilizzati gli strumenti propri dell'A.I. anche in ambienti diversi. Affrontiamo per primo l'argomento relativo ai metodi e agli strumenti.

Un modo nuovo di programmare

L'esportazione verso le altre discipline informatiche dei risultati metodologici ottenuti in questo campo, dei linguaggi utilizzati e degli ambienti per lo sviluppo dei programmi, collaterali rispetto alle mete che l'A.I. si era posta, fu inizialmente implicita. Gli stessi ricercatori che, in contesti differenti da quelli dell'intelligenza artificiale, si trovavano a risolvere problemi di disegno e di programmazione non banali, utilizzavano in modo naturale e redditizio gli strumenti in loro possesso.

Ma il riconoscimento esplicito dell'utilità, e in molti casi della necessità, degli strumenti sviluppati in ambito A.I. si ebbe quando incominciò a presentarsi sempre più spesso, sul finire degli anni '70, il problema di realizzare sistemi informativi che richiedevano un atteggiamento nuovo. Consideriamo ad esempio il caso dei grossi programmi che controllano i complessi sistemi automatici di estrazione del petrolio, che sono composti di apparecchiature meccaniche ed elettroniche che spesso vengono sostituite e aggiornate; oppure ai sistemi che assistono i progettisti di circuiti elettronici integrati su larghissima scala: esistono numerose e complicate regole che vanno rispettate e che variano velocemente con la tecnologia; oppure ancora i sistemi che provvedono, colloquiando con l'editore, alla impaginazione di quotidiani e periodici. In tutte queste applicazioni, come in molte altre, il problema che il sistema da realizzare deve risolvere non è completamente definito all'inizio del lavoro; anzi spesso l'utente non sa nemmeno esattamente che cosa si aspetta dal prodotto. Ciò che il programmatore deve fare è offrire proposte-tentativi che, con l'uso, permettano di definire nuove esigenze e dare luogo ad un altro tentativo. Questo ciclo può non finire mai, ma il programma deve essere utilizzabile in ogni sua parte già sviluppata mentre se ne continua l'evoluzione.

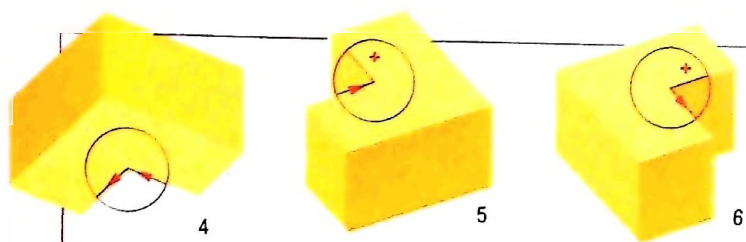
Le metodologie di sviluppo software che sono state messe a punto negli anni passati, pur rimanendo di grande aiuto nell'affrontare problemi "tradizionali" come quelli di EDP (*Electronic Data Processing*) e di automazione industriale classica, non sono completamente adeguate per superare le difficoltà di cui si parlava. Permettono di ottenere, a partire da specifiche altamente particolareggiate e formalizzate, programmi modulari e leggibili. L'obiettivo è quello di evitare che ogni parte del programma faccia qualcosa di non previsto e influenzi quindi in modo non controllabile il comportamento dell'intero sistema. Il punto è però che gli stessi criteri



Riconoscimento di un oggetto tridimensionale a partire da uno schema bidimensionale. Il computer assume che ogni segmento nel disegno sia o un bordo occultante di un oggetto in prospettiva o un bordo che punta verso l'osservatore (bordo convesso) o in direzione opposta (bordo concavo). Ogni bordo occultante è indicato da una freccia; ogni bordo convesso interno da un segno +; ogni bordo concavo interno da

un segno -. Più bordi confluiscono in un vertice, di cui esistono 16 tipi diversi. Questo schema di etichettatura fu proposto inizialmente da D.A. Huffman del MIT e da M.B. Clowes dell'Università del Sussex. Esso fu poi generalizzato per permettere il riconoscimento di ombre, crepe e altri tipi di corpi. La figura mostra i 16 tipi di vertici, suddivisi in quattro categorie (a L, a freccia, a forcella e a T).

che permettono di raggiungere questo obiettivo impongono una notevole *rigidità* nel disegno e nell'implementazione del sistema che si rivela salutare nei casi normali, ma che impedisce di effettuare velocemente, e con poca fatica, modifiche sostanziali. Lo stesso discorso vale per i linguaggi di programmazione: l'uso di linguaggi "classici", quali ad esempio Fortran o Pascal, garantisce una maggiore robustezza dei prodotti, e talvolta una maggiore efficienza, ma invariabilmente ostacola la flessibilità e la dinamicità dello sviluppo dei programmi. L'insistenza nell'uso di metodologie e strumenti tradizionali, anche nei casi in cui questo tipo di flessibilità era requisito fondamentale, ha portato spesso al fallimento del progetto; ciò avveniva ovviamente quando il realizzatore si accorgeva che le specifiche sulle quali si era basato andavano pesantemente riviste e che di conseguenza avrebbe dovuto ricominciare da capo. È chiaro quindi che le tecniche, i linguaggi e gli strumenti dell'A.I. sono ben più adeguati in questo campo di applicazioni, recentemente indi-



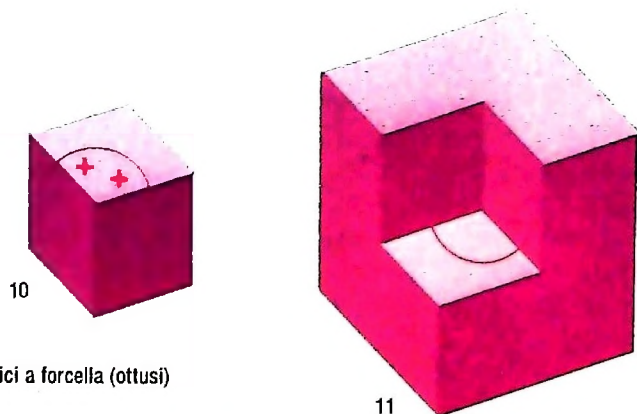
cato con il significativo nome di *Exploratory Programming* (programmazione esplorativa).

I primi risultati dei nuovi metodi

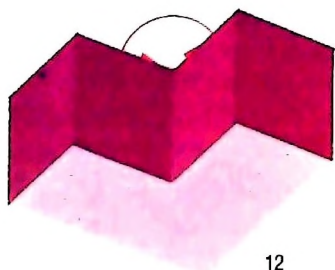
Benché ormai la pratica comune abbia molto assottigliato il confine tra l'atteggiamento classico e quello proprio dell'A.I., tentiamo di identificare ora quali sono i principali risultati esportati dall'A.I. verso le altre discipline e quali sono stati gli stimoli che hanno permesso il conseguimento di tali risultati in questo ambito. Come si è detto, il principale ostacolo, per i ricercatori A.I., è la *complessità*. È quindi facile classificare i loro successi, come è stato fatto nel precedente numero rispetto alle difficoltà, in base alle complessità incontrate nelle varie fasi di ciò che si può definire il ciclo di ricerca in campo informatico sperimentale.

1) Analisi: per esplorare ed analizzare i meccanismi che stanno alla base di ognuna delle attività umane che si intendono simulare, sono chiaramente necessari potenti strumenti di indagine. La necessità di ricorrere a metodi d'analisi propri di altre discipline scientifiche e umanistiche, e spesso all'intervento diretto di ricercatori esterni, ha reso gli studiosi A.I. piuttosto aperti al dialogo. Spesso infatti è solo la discussione con biologi, fisiologi e matematici che permette di chiarire quali sono i punti chiave della ricerca stessa. Si può dire quindi, e questo è un aspetto abbastanza trascurato, che da queste esperienze è nata la disponibilità ad utilizzare linguaggi specifici e tecniche esplorative non proprie e questo caratterizza i migliori ricercatori A.I. Come vedremo, in molti dei campi di applicazione odierni, l'essere in grado di assorbire conoscenza all'interno di contesti specifici attraverso il dialogo con esperti è fondamentale.

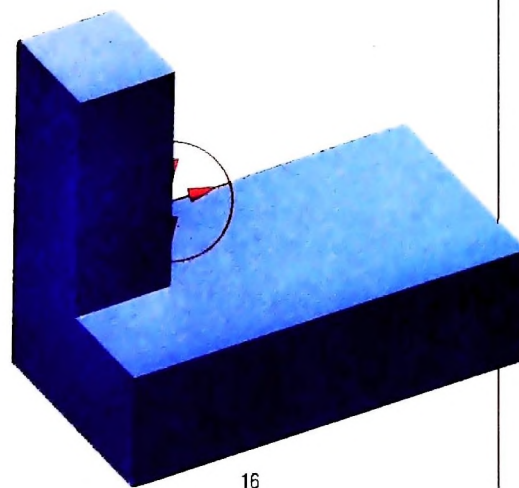
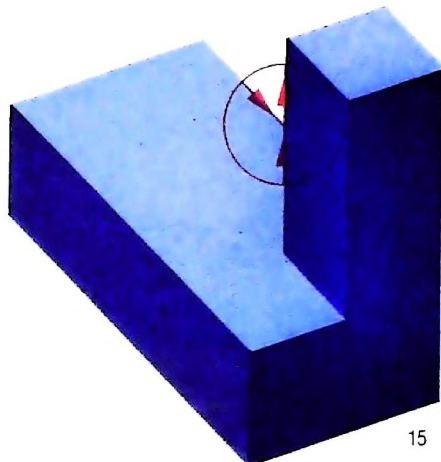
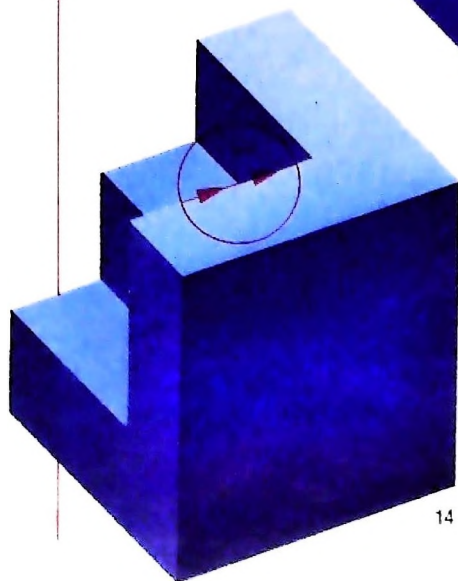
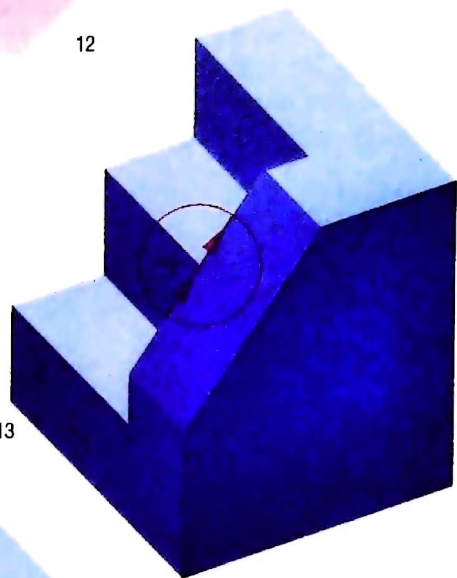
2) Individuazione di un modello: ogni tentativo di costruzione di un modello, anche parziale e approssimativo, di sistemi di complessità elevatissima come quelli dell'attività intellett-



vertici a forcina (ottusi)



vertici a T



tuale umana, contiene una quantità enorme di informazioni spesso disomogenee e sempre collegate fra loro in modo intricato. Le conoscenze relative ai fenomeni da simulare devono essere quindi organizzate in modo preciso, rappresentate in modo da non ammettere ambiguità e conservate in modo da poter essere facilmente rintracciate e raggruppate in base ai loro mutui legami. Si presenta anche il problema di impedire che nuove informazioni acquisite possano creare contraddizioni e conflitti con quelle già disponibili: le dimensioni di tale bagaglio di informazioni impediscono spesso che questo genere di operazioni di organizzazione e di verifica vengano svolte manualmente con carta e penna. Si pensi ad esempio quante informazioni devono essere a disposizione di un sistema in grado di riconoscere degli oggetti di uso corrente (tavoli, sedie, suppellettili e candelabri) sulla base di immagini di una stanza di una casa qualunque ottenute attraverso una telecamera. Per ogni oggetto riconoscibile deve essere nota la fisionomia in termini di oggetti più semplici, fino al livello di triangoli, sfere e cilindri. Il sistema deve essere in grado di fare considerazioni sulla natura degli oggetti in base al loro colore e alla reazione dei vari materiali alla luce. I ricercatori nei vari campi dell'A.I. sono stati quindi portati a sviluppare sistemi automatici per la rappresentazione e l'organizzazione di conoscenze in generale e per campi specifici. Tali sistemi permettono di controllare e manipolare grandi insiemi di informazioni basandosi sulla struttura che in esse viene riconosciuta dall'esperto del campo in cui si agisce. La comunicazione con l'utente avviene tramite linguaggi che permettono di rappresentare in modo naturale le informazioni e la loro struttura. Questi sono detti appunto linguaggi per la rappresentazione della conoscenza e sono attualmente al centro dell'interesse di molti gruppi di ricerca, tanto da costituire una vera e propria disciplina autonoma.

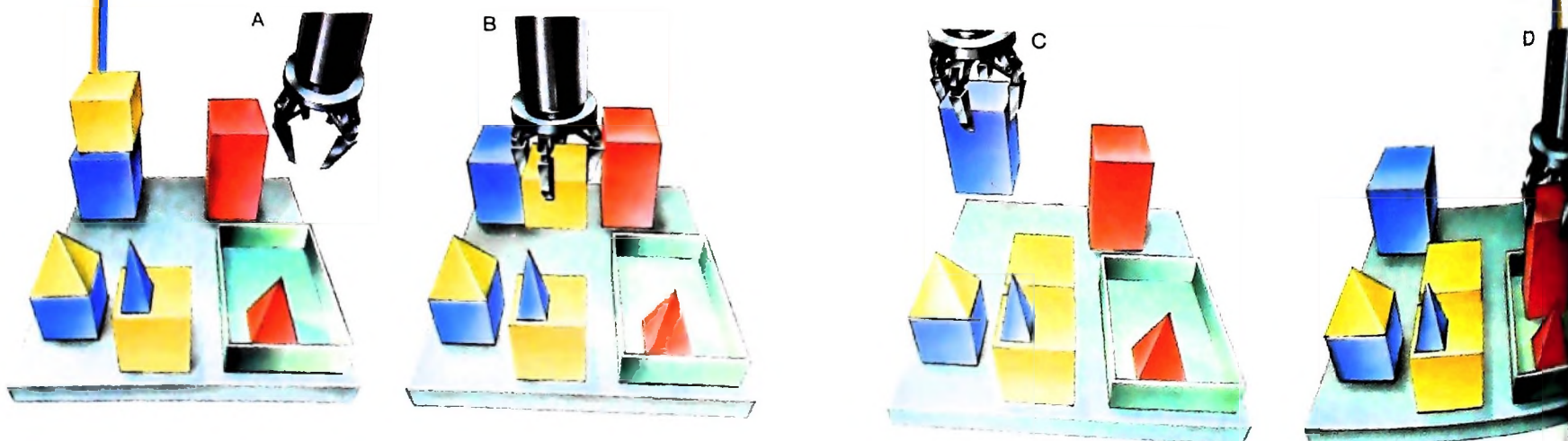
3) Realizzazione della simulazione: la complessità dei pro-

grammi che devono essere prodotti in A.I. è quasi sempre elevatissima. Come si è detto inoltre, il problema principale è costituito dal fatto che chi disegna il sistema spesso non dispone di un insieme dettagliato di specifiche e comunque sa che il suo prodotto sarà soggetto a frequentissime modifiche anche sostanziali. Certamente il programma avrà vita breve, ma dovrà essere sviluppato da vari gruppi di persone cooperanti. È chiaro quindi che il rapporto tra programmatore e calcolatore deve essere il più amichevole possibile: chi disegna ed implementa deve potersi concentrare sugli aspetti concettuali del problema senza essere ostacolato dai dettagli tecnici del programmare. Si può dire che ogni ricercatore tende a crearsi il proprio ambiente favorevole e i propri linguaggi di programmazione. Raramente vengono stabiliti degli standard e per lunghi anni la diffusione di metodologie e tecniche è stata affidata alla comunicazione tra persone singole o a brevi note interne. Solo recentemente, col diffondersi in ogni campo dell'informatica dei risultati dell'A.I., sono apparsi articoli e libri che trattano di questo tipo di strumenti per lo sviluppo dei programmi. L'intervento dell'A.I. nel campo dello sviluppo software è stato rilevante per quanto riguarda i linguaggi di programmazione, le tecniche con cui costruire programmi che ottengano nel modo migliore certi risultati, i sistemi che assistono il programmatore durante le varie fasi della sua attività e addirittura la progettazione di sistemi di calcolo particolari e di dispositivi di comunicazione uomo-macchina. Molti di questi interventi sono stati ormai completamente assorbiti e non conservano quasi traccia della loro origine avendo subito una evoluzione anche notevole indipendente dalla ricerca in ambito A.I. Questo insieme di argomenti merita senza dubbio una trattazione estesa e verranno quindi trattati nei prossimi articoli di questa serie. In futuro si parlerà anche dello stato di evoluzione e dei concetti base di alcuni dei principali campi di ricerca in A.I.

Il programma SHRDLU

All'Università di Stanford Terry Winograd ha elaborato un programma chiamato SHRDLU, che permette a un robot dotato di visione di riconoscere degli oggetti e di eseguire determinate operazioni. A sinistra, l'operatore chiede al robot di sollevare il prisma blu, operazione non semplice in quanto esso si trova sotto il cubo giallo (in A). Per eseguire il comando, il robot deve prima spostare il cubo giallo (B) e quindi può alzare il prisma blu (C). Se si chiede al robot di spostare la piramide, senza specificare quale, SHRDLU non può eseguire l'operazione in quanto le informazioni che ha sono insuffi-

cienti. Se l'operatore chiede di spostare il prisma rosso nella scatola, il robot capisce ed esegue il comando (in D). Tutto ciò presuppone che il robot "veda" e classifichi sia determinate forme geometriche (viste in prospettiva) sia alcuni colori principali, e la cosa richiede un programma di complessità notevole, oltre ad appositi sensori e un hardware specializzato nell'elaborazione di immagini. A tale scopo si usano spesso "parallel processors" di architettura diversa da quella convenzionale di Von Neumann, più adatta all'elaborazione sequenziale che a quella parallela.



Lezione 11

Array e stringhe

Immaginiamo di costruire un programma che permetta di visualizzare il listino prezzi di un bar. L'obiettivo è di fornire una tabella del seguente formato:

Prezzi delle consumazioni	
caffè	500
cioccolata	1000
cappuccino	800
chinotto	700
•	•
•	•

Il programma inizia richiedendo quante sono le consumazioni previste, quindi ne chiede il nome e il prezzo. Tali informazioni sono via via inserite dal programma in due tabelle interne, che saranno in seguito usate per visualizzare il listino completo. Costruiamo lo schema del programma, usando come riferimento il PASCAL:

- Richiede numero consumazioni previste
- Richiede allocazione di due array
- FOR I:=1 TO numero consumazioni DO
 - BEGIN
 - Chiede consumazione e la inserisce nella tabella dei nomi in posizione I
 - Chiede prezzo e lo inserisce nella relativa tabella in posizione I
 - END
- FOR I:=1 TO numero consumazioni DO
 - visualizza consumazione e relativo prezzo

Il programma BASIC risulterà come segue:

```

10 INPUT "Quante consumazioni";N
20 DIM C$(N),P(N)
25 REM Costruisce array consumazioni e prezzi
30 FOR I=1 TO N
40 INPUT "Consumazione e prezzo";C$(I),P(I)
50 NEXT I
60 REM Visualizza listino
70 CLS
80 PRINT "      *** LISTINO PREZZI ***"
90 PRINT
100 FOR I=1 TO N
110 PRINT C$(I),P(I)
120 NEXT I

```

Si osservi l'istruzione 20, che richiede l'allocazione di due array, uno per i prezzi destinato a contenere quindi informazioni numeriche e l'altro destinato a contenere

i nomi delle consumazioni. Si osservi in particolare che i due array sono distinti dall'interprete BASIC grazie alle caratteristiche del loro nome: il nome dell'array destinato a contenere i nomi, infatti termina con un "\$". In realtà, per l'elaboratore, i nomi delle consumazioni non sono altro che sequenze di caratteri: parleremo di "stringhe di caratteri" per indicare dati che, indipendentemente dal loro significato, sono costituiti da sequenze di caratteri. Tali sono per esempio:

caffè
AAAA

questa è una stringa di caratteri
ASD23

ma anche

123 12+7

possono essere stringhe di caratteri, qualora le si elabori come informazioni e non come valori numerici su cui si effettuino operazioni aritmetiche. Abbiamo così individuato due tipi fondamentali di variabili: quelle numeriche e le variabili stringa. Già abbiamo usato variabili numeriche: esse sono caratterizzate da identificatori quali

A,B...Z,A1,B1...B9,...

Le variabili stringa assumono invece identificatori quali:

A\$,B\$...Z\$

Si noti che, in espressioni di assegnamento, i caratteri assegnati alla variabile stringa devono essere compresi tra doppi apici, come nell'esempio seguente:

```
10 LET A$="questa e' una stringa"
```

Le stringhe non hanno una dimensione prefissata e possono pertanto contenere tanti caratteri quanti ne vengono di volta in volta assegnati. Pertanto tutti gli assegnamenti seguenti sono corretti:

```
10 LET A$="a"  
20 LET A$="questa e' una stringa"  
30 LET A$="questa e' una stringa piu' lunga"
```

Di volta in volta, l'interprete BASIC occuperà uno spazio di memoria differente a seconda del numero di caratteri che vengono assegnati. Non esistono limiti al numero di caratteri a parte la quantità di memoria disponibile. Qualora un'istruzione di assegnamento richiedesse uno spazio di memoria maggiore di quello disponibile, l'elaboratore fornirà un'indicazione d'errore del tipo OM.

Come reperire informazioni in un array: ricerca sequenziale

Tornando all'esempio del programma che costruisce il listino prezzi, immaginiamo di aggiungervi una funzionalità di ricerca, che consenta all'utente di ottenere, per esempio, il prezzo di una specifica consumazione.

Lo schema che realizza tale funzionalità è il seguente:

- Chiede la consumazione di cui si vuole il prezzo
 - Pone l'indice I uguale a 1
 - REPEAT
 - IF l'elemento I dell'array è uguale alla consumazione richiesta THEN
 - Visualizza il prezzo corrispondente
 - Ricorda che l'elemento è stato trovato
 - Incrementa l'indice I
- UNTIL I > N OR l'elemento è stato trovato
- IF l'elemento non è stato trovato THEN
- Visualizza messaggio

Le istruzioni sopra riportate realizzano un'iterazione del tipo REPEAT...UNTIL, poiché non è noto a priori il numero di iterazioni necessarie, che dipendono dalla presenza e dalla posizione della consumazione richiesta. Il programma risulterà:

```

125 REM Indica prezzo consumazione richiesta
127 CLS
130 PRINT "      *** INFORMATORE PREZZI ***"
132 PRINT
140 INPUT "Consumazione";A$
150 REM Inizio ricerca
155 LET F$="Non trovato"
160 LET I=1
170 REM Inizio iterazione
180 IF C$(I) <> A$ THEN 210
190 PRINT "Prezzo: ";P(I)
200 LET F$="Trovato"
205 GOTO 220
210 LET I=I+1
220 IF F$="Non trovato" AND I<=N THEN 170
230 IF F$="Non trovato" THEN PRINT "Consumazio
ne ";A$;" assente"
240 REM Fine programma

```

Si noti che è stata usata la variabile F\$ per ricordarsi che la consumazione richiesta è stata trovata e quindi l'iterazione deve essere interrotta. La struttura usata è la migliore dal punto di vista della coerenza con i principi della programmazione strutturata, ma la presenza di F\$ come indicatore di evento può disturbare il "purista" della programmazione. Osserviamo un programma alternativo (non si riportano le istruzioni da 100 a 140 che rimangono invariate):

```

150 FOR I=1 TO N
160 IF C$(I)=A$ THEN GOTO 200
170 NEXT I
180 PRINT "Consumazione ";A$;" assente"
190 GOTO 210
200 PRINT "Prezzo ";P(I)
210 REM Fine programma

```

In questo caso, il programma è molto più semplice; attenzione però che:

- si usa un FOR, ma il numero di iterazioni non è noto a priori
- la struttura risultante ha un solo punto d'ingresso (il FOR), ma due di uscita (il salto all'istruzione 200 e l'uscita regolare all'istruzione 170).

Quindi non sono rispettate le regole della programmazione strutturata: tali soluzioni sono adottabili localmente in un programma perché si sia consci di ciò che si sta facendo. Si noti che la ricerca nell'array è stata effettuata scandendo tutti gli elementi partendo dal primo: è stato cioè usato un algoritmo di ricerca sequenziale.

In questo caso se le consumazioni sono richieste con la stessa frequenza, c'è la stessa probabilità di trovare la consumazione richiesta all'inizio o alla fine della tabella: mediamente, ne scandiremo N/2 elementi. Conoscendo invece la frequenza delle richieste, converrà inserire all'inizio della tabella i prodotti più richiesti, per risparmiare tempo nella scansione degli elementi. Vedremo nel seguito altri algoritmi di ricerca.

Possiamo infine completare la suddetta funzionalità di ricerca in modo che siano possibili più ricerche in ciclo fino a quando l'utente non digita un comando di fine. Il programma così completato risulterà come segue:

```
100 REM Indica prezzo consumazione richiesta
110 CLS
120 PRINT "      *** INFORMATORE PREZZI ***"
130 PRINT
140 INPUT "Consumazione";A$
150 FOR I=1 TO N
160 IF C$(I)=A$ THEN GOTO 200
170 NEXT I
180 PRINT "Consumazione ";A$;" assente"
190 GOTO 210
200 PRINT "Prezzo ";P(I)
210 REM Fine programma
200 LET F$="Trovato"
205 GOTO 220
210 LET I=I+1
220 IF F$="Non trovato" AND I<=N THEN 170
230 IF F$="Non trovato" THEN PRINT "Consumazio
ne ";A$;" assente."
240 GOTO 140
250 REM Fine programma
```

Cosa abbiamo imparato

In questa lezione abbiamo visto:

- Il concetto di stringa
- L'uso di variabili stringa in BASIC
- L'algoritmo di ricerca sequenziale in un array

LE TECNOLOGIE

Esaminiamo i diversi tipi di macchine che possono essere utilizzati nelle varie applicazioni del C.A.I.

Dopo aver illustrato il valore formativo dell'informatica e le applicazioni alla didattica nei due aspetti di C.A.I. (Computer Aided Instruction) al docente e C.A.I. al discente, ci sembra opportuno dare uno sguardo complessivo al futuro sviluppo del nostro discorso.

Per prima cosa presenteremo quelli che possiamo chiamare gli strumenti di lavoro e i metodi impiegati. Più esattamente illustriamo qui in modo sintetico gli aspetti tecnologici delle applicazioni dell'informatica alla didattica.

In seguito dedicheremo un certo spazio alle differenti metodologie utilizzate in queste applicazioni ed ai problemi che le riguardano.

Esamineremo quindi l'impiego dell'informatica nelle diverse discipline, citando esperienze precise con i risultati che han-

no raggiunto e le relative valutazioni.

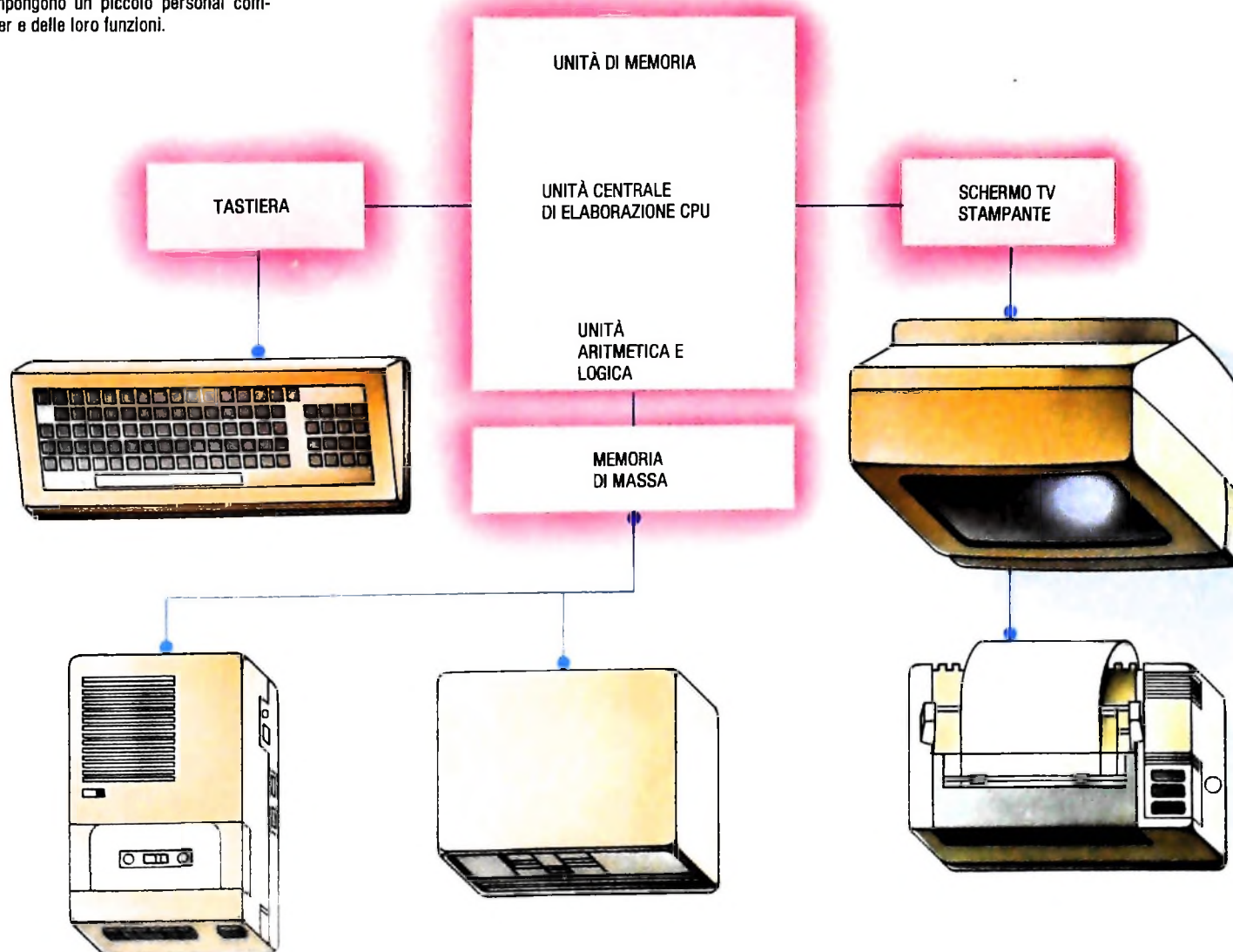
Concluderemo infine con l'analisi del ruolo fondamentale che può avere l'informatica nel problema della valutazione e più in generale della gestione della classe.

Vediamo anzitutto gli aspetti tecnologici.

Pur senza voler fare la storia degli elaboratori, ricordiamo che le prime macchine impiegate nel C.A.I. erano piuttosto grosse, dotate di grandi capacità di memoria e di calcolo. Ad esse si potevano collegare parecchi terminali indipendenti, anche distanti, attraverso linee telefoniche.

Poi sono arrivati i "minielaboratori", più piccoli come capacità di memoria, ma altrettanto potenti come capacità di calcolo. Anche ai "minielaboratori" possono essere collegati parecchi terminali.

Schema di massima delle parti che compongono un piccolo personal computer e delle loro funzioni.



Il ruolo del personal

Oggi tuttavia le prospettive di gran lunga migliori nella didattica sono offerte dai "microelaboratori" o "personal".

Si tratta di strumenti piuttosto economici e dalle prestazioni notevoli. I vantaggi che offrono sono diversi: anzitutto il costo, dell'ordine dei milioni di lire: da 2-3 a 8-10 a seconda della potenza e soprattutto del tipo di periferiche collegate. Il costo ridotto equivale alla possibilità di acquistare numerose macchine dello stesso tipo, ottenendo quindi una maggiore disponibilità di servizi, con intercambiabilità dei relativi programmi.

L'impiego di più macchine in uno stesso ambiente è utile anche in caso di guasto di una di esse, infatti il lavoro di una può essere tranquillamente svolto da un'altra senza particolari problemi.

Un altro vantaggio è la facile sistemazione: il personal ha un ingombro molto ridotto, è facilmente trasportabile e non richiede di essere collocato in un ambiente speciale (aria condizionata o altro).

Il personal infine ha il vantaggio di un più semplice accesso alla macchina: in parecchi casi è sufficiente accenderlo, in altri occorre solo inserire un dischetto.

Lo schema di una configurazione minima di un personal è abbastanza semplice e comprende cinque unità, spesso racchiuse in un unico contenitore.

All'unità centrale (o CPU, cioè Central Processing Unity) che controlla il sistema e interpreta le istruzioni del programma sono strettamente collegate l'unità di calcolo (aritmetica e logica) e l'unità di memoria. La prima, come dice il nome, viene impiegata nei calcoli di qualunque tipo, che in realtà si traducono, per quanto riguarda la macchina, in lunghe sequenze di calcoli molto semplici, utilizzati poi in modo diverso dai vari programmi. La seconda invece è usata per immagazzinare i programmi e i dati.

Esistono poi le unità di ingresso e di uscita. Le prime sono la tastiera, molto simile a quella delle macchine per scrivere, con eventualmente un tastierino numerico a lato e l'unità di memoria di massa a nastro magnetico o a disco magnetico, per memorizzare in modo permanente dati e programmi.

Le unità d'uscita sono ancora l'unità di memoria di massa e inoltre il video, sostituibile nei tipi più economici con un comune televisore, e la stampante.

Una caratteristica estremamente notevole è quella costituita dalle proprietà grafiche del video, cioè dalla possibilità di visualizzare non solo testi alfabetici e numerici, ma linee che rappresentano grafici di tipo scientifico o disegni di ogni genere. Senza le proprietà grafiche le applicazioni alla didattica, soprattutto per quanto riguarda le materie scientifiche, sarebbero molto limitate.

Quanto alle stampanti, ne esistono di tutti i tipi, con prezzi e prestazioni diverse. Molto utile per la didattica è il tipo con prestazioni grafiche, specialmente quando si può eseguire quella che si chiama solitamente una *hard copy* del video, cioè la riproduzione sulla stampante di tutto quello che in un dato momento compare sul video.

Una nuova proposta

Per il C.A.I. al discente è stata recentemente sperimentata una soluzione assolutamente originale e veramente innovatrice.

Si tratta di un sistema costituito da una unità centrale (personal o mini) a cui si accede tramite una tastiera pilota (per il docente) e più tastiere (fino a 10 o 15, una per ogni banco) per gli studenti.

La tastiera pilota ha il controllo anche sulle altre, nel senso che può attivare a piacere questa o quella tastiera (una per volta, evidentemente!) in modo da coinvolgere i singoli studenti nel processo di impostazione dell'algoritmo o di controllo e di modifica del programma.

L'uscita avviene su video a grande schermo, in modo che gli studenti possano interagire con il sistema stando seduti ciascuno al proprio posto.

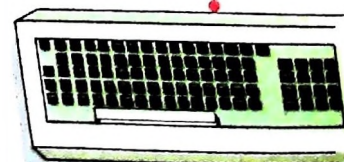
Il docente inoltre ha un piccolo monitor per seguire lo svolgimento di tutto il lavoro, dal momento che il grande schermo si trova alle sue spalle.

È evidente il rinnovamento radicale della didattica con questo sistema: anzitutto l'uscita su grande schermo evita l'accalcarsi degli studenti davanti al monitor del calcolatore. Ciascun studente può prendere appunti, riportare risultati, fare osservazioni stando al proprio posto e seguendo perfettamente quanto si sta facendo con l'elaboratore.

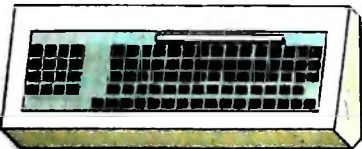
Inoltre ognuno, sotto la guida dell'insegnante, può diventare effettivamente parte attiva nel lavoro che si sta svolgendo in quel momento.

La vera novità di questo sistema, tuttora in fase sperimentale, è data dal rapporto diverso tra insegnante e allievi, un rapporto costruttivo e impegnativo per tutti, un rapporto che incrementa sensibilmente la collaborazione allievo-insegnante e allievo-allievo volta al raggiungimento di un dato obiettivo.

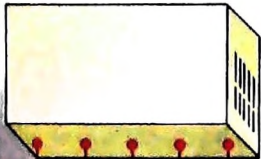
tastiere secondarie



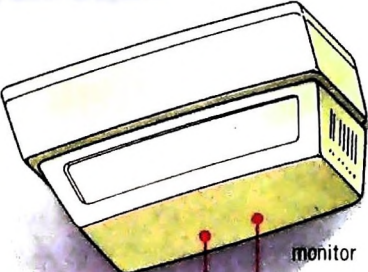
schermo



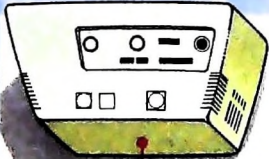
tastiera master



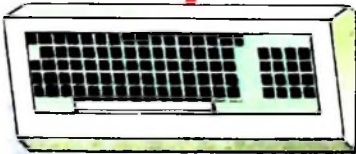
microcalcolatore



monitor



proiettore video a grande schermo



Glossario

Menù - in un programma, la visualizzazione sullo schermo di una serie di possibilità di scelta per l'utente. È uno strumento di semplificazione dell'interazione fra utente e calcolatore: l'utente non deve scrivere istruzioni complesse per indicare al programma che cosa desidera, ma è il programma stesso che offre un elenco esaustivo di tutte le possibilità, per la situazione data; l'utente effettua la sua scelta premendo un solo tasto (per esempio un numero o una lettera) corrispondente alla possibilità desiderata.

Minifloppy - un floppy disk del diametro di 5 pollici e 1/4.

Output - uscita.

Output, dispositivo di - un dispositivo che consente l'uscita di informazioni dall'unità di elaborazione di un calcolatore, per il loro trasferimento ad altre apparecchiature o per la comunicazione all'utente. Tipici dispositivi di output sono il video, la stampante, il plotter.

PILOT - un linguaggio di programmazione di alto livello, di facile apprendimento, pensato per utenti non esperti, in particolare nell'ambito delle applicazioni scolastiche o attinenti all'istruzione in generale.

Pixel - abbreviazione di "picture cell", indica l'elemento minimo di

un'immagine, in particolare su schermo. Le dimensioni del pixel dipendono dalle capacità grafiche della macchina e dalla capacità di risoluzione del video utilizzato.

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

Silicio - elemento chimico non metallico: è l'elemento più abbondante sulla Terra, dopo l'ossigeno. Viene utilizzato per la realizzazione di componenti elettronici, celle solari ecc.

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

Il plotter, la tavoletta grafica, il lettore ottico

Questo è il personal nella sua versione essenziale. Ad esso possono essere aggiunte anche altre unità periferiche più sofisticate. Tra queste ricordiamo innanzitutto il *plotter*, uno strumento per tracciare i grafici su carta, con penne a più colori. Si tratta evidentemente di un passo avanti rispetto alla stampante con proprietà grafiche sia per la presenza del colore, sia per la precisione del tratto nel disegno.

Un altro sussidio utile in questo senso è la *tavoletta grafica*, cioè superficie sulla quale si possono tracciare linee a mano libera (cerchi, ellissi, quadrati, rettangoli o figure diverse, sfondi vari, colori ecc.), che la tavoletta trasferisce nella memoria e sul video del calcolatore. Sembra comunque offrire maggiori vantaggi per certi scopi particolari piuttosto che per la didattica delle materie ordinarie.

Infine esiste il *lettore ottico*, uno strumento utilissimo per automatizzare l'inserimento nel calcolatore di notevoli masse di dati: ad esempio registrazione dei risultati di esperienze in laboratorio, statistiche, valutazione di grandi quantità di esercizi, test o problemi.

Modi e metodi d'impiego

Per quanto riguarda l'impiego di questo materiale dobbiamo riproporre la distinzione tra il C.A.I. al docente e il C.A.I. al discente. È chiaro che se in una scuola esiste soltanto un elaboratore le scelte non sono molte: l'insegnante lo può utilizzare in classe come supporto della sua lezione, ma gli allievi non possono vedere agevolmente le uscite video stando al proprio posto, a meno di non disporre di apparecchiature

per la proiezione su grande schermo. Quando poi sono gli allievi stessi a lavorare sulla macchina ciò richiede parecchi turni e inevitabilmente tempi d'accesso limitati.

Avendo invece maggiori disponibilità l'ideale per il C.A.I. al discente sarebbe un'aula appositamente attrezzata con 10 o 15 personal, e uno-due allievi per ogni posto.

In questo caso ogni allievo avrebbe un accesso diretto all'elaboratore, senza limiti di tempo dovuti a turni ed evidentemente l'esperienza sarebbe molto più ricca e costruttiva.

Non dimentichiamo che il valore dell'informatica dipende essenzialmente dalla risposta che la macchina fornisce all'uomo e ai suoi messaggi. L'allievo deve premere sui tasti, introdurre il programma che ha preparato, controllare se l'algoritmo scelto funziona oppure no, se il programma è stato scritto in modo corretto.

Questo per l'insegnamento dell'informatica; a maggior ragione l'accesso diretto alla macchina è essenziale nell'uso dell'informatica come strumento per l'insegnamento di altre discipline. Anche in questo caso ogni allievo, o al massimo ogni coppia di allievi, deve poter avere accesso alla macchina per verificare il proprio progresso, i propri dubbi, i propri errori. E ciò senza pressioni per dare spazio agli altri allievi.

Recentemente è stata annunciata la proposta di aule appositamente attrezzate per l'informatica: contengono un certo numero di elaboratori del tipo personal collegati con una unità centrale. Il docente all'unità centrale può controllare il lavoro dei singoli allievi, ciascuno dei quali opera su un personal periferico.

Si tratta di una struttura concettualmente simile al laboratorio linguistico, ovviamente adattato alla presenza degli elaboratori, in cui l'intervento dell'insegnante sarebbe molto più frequente ed efficace.

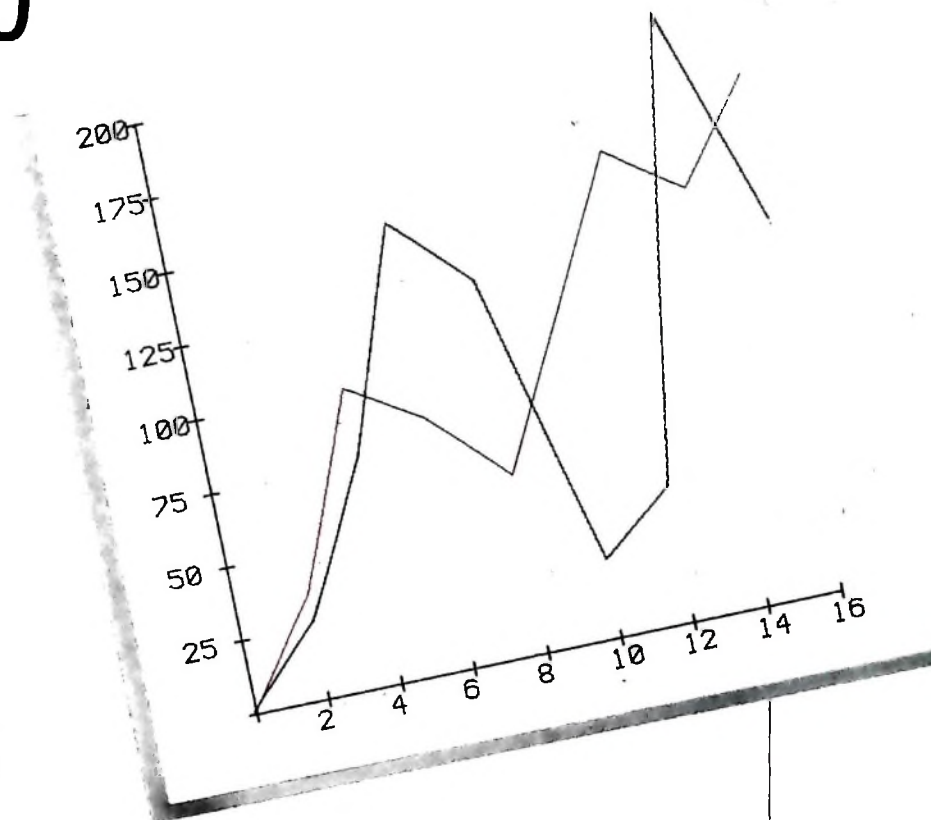
SFRUTTARE AL MEGLIO PL10

Utilizzando tecniche più raffinate è possibile realizzare in un array un grafico con i parametri.

Vedremo ora come utilizzare in maniera più sofisticata il microplotter PL10, utilizzando come esempio il programma che disegna due grafici visto nella lezione precedente.

Avevamo visto infatti che questa periferica, piuttosto versatile, può essere utilizzata contemporaneamente per scrivere testi e tracciare disegni all'interno dello stesso programma. Questo avviene, lo ricordiamo, inviando a PL10 i particolari comandi descritti nella scorsa lezione sotto forma di stringhe di caratteri.

Ogni comando, naturalmente, ha una potenzialità notevole che è possibile sfruttare utilizzandolo in modo particolare, ed è proprio questo che vedremo nelle prossime pagine.



```

10 INPUT "Valore di incremento per le gr
aduaZIONI di x e di y";N,M
15 LPRINT CHR$(18)
20 DIM X(9),Y(9)
22 X(1)=0
24 FOR I=2 TO 9
26 X(I)=X(I-1)+50
28 NEXT I
30 LPRINT "M50,0"
40 LPRINT "I"
50 REM Disegna l'asse delle x.
60 LPRINT "X1,50,8"
70 LPRINT "H"
80 FOR Z=1 TO 8
90 W=5*Z
100 LPRINT "M"+STR$(W*10-21)+"", "+STR$(-2
1)
110 LPRINT "P"+STR$(N*Z)
120 NEXT Z
130 LPRINT "H"
140 REM Disegna l'asse delle y.
150 LPRINT "X0,50,8"
160 LPRINT "H"
170 LPRINT "M-50,0"

```

Se diamo in input i valori 2 per le x e 25 per le y, otterremo come output del programma il grafico riprodotto nell'illustrazione. Se ora volessimo variare i dati e la scala del nostro grafico, ci basterebbe modificare i valori delle istruzioni 300 e 400 e, per la scelta della scala sui due assi, dare in input i valori appropriati di N e di M.

```

180 LPRINT "I"
190 FOR Z=1 TO 8
200 W=5*Z
210 LPRINT "M"+STR$(0)+"", "+STR$(W*10-10)
220 LPRINT "P"+STR$(M*Z)
230 NEXT Z
240 LPRINT "H"
250 LPRINT "M50,0"
260 LPRINT "I"
270 REM Cambia colore
280 LPRINT "C2"
290 REM disegna un grafico
300 DATA 0,50,150,300,250,50,90,400,250
310 READ Y(1),Y(2),Y(3),Y(4),Y(5),Y(6),Y
(7),Y(8),Y(9)
320 FOR I=1 TO 9

```

Verso la stampa di qualità

Negli ultimi anni si sono venute gradualmente raffinando le capacità di stampa di dispositivi dal costo medio o basso. Essi si basano fondamentalmente sulla rappresentazione di ogni carattere mediante una matrice di punti, ad ognuno dei quali viene associata l'informazione elementare 0-1, punto bianco - punto nero.

Molto diversificate sono invece le tecnologie applicate a questa comune filosofia: dalle tradizionali stampanti ad impatto (testa ad aghi), a quelle elettrostatiche, alle fotocompositrici, alle più recenti stampanti laser o a spruzzo d'inchiostro (inkjet).

Parametri fondamentali che contraddistinguono queste classi sono il costo e la cosiddetta risoluzione grafica, consistente nella minima distanza cui il dispositivo è in grado di riprodurre due punti; solitamente la risoluzione è espressa in punti per pollice (nel gergo tecnico si usa il termine inglese dot-per-inch, DPI).

Le stampanti ad impatto presentano risoluzioni comprese fra 50 e 150 DPI e costi del tipo 1-5 K\$; per le elettrostatiche (printer/plotter) si passa invece al 200-400 DPI e 10-20 K\$, ed infine a circa 1000 DPI per le fotocompositrici, con costi però ben al di sopra dei 100 K\$.

Chiaramente i modelli più costosi offrono velocità di produzione dell'ordine

delle decine o centinaia di pagine al minuto, ovvero quanto serve a soddisfare le esigenze di un centro tipografico. Al di sopra dei 200 DPI si ottengono comunque stampe la cui qualità dipende solo da quelle delle cosiddette fonti.

Esse sono un insieme di caratteri di stile omogeneo, presenti in firmware nella stampante o in software sul calcolatore. Anche nella fascia inferiore, dai 100 ai 200 DPI, è proprio grazie all'uso delle fonti, a volte persino ridefinibili dall'utente stesso, che è possibile realizzare in tempi ragionevoli stampe contenenti, oltre ai soliti caratteri alfanumerici, anche tutti quei simbolismi tecnici, scientifici o particolari che spesso ci sono necessari.

Attualmente le stampanti a matrice di punti stanno gradualmente sostituendo quelle a "margherita", che sono basate sulla stessa tecnologia costruttiva delle macchine per scrivere e quindi in grado di fornire risultati ad esse similari.

Infine, tra le tecnologie innovative, è il caso di ricordare che le stampanti laser, con risoluzione attorno ai 300 DPI e costi sul 20 K\$, rappresentano grazie a una buona velocità di produzione e all'uso di carta in foglio singolo, la probabile giusta via per soddisfare le richieste di un tipico ufficio di medie dimensioni.

ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz
 ABCDEFGH abcdefg 123 %&
 ABCDEFGH abcdefg 123 %&
 ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ
 αβγδεζηθικλμνξοπρστυφχψω
 $\int f(x)dx = \lim_{\Delta x \rightarrow 0} \left\{ \sum f(x) \Delta x \right\}$
 $\sin^2[\sqrt[3]{(4x^2)}] - \log_{10}(2y^2) + 4\theta^2 = 189.9 + 4\beta^2 - \delta^{10}$





ARCHIVIO EDDIS



Gli esempi di vario tipo, che riproducono abbastanza bene diversi caratteri usati in tipografia e che riescono a realizzare una espressione matematica complessa, con lettere greche, esponenti e simboli deponenti sono stati realizzati con una stampante a matrice di punti moderna, sofisticata e "intelligente". Le illustrazioni a colori sono invece eseguite con una stampante a getto d'inchiostro. La figura in alto è una elaborazione al calcolatore di una immagine ottenuta con una telecamera.

```

330 A$="D"+STR$(X(I))+", "+STR$(Y(I))
340 LPRINT A$
350 NEXT I
360 LPRINT "H"
370 LPRINT "C3"
380 REM Disegna un'altro grafico usando
390 REM il comando J
400 DATA 0,0,50,70,50,130,50,-30,50,-50,
50,100,50,100,50,-36,50,67
410 READ X(1),Y(1),X(2),Y(2),X(3),Y(3),X
(4),Y(4),X(5),Y(5),X(6),Y(6),X(7),Y(7),X
(8),Y(8),X(9),Y(9)
420 FOR K=1 TO 9
430 B$="J"+STR$(X(K))+", "+STR$(Y(K))
440 LPRINT B$
450 NEXT K
460 LPRINT "H"
470 END

```

Esaminiamo ora le varie fasi del programma in modo più dettagliato.

Commento al programma

10

Vengono richiesti gli incrementi da assegnare fra una graduazione e l'altra lungo gli assi di riferimento, ossia i valori di scala dei due assi.

20

Si definiscono due array, composti ciascuno da tanti elementi quanti sono gli incrementi (ovvero le tacche) presenti sugli omonimi assi, aumentati di uno. In questo caso gli elementi sono nove dato che gli incrementi sugli assi sono entrambi pari a 8 (vedi istr. 60 e istr. 150). L'aumento di uno è dovuto al fatto che si deve conteggiare anche l'origine.

22-28

Queste quattro istruzioni servono per assegnare agli elementi dell'array X, i valori corrispondenti alle ascisse dei punti graduati lungo l'asse delle x.

30-40

Si effettua un cambio di origine del sistema di riferimento. Questo per lasciare lo spazio necessario a segnare la scala a fianco dell'asse delle y.

80-120

Questo ciclo di FOR calcola i valori numerici da assegnare alle graduazioni della scala lungo l'asse delle x. Le istruzioni 100 e 110 rappresentano un nuovo modo di scrivere i comandi "Mx, y" e "Pstringa" visti in precedenza. Nella 100 viene calcolato il valore di x mediante un'espressione algebrica,

mentre quello di y è una costante; i valori -21 servono per posizionare correttamente sotto l'asse delle x le cifre calcolate. Nella 110 si calcolano e si stampano i valori di scala degli assi utilizzando il valore di input N. Si noti che mediante l'istruzione STR\$ e il simbolo + si possono fornire al plotter tutti i comandi. Questo perché il PL10, in modalità grafica, accetta in input solo stringhe di caratteri, e l'istruzione STR\$ trasforma dei numeri in stringhe. Il simbolo + serve per unificare in un'unica stringa l'intera istruzione, mentre fra virgolette si mettono tutte quelle parti dell'istruzione stessa che costituiscono la struttura del comando, come "M" e la virgola "," nella 100.

170-180

Si effettua un nuovo cambio di origine del sistema di riferimento. La nuova origine si trova ora nella posizione all'estrema sinistra della striscia di carta, nella stessa posizione in cui viene definita all'accensione del plotter. Ciò per potersi spostare a sinistra dell'asse delle y per scrivere la graduazione dei valori sull'asse stesso.

190-230

Questo ciclo di FOR calcola i valori numerici da assegnare alle graduazioni lungo l'asse delle y. Valgono osservazioni analoghe a quelle fatte per le istruzioni 80-120 a proposito di comandi "M" + STR\$ e "P" + STR\$.

250-260

Si effettua un nuovo cambio di origine del sistema di riferimento, che viene a coincidere così con l'origine degli assi.

300-310

L'istruzione DATA serve per memorizzare una serie di dati da utilizzare nel corso del programma. Essi sono accessibili per l'istruzione READ, la quale associa ad ogni variabile il corrispondente valore che legge in DATA. Per esempio, alla variabile Y (6) viene assegnato il valore (50). Le variabili presenti nella READ sono gli elementi dell'array Y definito nell'istruzione 20. I valori contenuti in DATA si riferiscono alle ordinate dei punti che costituiranno il grafico da disegnare. Ossia, i punti di coordinate (X (I), Y (I)), con I che può assumere valori da 1 a 9, sono gli estremi dei segmenti che definiranno il grafico.

320-350

Questo ciclo di FOR esegue il disegno del grafico utilizzando i dati memorizzati negli elementi degli array X e Y. La 330 definisce una stringa, che per esempio quando I=2 vale "D50,50". La 340 è l'esecuzione del comando Draw (Disegna) visto in precedenza.

400-450

Queste istruzioni servono a far rilevare come sia necessario fornire in DATA anche i valori per gli elementi dell'array X, qualora si voglia far uso del comando di "disegno relativo J". Si possono applicare commenti analoghi a quelli forniti per le istruzioni 300-350.

Banca Aperta

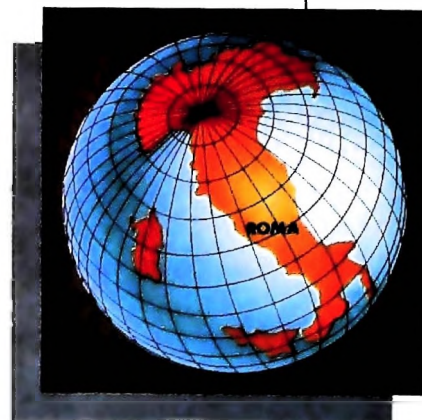
LE NUOVE RISPOSTE DEL BANCO DI ROMA.



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

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

nuovo modo di essere banca, un rapporto più personalizzato, un clima più agevole, più professionale e una maggiore rapidità in ogni operazione. Un ulteriore passo avanti verso la completa consulenza finanziaria che il Banco di Roma intende mettere a disposizione dei propri clienti. Tra i numerosi servizi offerti ricordiamo: Prestito Personale, Prestito Casa, gestione dei patrimoni, Leasing, assistenza all'import-export, attraverso ben 60 sedi estere in 30 Paesi dei 5 continenti. Tutto questo perché il Gruppo Banco di Roma è in grado di gestire ogni servizio specifico con grande professionalità, fornendo anche informazioni dirette a domicilio attraverso i sistemi Videotel e Voxintesi.



 **BANCO DI ROMA**
CONOSCIAMOCI MEGLIO.

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

Per informazioni rivolgersi ai negozi Olivetti o Olivetti M10 Punto di Vendita
 Olivetti, Olivetti Olivetti, Divisione Personal Computer, Via Meravigli 12,
 20123 Milano
 NOME COGNOME
 VIA
 CAPOLTA
 TELEFONO