

4

# CORSO PRATICO COL COMPUTER

è una iniziativa **FABBRI EDITORI**

in collaborazione con **BANCO DI ROMA**

e **OLIVETTI**

420075

F4 F5 F6 F7 F8

BATTERY LOW

diretto da **GIANNI DEGLI ANTONI**

BREAK

F8

PASTE

LABEL

PRINT

PAUSE

**VINCI IL COMPUTER:**

Estrazione di 10  
PERSONAL OLIVETTI M-10

**IL PRIMO COUPON  
PER LA SECONDA  
ESERCITAZIONE  
GRATUITA**

**IL TOTOCALCIO  
COL COMPUTER**

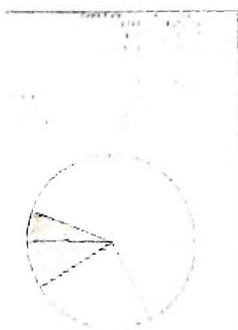
FABBRI EDITORI

Spazio in abbonamento postale G.P. n. 20 - 2000

# DITELO COI GRAFICI

Sharp vi offre quattro tipi di grafico, in quattro colori diversi. Basta sfiorare un tasto.

## SHARP GRAPH EL-7050



- 1 Immettete i dati come con una normale calcolatrice.
- 2 Sfiorate un tasto: ecco il risultato grafico.
- 3 Volete un grafico diverso, senza cambiare dati? Sfiorate un altro tasto. Potete usare la funzione continua per avere un grafico con moltissimi elementi.
- 4 Potete sovrapporre le spezzate per un confronto immediato.
- 5 Potete elencare i risultati in ordine decrescente.

Potenza dei grafici: fanno vivere i numeri dando forza alla loro fredda presenza. Sfruttate tutta la potenza dei grafici con Sharp EL-7050! Vi basterà sfiorare un tasto e questa piccola calcolatrice presenterà i suoi risultati in forma grafica. Grafici a bastoni, a spezzata, circolari, a bande. A quattro colori con o senza tratteggio. Tutto con la semplicità di una calcolatrice, senza dover imparare nessun linguaggio di computer. Grazie a Sharp e alla sua tecnologia potrete fare conoscere le vostre opinioni nel modo migliore: con un grafico.

L. 289.500 + IVA

### Altre caratteristiche

- Stampa dei dati immessi, calcolo delle medie e dei totali
- Funzione di scrolling, avanti e indietro
- Stampa di una tabella e calcolo di tutti i dati in quadratura
- Dati e risultati fino a otto cifre



concessionaria  
per l'Italia

MELCHIONI



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

Testi  
GOFFREDO HAUS, CLAUDIO PARPELLI, VIRGINIO SALA, 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

## CORSO DI PROGRAMMAZIONE E BASIC: LA CASSETTA

La cassetta contiene dieci programmi didattici, in parte da modificare o completare, come sussidio didattico al Corso di Programmazione e BASIC. I programmi riguardano vari tipi di problemi, dalla generazione di frasi a esempi di variazione di testi, a problemi geometrici e aritmetici, a giochi e altro ancora.

Per conoscere le modalità d'uso dei programmi, effettuate le operazioni, come indicato nella 1<sup>a</sup> di copertina del n. 3

A cura di ETNOTEAM S.p.A. - Milano  
(A. Bicego - M. Malocchi)

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

Aut. Min. Conc.

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 - Registrato presso il Tribunale di Milano - 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. 4 - esce il giovedì - Spedizione in abb. postale - Gruppo II/70. L'Editore si riserva la facoltà di modificare il prezzo nel corso della pubblicazione, se costretto da mutate condizioni di mercato.



Il programma ADDRSS che appare nel menù di M10, che fa parte cioè del gruppo di programmi forniti direttamente dal costruttore e residenti in modo permanente nella memoria della macchina, è, come si può sospettare dal nome (abbreviazione di "address", indirizzo), un programma che può gestire una rubrica telefonica. Presenta punti di contatto con il programma SCHEDL (che vedremo nel seguito), dal quale si differenzia per la possibilità di realizzare automaticamente

ADDRSS è un programma che consente la gestione di un indirizzario o di una rubrica telefonica, realizzata mediante il programma TEXT in un file chiamato ADRS.DO. Consente il recupero di un nominativo con il relativo indirizzo e numero telefonico, e permette la gestione anche di informazioni di altra natura.



chiamate telefoniche (con le opportune apparecchiature). ADDRSS è un programma per la *gestione* della rubrica telefonica, non però per la sua *realizzazione*: la preparazione della rubrica va fatta con il programma TEXT. Una volta preparata la rubrica, con il programma ADDRSS si possono effettuare tutte le procedure di ricerca desiderate.

### Rubrica telefonica

Accendete il computer. Come al solito, apparirà il menù: provate a posizionare il cursore sul nome del programma ADDRSS e premete poi il tasto ENTER, seguendo lo stesso procedimento usato per accedere al programma TEXT. Questa volta, però, le cose non vanno come forse vi aspettavate: M10 vi risponde con un segnale sonoro di avvertimento e sul

display appare, su due righe, la scritta: ADRS.DO not found Press space bar for MENU, che significa "non ho trovato il file ADRS.DO; premi la barra spazio per tornare al menù". Questo è quel che volevamo dire parlando di un programma che gestisce, ma non costruisce, la rubrica telefonica: ADDRSS richiede la presenza di un file ADRS.DO, che, come indica il suffisso, è un file di testo. ADRS.DO è quindi anche l'unico nome che si possa dare alla nostra rubrica telefonica, per poterla gestire con ADDRSS.

Il primo passo indispensabile, dunque, è creare il file con la rubrica: per questo torniamo al menù, scegliamo il programma TEXT, e alla richiesta della macchina, File to edit?, comuniciamo che il nome del file è ADRS. Premuto come al

solito ENTER, ci ritroviamo nella situazione di poter scrivere quello che vogliamo, come con una macchina per scrivere.

In effetti, il file ADRS non presenta caratteristiche peculiari: potremmo, volendo, riempirlo di un testo qualunque. Le informazioni che andiamo a inserirvi non debbono pertanto avere un formato standard prestabilito, ma possono essere organizzate a piacimento. Possiamo, per esempio, inserire un primo nome (usiamo dei dati fittizi): Bianchi Riccardo, viale Campania 177, Milano, tel. 434445. Questo è il "formato" che usiamo normalmente per inserire un nominativo nella nostra rubrica e nulla vieta di usarlo su M10.

C'è solo un vincolo da tenere presente: se si possiede il dispositivo per le chiamate automatiche, il formato per poter utilizzare tale dispositivo deve essere il seguente:

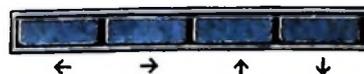
Nome: numero telefonico: indirizzo.

In particolare, il numero telefonico deve essere preceduto e

```
ADRS.DO not found
Press space bar for MENU
```

Se si accede al programma ADDRSS (con il normale procedimento da menù, mediante i tasti di posizionamento del cursore e il tasto ENTER) prima di aver creato il file ADRS.DO con le informazioni pertinenti, mediante il programma TEXT, sul display appare questa scritta,

con la quale il programma ci dice di non aver trovato il file ADRS.DO e ci invita a ritornare al menù principale. ADDRSS è un programma di gestione delle informazioni e non è utilizzabile per la immissione nella macchina delle informazioni stesse.



```
Colli Andrea:06=532223:Via Verdi
412,Roma^
Franchi Gilberto:666777:Via Caporetto
78,Milano^
```

Il programma ADDRSS non prevede che le informazioni nella rubrica telefonica siano organizzate secondo un formato particolare: possono essere immesse come si preferisce, in funzione della propria comodità e delle proprie esigenze. Esiste un vincolo solo se il modello

di M10 che si è acquistato dispone di modem integrato e della possibilità di chiamata telefonica automatica: in tal caso il numero telefonico deve essere preceduto e seguito da uno speciale carattere di identificazione (i due punti), come si vede nell'esempio qui sotto.



nome

:numero tel:

indirizzo

seguito da un segno di due punti (:) che funge da identificatore per la macchina. Se il numero è preceduto da un prefisso di teleselezione, è opportuno separarlo con un segno di uguale (=). Tale segno determina una pausa di due secondi nella composizione del numero, subito dopo il prefisso. (Due segni = determinano una pausa di quattro secondi ecc.)

Scegliere un formato standard è utile anche se non si possiede il dispositivo di chiamata automatica: si può scegliere quello come si preferisce, ma è bene tenere sempre quello, per facilitare le successive ricerche, per rendere più immediato l'inserimento di correzioni e per avere informazioni chiaramente comprensibili quando, in seguito a una operazione di ricerca, verranno visualizzate nuovamente sul display.

Supponiamo ora di adottare lo standard per il dispositivo di chiamata automatica, e inseriamo un gruppo di nominativi nel nostro file. Continuiamo nel nostro esempio (dove supponiamo di essere a Milano, e quindi di non dover memorizzare il prefisso di teleselezione per i numeri della città) e co-

minciamo a correggere il primo nominativo, il quale dovrà essere:

Bianchi Riccardo: 434445: viale Campania 177, Milano.

Possiamo o meno lasciare una riga di spazio e quindi:

Colli Andrea : 06=532223 : via Verdi 412, Roma

Franchi Gilberto : 666777 : via Caporetto 78, Milano

Marconi Francesca : 999887 : corso Spagna 100, Milano

Negri Bruno : 035=678901 : via Donizetti 13, Bergamo

Rossi Bianca : 222324 : piazza Monterosa 6, Milano

Verdi Massimo : 8784485 : viale XXV aprile 260, Milano.

Non è necessario inserire i nominativi in ordine alfabetico: quando si deve aggiungere un nuovo numero nella nostra rubrica telefonica, lo si può semplicemente aggiungere in fondo all'elenco, perché ai fini del programma l'ordinamento alfabetico è ininfluente.

Dopo aver riletto l'elenco e corretto gli eventuali errori di battitura, chiudiamo il file e ritorniamo al menù con l'usuale procedimento, premendo cioè F8.

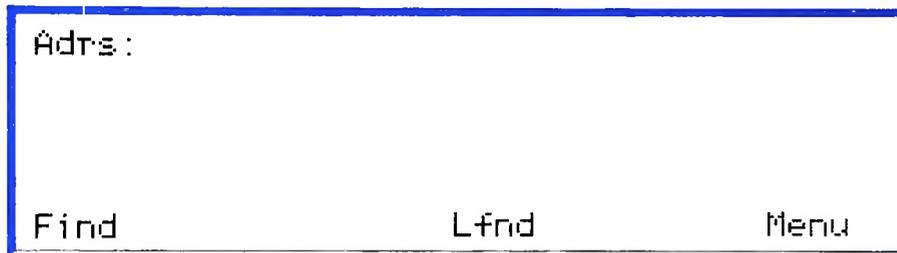
### Ricerca nella rubrica telefonica

Avendo preparato il file ADRS.DO, possiamo accedere al programma ADDRSS. Ci posizioniamo con il cursore, premiamo ENTER. Il menù scompare e ci appare sul display la visualizzazione standard di questo programma: in alto la scritta Adrs: seguita dal cursore lampeggiante; in basso il significato delle tre funzioni che risultano abilitate in questa

prima fase, e cioè F1 (Find), F5 (Lfd) e F8 (Menù).

Come potrete notare, chi ha preparato i programmi per M10 ha cercato di mantenere una certa coerenza complessiva: funzioni di natura analoga sono assegnate ai medesimi tasti in tutti i programmi (le funzioni Find e Menù erano attribuite rispettivamente ai tasti F1 e F8 anche nel programma TEXT). Questo semplifica notevolmente l'apprendimento e rende complessivamente più agevole l'utilizzazione della macchina: è un principio molto semplice, che val la pena di tenere presente anche nella stesura dei propri programmi.

Nel programma ADDRSS, dunque, non c'è bisogno di premere il tasto LABEL per visualizzare il significato dei tasti funzione abilitati: ci pensa il programma stesso, automaticamente, non appena caricato. Tuttavia, una volta sicuri dei significati, è possibile togliere dal display queste scritte premendo LABEL. Premendo una seconda volta LABEL, le scritte ricompaiono: il funzionamento è a questo punto del tutto identico al funzionamento in TEXT.



Ecco come appare il display se si seleziona il programma ADDRSS da menù (mediante cursore e tasto ENTER) dopo la creazione del file ADRS.DO. In alto compare il "pronto" del programma, mentre sull'ultima riga viene immediatamente visualizzato il significato dei tasti di funzione abilitati dal program-

ma, che sono F1, F5 e F8. F1 e F5 servono per la ricerca di informazioni: il primo visualizza le informazioni identificate sul display, il secondo presuppone il collegamento di una stampante e le visualizza su carta. F8, come al solito, consente di uscire dal programma e di tornare al menù.



F1

Se premiamo il tasto F1, per predisporci alla ricerca di qualche informazione nel file ADRS.DO, sul display compare la scritta Find subito accanto al "pronto" del programma, seguita dal cursore

lampeggiante. A questo punto possiamo digitare sulla tastiera la sequenza di caratteri che ci interessa ricercare: il nome di una persona, se vogliamo recuperare indirizzo e numero telefonico, il

nome di una città, se ci interessa verificare gli indirizzi di tutte le persone che abitano in quella città ecc. Non è importante, in questa situazione, la distinzione fra lettere maiuscole e minuscole.



Vogliamo dunque cercare un nominativo nella nostra rubrica. La funzione che ci serve è Find: premiamo il tasto F1. Sulla prima riga del display, accanto a Adrs: compare anche la scritta Find, quindi la macchina aspetta che digitiamo alla tastiera la sequenza di caratteri che vogliamo ricercare. Supponiamo di aver bisogno del numero telefonico della signora Francesca Marconi. Possiamo dunque digitare Marconi alla tastiera e il programma ci risponde facendo apparire sul display tutte le informazioni pertinenti, come le avevamo inserite, cioè: cognome e nome, numero telefonico, indirizzo. Il tempo di risposta del programma è minimo e resta trascurabile anche con una rubrica molto più ricca della nostra. Notate ancora una volta che nell'impostare la sequenza di

caratteri per la ricerca, la distinzione fra maiuscolo e minuscolo non è rilevante: la macchina recupera sempre le stesse informazioni anche se digitiamo marconi o MARCONI al posto di Marconi.

Notate poi che il programma, in risposta alla nostra richiesta di ricerca, ci fornisce tutto il blocco di testo, all'interno del quale compare la sequenza di caratteri fornita come chiave di ricerca, fino al punto che abbiamo delimitato con un ritorno a capo. Se, per esempio, avessimo memorizzato i nominativi della nostra rubrica in ADRS.DO in questo formato:

**Marconi Francesca : 999887 :  
corso Spagna 100, Milano**

## Il modem

L'illustrazione mostra M10 collegato a un modem/accoppiatore acustico esterno. Il modem (modulatore/demodulatore), abbinato all'accoppiatore acustico, consente da un lato la trasformazione dei segnali digitali (discreti, di tipo numerico) del calcolatore in segnali analogici (continui, di tipo acustico in questo caso) che possono viaggiare lungo le normali linee telefoniche e, dall'altro lato, la trasformazione inversa dei segnali acustici che viaggiano sulle linee telefoniche in segnali di-

gitali che possono essere accettati dal calcolatore. Il modem costituisce lo strumento fondamentale per poter mettere in comunicazione a distanza due calcolatori o, più in generale, due apparecchiature digitali per l'elaborazione di dati.



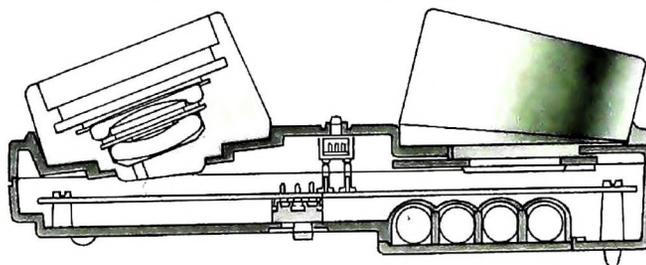
dando cioè un ritorno a capo anche dopo il numero telefonico, richiedendo la ricerca della sequenza di caratteri Marconi il programma ci fornirebbe in risposta solo la prima riga, **Marconi Francesca : 999887** : e non tutte le informazioni relative a quel nominativo. Il programma non ha alcun modo di stabilire che tutte quelle informazioni sono pertinenti alla signora Marconi: opera esclusivamente in termini di "record", singoli blocchi di testo delimitati da un ritorno a capo (tasto ENTER).

Questo fatto può anche essere sfruttato per nostra comodità, qualora volessimo arricchire la nostra rubrica, come vedremo fra poco, e non fare apparire automaticamente sul display tutte le informazioni.

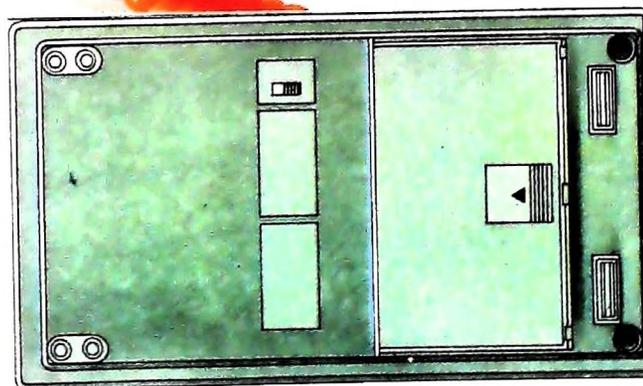
La ricerca non deve essere fatta necessariamente per cognomi: si può far ricercare al programma qualunque sequenza di caratteri. Se la sequenza non compare nel file ADRS.DO, il programma si limita a spostare il cursore sulla riga successiva del display, facendo riapparire la scritta Adrs:. Succedrebbe così, nel nostro esempio, qualora chiedessimo al programma di rintracciare la sequenza **largo**, per dirne solo una a caso.

Potremmo impostare (dopo aver premuto F1) la sequenza **Milano**: sul display apparirebbero tutti i nominativi nel cui "record" compare la parola **Milano**, cioè, nel nostro caso, gli indirizzi di tutte le persone il cui nome è compreso nella rubrica e che abitano a Milano.

vista in sezione dell'accoppiatore acustico Olivetti M10



Esiste un modello di M10 con modem già incorporato, che consente il collegamento diretto al telefono: gli esemplari di tipo europeo ne sono sprovvisti, mentre il dispositivo è standard sui modelli americani. In questo caso è possibile effettuare chiamate telefoniche automatiche.



faccia inferiore dell'accoppiatore acustico

## Quit e More

Se avete usato per prova la nostra rubrica fittizia, inserendo come sequenza di caratteri per la ricerca Milano, interviene un fatto nuovo: sul display compaiono tre record completi (quelli dei signori Bianchi, Franchi e Marconi), la settima riga rimane bianca e sull'ultima in basso le scritte relative ai tasti funzione F1, F5 e F8 scompaiono per lasciare il posto a due scritte relative ai tasti funzione F3 e F4: More e Quit, rispettivamente.

Questo significa che, nella sua ricerca, il programma ha identificato un numero di record pertinenti superiore a quello dei record visualizzabili sul display: pertanto in qualche modo ce lo deve far sapere e nel far questo ci offre subito la possibilità di procedere ulteriormente nella visualizzazione (con il tasto F3, More) o di uscire da questa situazione per tornare all'inizio del programma (con il tasto F4, Quit). Se premiamo F3, il programma effettua lo scrolling del testo, facendo scomparire i primi record e visualizzando gli ultimi due che soddisfano alla condizione da noi posta. A questo punto non

ci sono più record pertinenti, e le scritte Quit e More scompaiono per lasciare il posto nuovamente alle scritte relative a F1, F5 e F8.

Notate un piccolo particolare: compaiono solo i significati dei tasti F3 e F4, in questa situazione, quindi questi sono gli unici due tasti abilitati al momento. Se, ricordando qual era il tasto che riportava al menù iniziale, premete F8 per uscire non solo da questa situazione, ma dal programma ADDRSS in generale, non otterreste nessun risultato: F8 ora non è abilitato. Bisogna prima uscire da questa situazione, arrivando alla fine della visualizzazione dei record identificati con F3, oppure premendo F4; poi è possibile, con il tasto F8, tornare al menù.

Un discorso analogo vale per la funzione Find: in quella situazione non è attivata, e di conseguenza per impostare una nuova ricerca è necessario innanzitutto uscirne premendo il tasto F4.

Tornate alla posizione iniziale del programma, ma ora premete LABEL, per far scomparire le scritte relative ai tasti funzione. Rieffettuate la ricerca della sequenza Milano. Vi

```
Bianchi Riccardo:434445:Viale Campania
177,Milano^
Franchi Gilberto:666777:Via Caporetto
78,Milano^
More Quit
```



Se la richiesta di ricerca al programma è abbastanza generica (per esempio se impostiamo la sequenza di caratteri Milano, per avere tutti i nominativi nella

rubrica di persone che abitano a Milano), può darsi che il display non sia in grado di contenere tutti i record pertinenti. In questo caso non riappare il

"pronto" del programma, mentre sull'ultima riga del display vengono visualizzati i significati di due funzioni che il programma abilita in questa situazione:

F3 e F4, rispettivamente More e Quit, che consentono la visualizzazione di un ulteriore gruppo di record oppure di uscire dalla situazione.

```
Rossi Bianca:222324:Piazza Monterosa
6,Milano^
Verdi Massimo:8784485:Viale XXV Aprile
260,Milano^
Adrs:
Find Lfnd Menu
```



Per visualizzare i successivi record che rispondono al tipo di richiesta effettuata è necessario premere F3 se anche il successivo gruppo di righe non consente

di esaurire tutti i record pertinenti, restano attivati i tasti funzione F3 e F4. Quando la visualizzazione è completa, il programma riattiva automaticamente le

funzioni F1, F5 e F8. È possibile solo andare avanti nella visualizzazione, e non tornare indietro: per rivedere un record già passato è necessario ripetere l'operazione di ricerca". Si può eliminare

la visualizzazione delle funzioni attivate, premendo il tasto LABEL. Ripremendolo, i nomi delle funzioni riappaiono.

compaiono nuovamente i primi tre record, ma in basso non appare il significato di F3 e F4: il comando che avete dato premendo LABEL prima di impostare la ricerca continua a valere per la macchina. Tuttavia potete ugualmente capire subito che il programma non ha finito di visualizzare tutti i record che ha individuato, perché sull'ultima riga del display compare esclusivamente il cursore lampeggiante e non anche la scritta Adrs:.

### Sfogliare la rubrica

Una volta entrati nel programma ADDRSS, non è necessario effettuare una ricerca mediante una sequenza di caratteri: è possibile anche sfogliare la rubrica da capo a fondo, visualizzando tutti i record nello stesso ordine in cui sono stati immessi.

Per fare questo, è sufficiente premere F1 (Find) e quindi non digitare nulla, ma premere solo il tasto ENTER. Viene visualizzato l'inizio del file ADRS.DO e si attivano le funzioni

### La funzione Lfnd

In ADDRSS, il tasto F5 ha la funzione Lfnd, che è del tutto identica alla funzione Find, ma determina l'invio dei risultati della ricerca alla stampante anziché al display. Selezionando F5, sul display compare il "pronto" Lfnd; se per errore si imposta una ricerca con F5 senza avere una stampante collegata al calcolatore, il programma si blocca: il cursore scompare e non succede più nulla. Per uscire da questo blocco è necessario premere contemporaneamente i tasti SHIFT e BREAK PAUSE.

BREAK



PAUSE



```
Colli Andrea:06=532223:Via Verdi
412,Roma^
Verdi Massimo:8784485:Viale XXV Aprile
260,Milano^
More Quit
```



Nella preparazione del file ADRS.DO è bene prevedere le ambiguità possibili, ed eliminarle con qualche piccolo accorgimento. Se chiediamo, con F1, la ri-

cerca della sequenza di caratteri Verdi, nel nostro esempio, sul display compare l'indirizzo del signor Verdi e quello di una persona che abita in via Verdi. Per

evitare questo, si possono identificare i nominativi con qualche simbolo specifico, per esempio un +, che non si possa mai presentare in altra posizione.

More e Quit, cioè F3 e F4: premendo F3 si visualizzano alcuni record successivi, premendo F4 si esce dalla situazione. In questo modo, si può però solo sfogliare la rubrica: non è possibile effettuare inserimenti o correzioni. Per questo bisogna sempre tornare al programma TEXT e operare sul file ADRS.DO. Inoltre, la rubrica può essere sfogliata solo "in avanti", dall'inizio verso la fine del file. Se provate a spostare il cursore mentre siete in questa situazione, non succede nulla: il cursore non è attivo. Se per errore avete superato l'informazione che cercavate, dovete ricominciare a sfogliare la rubrica dall'inizio. Lo stesso vale anche per le ricerche che hanno come risultato un numero di record superiore a quello visualizzabile sul display: una volta proseguita la visualizzazione con F3 (More), non si può tornare indietro se non reimpostando la ricerca.

### Qualche accorgimento

Torniamo al nostro esempio fittizio e impostiamo la ricerca della sequenza di caratteri Verdi. La macchina visualizza:

Colli Andrea : 06=532223 : via Verdi 412, Roma

Verdi Massimo : 8784485 : viale XXV aprile 260, Milano.

Come già sottolineato, il programma prende la nostra sequenza di caratteri per quello che è: non sa se stavamo cercando il signor Verdi o una persona che abita in via Verdi, se non gli forniamo noi direttamente questa informazione. In una piccola rubrica come quella del nostro esempio, problemi di ambiguità come questo possono dare poco fastidio:

## Per cancellare un file dalla memoria di M10

```
OLIVETTI M10 BASIC 1.0
(C) 1983 Microsoft
17093 Bytes free
Ok
kill "PROVA.DO"
Ok
```

È necessario entrare in BASIC e impostare un'istruzione specifica, per l'esattezza: KILL "xxxx.yy", dove KILL ("uccidere" in inglese) è il comando BASIC che determina la cancellazione del file; al posto di xxxx va messo il nome del file; al posto di yy va messo il suffisso BA, CO o DO a seconda che il file contenga un programma in BASIC o in linguaggio macchina, o sia un file di testo. Se volessimo cancellare il nostro file PROVA (un file di testo), dovremmo impostare, selezionato il BASIC da menù, l'istruzione KILL PROVA.DO e farla seguire dal tasto ENTER.



F8



volevamo il numero di telefono del signor Verdi e ci è comparso sul display anche il numero del signor Colli, che abita in via Verdi. Poco male, ci raccapeziamo subito. Tuttavia, se la rubrica comprendesse un centinaio di nominativi o più, l'ambiguità potrebbe risolversi in un pasticcio, diminuendo di molto l'utilità e la comodità dello strumento elettronico rispetto a una comune agendina tascabile.

Con un minimo di ingegnosità, l'ambiguità può essere eliminata senza gravi problemi: basta per esempio escogitare dei codici che identifichino i nominativi rispetto agli indirizzi o ad altre informazioni contenute nel record. Il codice, per eliminare effettivamente l'ambiguità, deve essere un carattere o una sequenza di caratteri che non possano essere utilizzati altrimenti nel file, o tutta l'utilità svanirebbe. Così, per esempio, potremmo far precedere i cognomi da un segno +, da un < o da una sequenza come YXZ che non ha nessuna probabilità di trovare altra utilizzazione nel file. Così, i nostri record avrebbero un formato del tipo:

+Verdi Massimo : 8784485 : viale XXV aprile 260, Milano

e la sequenza di caratteri per la ricerca di un nominativo comincerebbe sempre con il segno +.

Come abbiamo detto, ADDRSS non serve necessariamente e solo per una rubrica telefonica: possiamo inserirvi anche informazioni di natura completamente diversa, per poi recuperarle con la funzione Find, per nostra comodità. Un aspetto interessante può essere già quello di arricchire la rubrica con informazioni "secondarie" relative ai nominativi inseriti: informazioni relative alla professione, agli interessi, o a qualunque altra caratteristica utile della persona in questione. Non dimenticate, inoltre, che anche il programma TEXT ha una funzione Find: nulla vieta di avere due file distinti, uno da gestire con ADDRSS e contenente le informazioni da rubrica telefonica e uno da elaborare normalmente con TEXT, contenente anche o solo le informazioni ulteriori.

Tenete presente che ADDRSS agisce solo sul file ADRS.DO: non potete avere due file distinti su cui ADDRSS agisca direttamente. Tuttavia, potete usare un piccolo trucco, se volete avere più di un file da gestire con ADDRSS, in modo da separare classi diverse di informazioni, secondo le necessità. Potete lasciare ADRS.DO vuoto, dopo averlo creato e creare invece tutta la serie di file di informazioni che vi servono, con altri nomi opportuni. Poi, al momento in cui vi serve, potete trasferire il contenuto di uno di questi file in ADRS.DO con le procedure Copy e Paste del programma TEXT, per poi operare con ADDRSS.

Finita l'utilizzazione, potete ripulire il file ADRS.DO con la procedura Cut del programma TEXT, in modo da lasciarlo vuoto per qualunque ulteriore operazione.

Il procedimento è un poco macchinoso, a prima vista, ma può avere qualche utilità se avete bisogno di lavorare in modo analogo con classi diverse (ma per altri aspetti simili) di informazioni che volete lasciare ben separate. Per la semplice ricerca, il programma TEXT è sufficiente e immediato; ma, se per caso voleste far stampare le informazioni che desiderate in maniera selettiva, procedere con ADDRSS è certo più rapido: ADDRSS opera per "record", cioè per blocchi di testo identificati dai "ritorno a capo" e quindi consente di mantenere come unità separate le informazioni attinenti a un nominativo piuttosto che a un altro, mentre TEXT considera tutto il testo memorizzato in un file alla stessa stregua.

In altre parole: se volete ricercare informazioni attinenti al signor Verdi, potete farlo anche con TEXT, usando la funzione F1: il cursore vi si posiziona vicino alla sequenza di caratteri Verdi e potete leggere quel che segue, eventualmente utilizzando il cursore per visualizzare parti ulteriori di testo. Se però volete isolare tutte le informazioni pertinenti a persone che vivono a Roma e sono comprese nella vostra rubrica, il programma ADDRSS è quello giusto.

Con TEXT potreste comunque raggiungere quelle stesse informazioni, tuttavia, con un maggior spreco di tempo.

## Sequenze d'esecuzione

Torniamo a esaminare algoritmi scritti in un linguaggio naturale e consideriamo il seguente, per effettuare una telefonata:

Algoritmo per effettuare una telefonata:

- Sollevare il ricevitore
- Comporre il numero
- Attendere la risposta
- Condurre la conversazione
- Deporre il ricevitore

Un tale algoritmo, opportunamente eseguito, determina una successione di operazioni da parte dell'esecutore, che è il reale obiettivo del nostro programma e che indicheremo con il nome di **SEQUENZA D'ESECUZIONE**.

Tutti i programmi fino ad ora esaminati inducevano una sequenza d'esecuzione.

Tuttavia, ciò è molto limitativo rispetto alla classe di problemi che vogliamo risolvere: infatti, osservando l'algoritmo per effettuare la telefonata, cosa succede se, dopo avere composto il numero, otteniamo il segnale di "occupato" o se il telefono non funziona o se la persona con cui vogliamo parlare non è presente?

In tutti questi casi, la sequenza d'esecuzione prevista dal programma non è più adeguata e deve essere differente; non solo, ma la corretta sequenza d'esecuzione non può essere decisa a priori, prima di eseguire l'algoritmo (sarebbe come disporre di un programma per fare le telefonate quando il destinatario è occupato, o per telefonare quando il telefono è guasto, e così via!), ma deve essere decisa proprio durante l'esecuzione! Ci troveremo quindi costretti ad ampliare la base delle istruzioni del nostro linguaggio di programmazione, con le **STRUTTURE DI CONTROLLO**.

## Strutture di controllo

Supponiamo semplicemente di alterare l'algoritmo precedente come segue:

Algoritmo per effettuare una telefonata:

- Alzare il ricevitore
- Comporre il numero
- SE il numero è "occupato" ALLORA
- Deporre il ricevitore

### ALTRIMENTI

- Attendere la risposta
- Condurre la conversazione
- Deporre il ricevitore

*In allegato a questo numero troverete una cassetta contenente testi e programmi per M10: si tratta di piccoli esempi didattici, da capire, da modificare o da correggere, relativi ai contenuti delle lezioni 11-25. Caricate il testo di nome ISTR: contiene le istruzioni di dettaglio.*

Questo algoritmo contiene un nuovo tipo di istruzione, che chiameremo **STRUTTURA DI CONTROLLO**, che non determina l'attuazione di operazioni da parte dell'esecutore, ma lo induce a prendere una decisione, sulla base della quale una tra due differenti sequenze d'esecuzione verrà selezionata. Si tratta del costrutto:

**SE....ALLORA....ALTRIMENTI.....**,

che fa scegliere la prima delle due alternative (quella che segue la parola-chiave **ALLORA**) o la seconda (quella che segue la parola-chiave **ALTRIMENTI**) secondo che la condizione espressa (dopo la parola-chiave **SE**) risulti verificata o meno. Infatti, se dopo avere composto il numero otteniamo il segnale di "occupato", la sequenza di esecuzione innescata risulta:

Sollevare il ricevitore  
Comporre il numero  
(occupato)  
Deporre il ricevitore

mentre se il numero risulta "libero", la sequenza innescata risulta:

Sollevare il ricevitore  
Comporre il numero  
(libero)  
Attendere la risposta  
Condurre la conversazione  
Deporre il ricevitore

Così, un solo programma permette di realizzare più di una sequenza di esecuzione.

### **Strutture di selezione**

In particolare, la struttura di controllo individuata permette di selezionare una tra due sequenze d'esecuzione, e per tale motivo viene detta **STRUTTURA DI SELEZIONE BINARIA**.

La maggior parte dei moderni linguaggi di programmazione mette a disposizione strutture di controllo di selezione; ad esempio, il linguaggio **PASCAL** (un moderno linguaggio di programmazione ideato da N. Wirth al Politecnico di Zurigo alla fine degli anni '60) permette di scrivere istruzioni del tipo:

```
IF a=b THEN
  write ("uguali")
ELSE
  write ("diversi")
```

che permette di visualizzare ("write" in inglese significa "scrivi") il messaggio "uguali" o "diversi" a seconda che il valore della variabile a sia o meno uguale a quello della variabile b.

Anche noi ci adegueremo alla scelta della nomenclatura del **PASCAL**, che è attual-

mente il linguaggio più usato come strumento di comunicazione sulle riviste di informatica (anzi, per le sue caratteristiche di "pulizia" concettuale, faremo sempre riferimento ad esso e lo introdurremo come linguaggio parallelo al BASIC!), e adotteremo, per indicare una struttura di selezione, il costrutto:

IF... THEN....ELSE....;

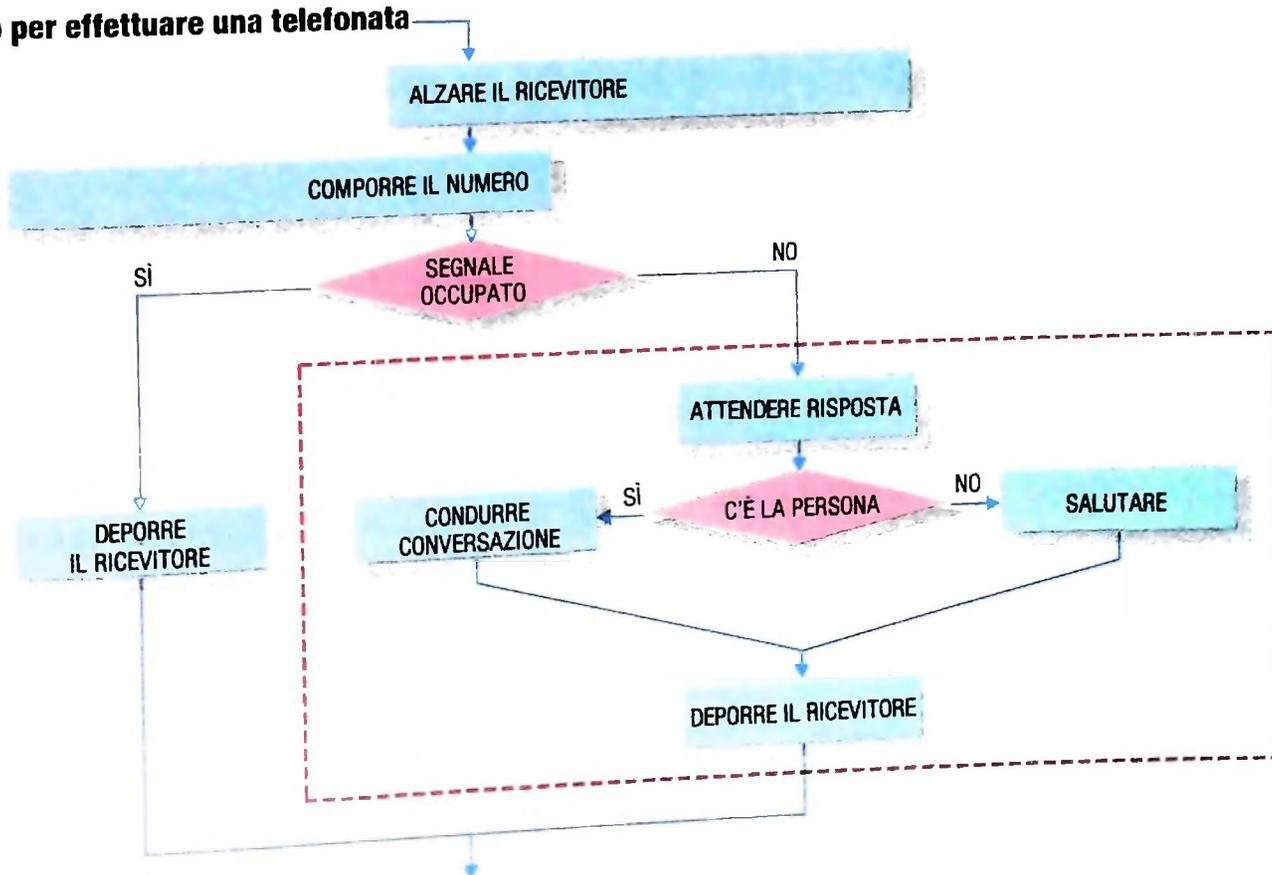
Così, l'algoritmo precedente può essere ulteriormente ampliato come segue:

Algoritmo per effettuare una telefonata

- Alzare il ricevitore
- Comporre il numero
- IF segnale occupato THEN
  - Deposare il ricevitore
- ELSE
  - Attendere la risposta
  - IF la persona c'è THEN
    - Condurre la conversazione
  - ELSE
    - Salutare
- Deposare il ricevitore

L'algoritmo presenta due strutture IF...THEN...ELSE...; innestate una nell'altra, secondo lo schema illustrato in figura:

**Algoritmo per effettuare una telefonata**



## Strutture di selezione in BASIC

Il linguaggio BASIC presenta due istruzioni fondamentali con cui realizzare modelli di selezione:

l'istruzione IF...THEN

e

l'istruzione GOTO.

La prima si comporta come segue:

```
10 IF A=1 THEN 100
```

Il suo significato è:

se la variabile A ha valore 1, allora salta all'istruzione che ha numero di linea 100 e prosegui da lì l'esecuzione del programma  
altrimenti, continua con l'istruzione successiva.

L'istruzione:

```
20 GOTO 50
```

invece determina la prosecuzione dell'esecuzione saltando all'istruzione con numero di linea 50 ("GOTO" in inglese vuole proprio dire "salta a"). Vedremo nel seguito come usarle per realizzare programmi con selezione.

### Cosa abbiamo imparato

In questa lezione abbiamo imparato:

- le **SEQUENZE DI ESECUZIONE**, cioè la successione di operazioni innescata dall'esecuzione di un programma.
- il concetto di **STRUTTURA DI CONTROLLO**, come strumento di un linguaggio di programmazione che non opera trasformazioni, ma permette di dirigere le operazioni scegliendo tra diverse sequenze d'esecuzione in dipendenza di **CONDIZIONI** che vengono controllate durante l'esecuzione stessa;
- la **STRUTTURA DI SELEZIONE**, che permette di scegliere una tra due vie possibili, che seguendo la nomenclatura inglese abbiamo indicato come **IF... THEN... ELSE...**
- la traduzione delle strutture **IF... THEN... ELSE** in **BASIC**, con l'uso delle istruzioni:
  - IF condizione THEN numero di linea
  - GOTO numero di linea.

# TOTOCALCOMPUTER

Un programma per fare pronostici sulla schedina del totocalcio.

I metodi per sviluppare un programma di pronostici del Totocalcio sono molteplici: si differenziano, sostanzialmente, nel numero di variabili di cui tengono conto per elaborare il pronostico. I più semplici sono basati solo ed esclusivamente sull'analisi della posizione in classifica delle singole squadre, quelli più complessi prendono in considerazione il rendimento, nel campionato in corso, di ogni singola squadra, il suo bioritmo "collettivo", il fattore campo e, infine, la statistica dei risultati.

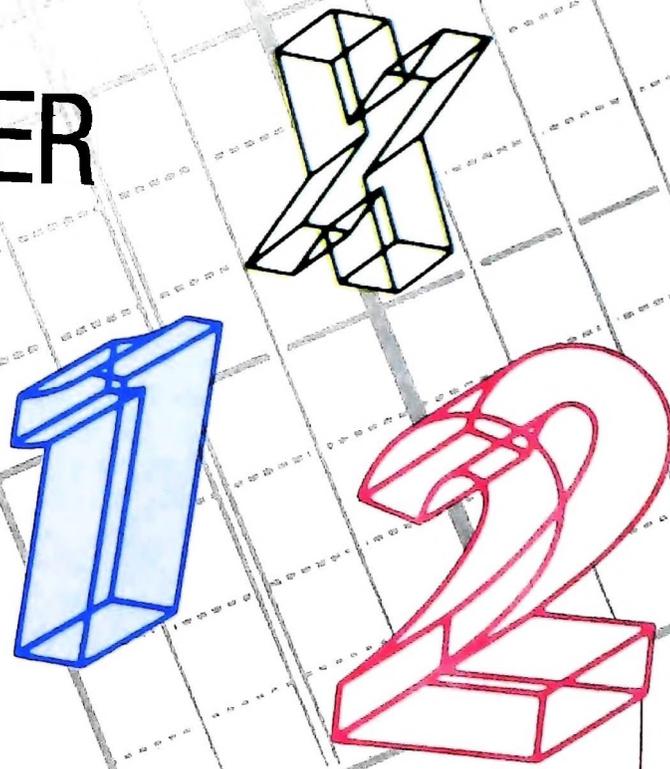
Il programma presentato non ha un alto grado di complessità, questo per permettere di comprendere, agli inizi, la metodologia di analisi di un problema ed il relativo sviluppo. Le previsioni sono basate sull'analisi della differenza di classifica tra le squadre contendenti e del rendimento nelle partite in casa e in trasferta di ogni singola squadra.

## Il programma

Il programma, scritto in un BASIC semplice, e corredato di istruzioni facilmente comprensibili, offre il vantaggio, a coloro che ne siano interessati, di apportarvi modifiche nel senso

di ampliarlo o di variarne la metodologia di previsione dei risultati.

Avremo, infatti, modo di vedere, in seguito, come, semplicemente utilizzando la medesima "struttura di base" del programma, sia possibile introdurre delle opportune routines



## Analizziamo più da vicino il programma

Linee 1-10	Commenti
Linee 19-35	Riserva una certa quantità di memoria per le variabili stringa (come default, "valore prestabilito all'accensione", la macchina riserva 256 byte). Dimensiona le variabili utilizzate. Ciò si rende necessario quando si debba lavorare con variabili indicizzate (vettori) con un numero di elementi superiore a 10 (tale infatti è la dimensione assegnata come default). Le variabili utilizzate hanno il seguente significato: P1\$(K) = nome della squadra in casa P2\$(K) = nome della squadra in trasferta L1(K) = punti in classifica della squadra P1 L2(K) = punti in classifica della squadra P2 H1(K) = punti in casa della squadra P1 H2(K) = punti in casa della squadra P2 P02(K) = punti in trasferta della squadra P2 F1\$(K), F2\$(K), F3\$(K) = pronostici
Linee 40-60	Visualizzazione e richiesta di informazioni
Linee 60-190	Richiesta di inserimento delle partite in schedina Richiesta di inserimento classifica e punti in casa
Linee 220-770	Stampa dei risultati sul display
Linee 1000-1090	Routine di computo numero colonne e costo
Linee 5000-5550	Routine di assegnazione pronostici
Linee 6000-8150	Routine di intestazioni
Linee 9000-9090	Routine di visualizzazione menù.

esterne allo scopo di aumentare il grado complessivo di sofisticazione dell'elaborazione.

### Note sull'utilizzo

Per comprendere meglio l'assegnazione dei pronostici, teniamo conto che l'istruzione BASIC SGN (X) utilizzata ha lo scopo di rispondere -1 se  $x < 0$ , 0 se  $x = 0$ , 1 se  $x > 0$ .

Il programma è abilitato a visualizzare una serie di 5 righe, dopo di che, per proseguire, basta digitare un tasto qualsiasi: questo determinerà la perdita in memoria delle informazioni precedenti e la visualizzazione delle prossime.

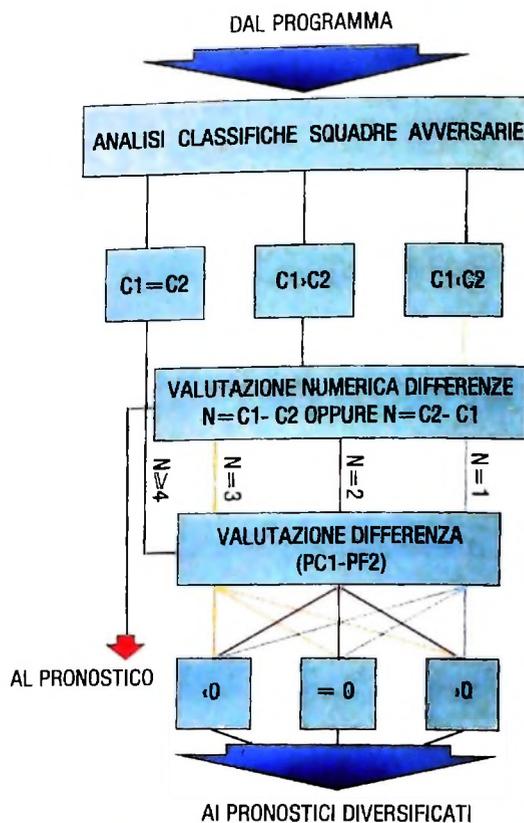
Da ricordare anche che, in fase di inserimento del programma, è sufficiente salvarlo una prima volta, dopo di che ogni qualvolta lo si richiamerà per inserire nuove istruzioni o apportare modifiche, tali operazioni verranno automaticamente memorizzate dall'M10 sotto il nome del programma che ha in memoria.

N.B. per inserire la "|" (barra verticale), bisogna digitare i tasti GRPH e H

Per rendere più chiaro quale sia stato il processo logico di assegnazione dei tre valori 1, x, 2 e delle loro combinazioni, analizziamo lo schema a destra ricordando che con PC1 si sono definiti i punti conquistati in casa dalla 1ª squa-

dra, con PF2 quelli ottenuti in trasferta dalla 2ª squadra, con C1 e C2 rispettivamente i punti in classifica della 1ª e 2ª squadra. I simboli >, =, < indicano rispettivamente "MAGGIORE di", "MINORE di", "UGUALE a".

### Schema procedurale routine 5000-5500



```

1  '*****
2  ' *
3  ' *      TOTOCALCOMPUTER
4  ' *
5  ' *  versione 1.1      by      C.V.P.
6  ' *
7  '*****
10 '
18 GOTO9000
19 CLEAR2500:CLS:SCREEN0,0
20 DIMP1$(13),P2$(13),L1(13),L2(13),H1(13),H2(13)
30 DIMPO2(13),F1$(13),F2$(13),F3$(13)
35 CLS
40 PRINTTAB(5)"TOTOCALCOMPUTER  by  C.V.P.":LINE(0,8)-(239,8)
45 PRINT@90,"Concorso nr. ";:INPUTCO%
50 PRINT@211,"Quando sei pronto
60 A$=INKEY$:IFA$=""THEN60                                premi un tasto"
70 GOSUB6000
80 FOR K= 1 TO 13
85 IF K = 5 OR K = 9 OR K=13 THEN GOSUB 6000
90 PRINT K;:PRINT "■";:INPUT P1$(K)
92 S = S + 40:PRINT @S,"■";:INPUT P2$(K)
95 NEXT K
100 GOSUB6040
110 FOR K = 1 TO 13
120 IF K =3 OR K=5 OR K=7 OR K=9 OR K=11 OR K=13 THEN GOSUB 6040
130 PRINT P1$(K);:PRINT TAB(17);:INPUT L1(K)
140 S = S + 40:PRINT @S," ";:INPUT H1(K)

```

```

150 PRINT P2$(K);:PRINT TAB(17);:INPUT L2(K)
160 S = S + 40:PRINT @S," "":INPUT H2(K)
170 P02(K) = L2(K) - H2(K)
180 C1 = L1(K):C2 = L2(K)
190 PC1 = H1(K):PF2 = P02(K)
200 GOSUB5000
210 NEXT K
220 CLS:PRINT "Concorso nr. ";C0%:PRINT " del:";DT$
225 LINE(0,8)-(239,8)
230 PRINT""
703 PRINT " ";
705 PRINT STRING$(12,244);:PRINT " Pronostici ";:PRINT STRING$(12,244)
707 PRINT ""
720 FORK=1TO13
730 IF K=4 OR K=7 OR K=10 THEN GOSUB 8000
740 PRINT K;:PRINT "■";P1$(K);"-";P2$(K);:PRINTTAB(31);
742 PRINTF1$(K);F2$(K);F3$(K)
760 NEXT K
770 A$=INKEY$:IF A$=""THEN 770
780 GOTO 9000
900 '*****
910 '* CALCOLO NR. COLONNE E COSTI
920 '*****
1000 CLS
1010 PRINT " CALCOLO NUMERO COLONNE E COSTO"
1020 LINE(0,8)-(239,8)
1030 PRINT""
1040 INPUT"Numero Triple:";NT
1050 INPUT"Numero doppie:";ND
1055 INPUT"Costo per colonna:";CC
1060 CL=(3^NT*(2^ND))
1070 SP=CC*CL
1080 PRINT"Numero colonne=";CL;:PRINT" Costo= #.";SP
1090 A$=INKEY$:IF A$="" THEN 1090
1100 GOTO 9000
4500 '*****
4505 ' routine pronostici
4510 '*****
5000 ON SGN(C1 - C2)+2 GOTO 5150,5200,5100
5100 N = (C1-C2):IF N>4 THEN N=4
5105 ON N GOTO 5110,5115,5120,5520
5110 ON SGN(PC1-PF2)+2 GOTO 5500,5510,5500
5115 ON SGN(PC1-PF2)+2 GOTO 5530,5510,5520
5120 ON SGN(PC1-PF2)+2 GOTO 5530,5520,5520
5150 N=(C2-C1):IF N>4 THEN N=4
5155 ON N GOTO 5160,5165,5170,5540
5160 ON SGN(PC1-PF2)+2 GOTO 5500,5500,5530
5165 ON SGN(PC1-PF2)+2 GOTO 5550,5530,5510
5170 ON SGN(PC1-PF2)+2 GOTO 5550,5550,5530
5200 IF PC1=PF2 THEN 5500
5210 IF PC1>PF2 THEN 5300
5230 IF PC1<PF2 THEN 5400
5300 N=(PC1-PF2):IF N>4 THEN N=4
5310 ON N GOTO 5500,5510,5510,5520
5400 N=(PF2-PC1):IF N>4 THEN N=4
5410 ON N GOTO 5500,5530,5550,5540

```

```

5500 F1$(K)="1":F2$(K)="X":F3$(K)="2":RETURN
5510 F1$(K) = "1":F2$(K) = "X":RETURN
5520 F1$(K) = "1":RETURN
5530 F1$(K) = "X":RETURN
5540 F1$(K) = "2":RETURN
5550 F1$(K) = "X":F2$(K) = "2":RETURN
6000 CLS:PRINTTAB(3)"Concorso nr. ";CO%;" del:";DT$:LI
      NE(0,8)-(239,8):S=99
6010 PRINT " ":PRINT"nr.■";:PRINT"SQUADRA IN CASA■";:PRINT"SQUA
      DRA FUORI CASA■"
6020 RETURN
6040 CLS: PRINT" ti invito ad inserire i seguenti dati":S=105
6042 LINE(0,8)-(239,8):PRINT ""
6050 PRINT STRING$(2,244);:PRINT " SQUADRA ";
6051 PRINT STRING$(6,244);:PRINT " PUNTI ";
6052 PRINT STRING$(1,244);:PRINT "P.in Casa ";
6053 PRINT STRING$(1,244)
6060 RETURN
7910 PRINT " "
8000 A$=INKEY$:IFA$=""THEN8000
8100 CLS:PRINT" Concorso nr ";CO%::PRINT" del:";DT$
8110 LINE(0,8)-(239,8)
8115 PRINT " "
8120 PRINT " ";
8130 PRINT STRING$(12,244);:PRINT " Pronostici ";:PRINT STRING$(12,244)
8140 PRINT ""
8150 RETURN
8900 '*****
8910 '* VISUALIZZAZIONE MENU *
8920 '*****
8930 '
9000 CLS
9010 LINE(10,6)-(229,6)
9020 LINE(229,6)-(229,60)
9030 LINE(229,60)-(10,60)
9040 LINE(10,60)-(10,6)
9050 PRINT@87,"1. Pronostici Totocalcio"
9060 PRINT@167,"2. Calcolo nr.colonne e costo"
9070 PRINT@245,"seleziona: ";:INPUTB
9080 IFB>2 OR B<1 THEN9070
9090 ON B GOTO 19,1000
9091 '

```

FINE DEL LISTATO

*N.B. Il ● sta a significare che la linea va a capo per esigenze editoriali, quindi nell'eseguirlo non interrompere la digitazione.*

### Esecuzione del programma su altri computer

Il programma allegato può essere eseguito anche su altri computer che non siano necessariamente M10. È però necessario apportarvi alcune modifiche specie per quanto riguarda la gestione video.

Analizziamole fase per fase:

- 1) eliminare tutte le istruzioni LINE (X,Y)-(X,Y);
- 2) in linea 19. eliminare l'istruzione SCREEN 0,0;

- 3) calcolarsi S che identifica la posizione sul video in cui stampare;
- 4) nelle linee 120,730 si sono fissati particolari valori di K per permettere la visualizzazione di K righe contemporanee sul display; se si disponesse di video con 16,24,48 linee, tali istruzioni potrebbero essere anche eliminate;
- 5) controllare i valori decimali nelle istruzioni STRING (verificare che il carattere stampato non sia incompatibile con il proprio computer).

# LA NOTA

L'informazione musicale trasmessa dipende dall'esecutore e dall'ascoltatore e ha differenti livelli di rappresentazione.

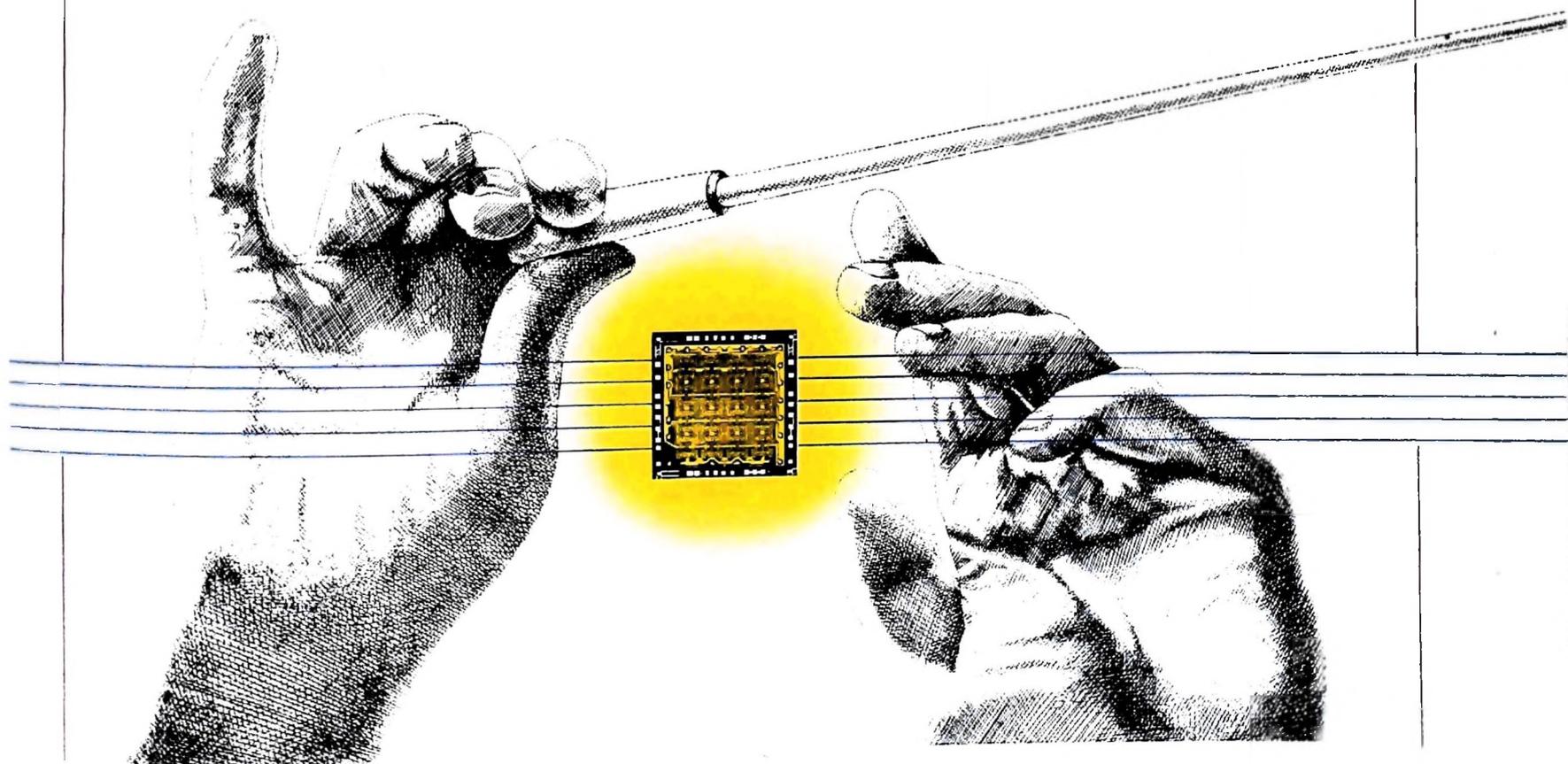
Ascoltando un brano musicale, ognuno di noi riconosce suoni e alcune caratteristiche, proprietà e forme musicali; a seconda dell'attenzione prestata in quel particolare momento, delle esperienze musicali precedenti e di molti altri fattori, le caratteristiche che percepiamo, sia acustiche che strutturali, sono generalmente differenti. In altre parole, l'*informazione musicale* che viene trasmessa dipende sia dal trasmettitore (l'esecutore) che dal ricevitore (l'ascoltatore). Sulla base di questa osservazione si articoleranno le discussioni di questa rubrica; cercheremo di comprendere (e quindi usare) i criteri che regolano l'organizzazione dei fenomeni musicali.

Notiamo, inoltre, che l'informazione musicale può assumere varie forme di rappresentazione a seconda che si pensi alla musica come a una sequenza di vibrazioni dell'aria o a una forma (astratta) di composizione musicale. In realtà, sono due modi diversi di osservare e di descrivere un fenomeno musicale. Possiamo, in altre parole, distinguere differenti li-

velli di rappresentazione dell'informazione musicale: un livello *strutturale* (corrispondente alla *forma* musicale), un livello *simbolico* (corrispondente alla notazione del pentagramma), un livello *fisico* (corrispondente alla descrizione del processo acustico), un livello *operativo* (corrispondente alla definizione delle azioni necessarie per eseguire un testo musicale).

Se usiamo un elaboratore per attività musicali possiamo comunicargli l'informazione musicale a qualunque livello di rappresentazione (generalmente a più livelli per ogni singolo testo); l'unica condizione è che gli forniamo una descrizione *completa*: l'elaboratore non è capace di interpretare le nostre istruzioni come uno strumentista d'orchestra.

Le prestazioni che ci fornisce un elaboratore riflettono generalmente la modalità ciclica in cui diamo informazioni, in cui l'elaboratore le trasforma, in cui riceviamo i risultati e in base ad essi diamo altre informazioni e così via, fino al conse-



guimento di risultati soddisfacenti per il musicista. L'informazione che forniamo all'elaboratore e quella che ci viene restituita dall'elaboratore possono presentarsi nella forma grafica (partiture, testi codificati ecc.) o sonora. In base a questa distinzione, caratterizziamo le sei principali attività di informatica musicale:

- analisi di testi musicali;
- elaborazione di testi musicali;
- sintesi di testi musicali;
- analisi del suono;
- elaborazione del suono;
- sintesi del suono.

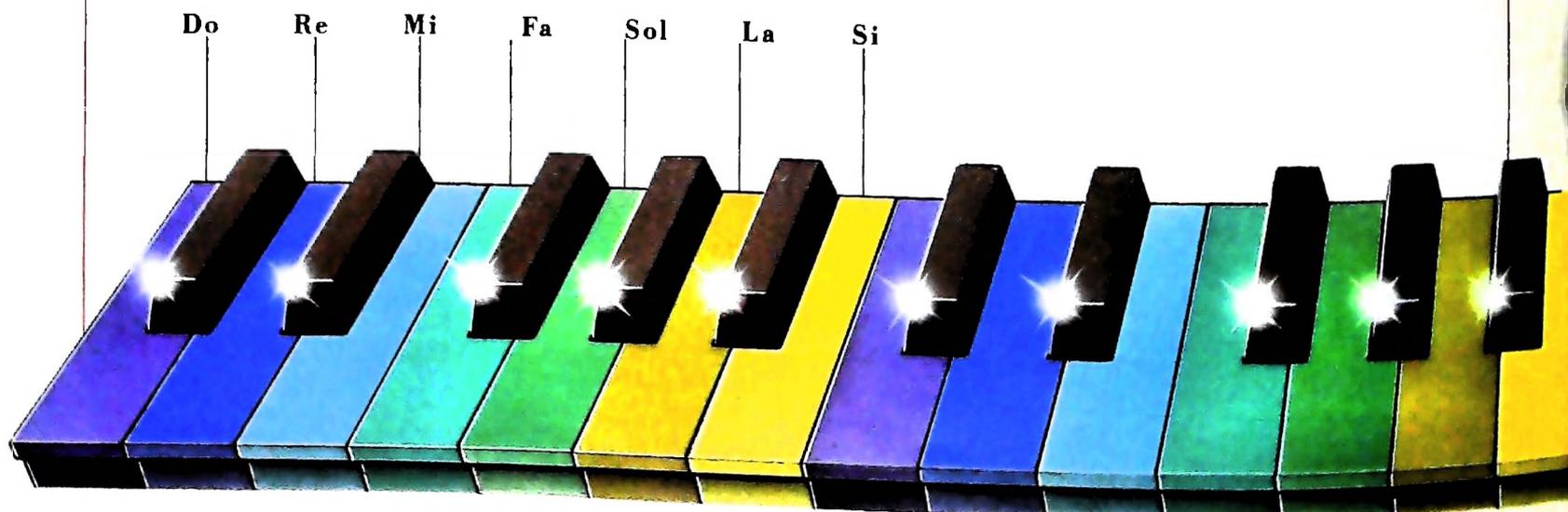
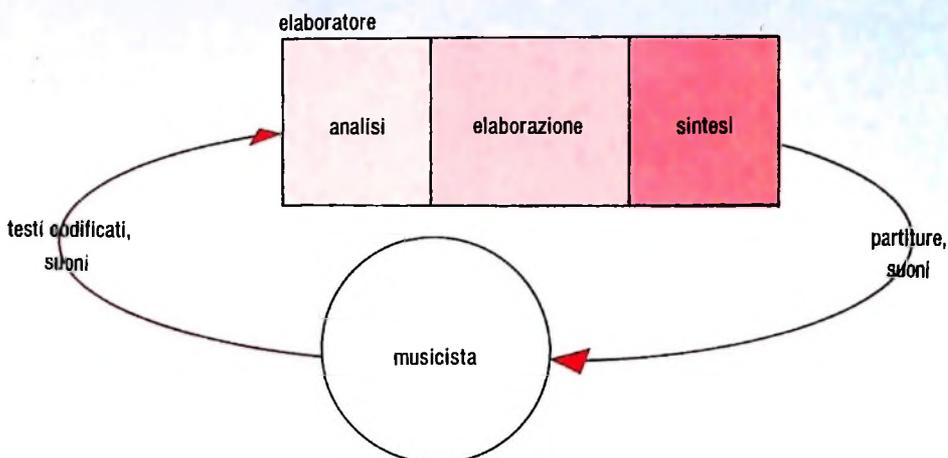
Mentre possiamo realizzare attività dei primi tre tipi mediante elaboratori *general purpose* (cioè non specializzati) anche di dimensioni e costi modesti, la manipolazione del suono (analisi, elaborazione o sintesi) richiede dispositivi particolarmente costosi e specializzati; il motivo essenziale è che un suono di buona qualità audio non è descrivibile con meno di 40-50 000 numeri costituiti, ciascuno, da almeno 16-24 cifre binarie (cifre che possono valere solo 0 o 1) per la durata di

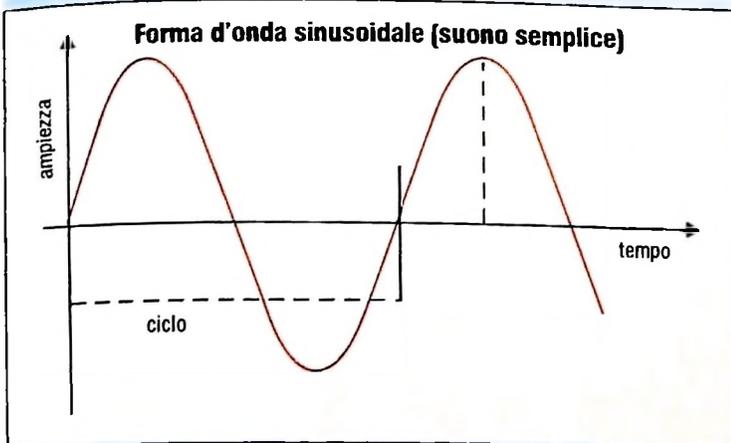
un secondo: ciò comporta un onere molto alto di calcoli e di spostamenti di informazioni in tempi molto brevi. Dispositivi capaci di soddisfare queste caratteristiche hanno ancora costi molto elevati (centinaia di milioni di lire) anche se è prevedibile un loro abbattimento nel prossimo decennio. Nell'ambito di questa rubrica faremo frequente ricorso a esempi acustici, riferendoci a un dispositivo a basso costo, l'elaboratore M10, ma comunque in grado di svolgere tutte le attività di informatica musicale a livello del testo e di sintetizzare suoni controllabili in frequenza e durata.

## Il suono

Nella nostra analisi partiremo dall'elemento basilare del fenomeno musicale: il *suono*. Ogni suono che percepiamo è caratterizzato da almeno quattro parametri percettivi fondamentali: l'*altezza* (suono grave/acuto), l'*intensità* (suono piano/forte), il *timbro* e la *durata* (suono breve/lungo). Ognuno di questi parametri dipende dai corrispondenti parametri fisici del fenomeno ondulatorio che ha originato il suono; l'altezza dipende dalla *frequenza* della vibrazione (quando il suono è *periodico*, cioè determinato), l'intensità dipende dall'*ampiezza*, il timbro dalla evoluzione della *composizione spettrale*, la durata dalla *estensione temporale* del suono.

Il diagramma a destra rappresenta il tipico ciclo di elaborazione dell'informazione musicale: il musicista fornisce le informazioni, il computer le elabora e fornisce i risultati, in base ai quali il musicista fornisce ulteriori informazioni, e via di seguito fino a che non vengono raggiunti i risultati desiderati. Le informazioni che vengono fornite alla macchina, così come quelle restituite in uscita dalla macchina, possono presentarsi in forma grafica (come partiture, testi codificati ecc.) o sonora.





Così, se vogliamo produrre un suono mediante uno strumento tradizionale (ad esempio un pianoforte), scegliamo un tasto a cui corrisponde un valore di altezza, lo premiamo con una certa forza a cui corrisponde un valore di intensità, lo teniamo premuto per un certo tempo a cui corrisponde la durata del suono; la definizione del timbro è implicita nella scelta del particolare strumento.

Se invece vogliamo generare un suono mediante un elaboratore, dobbiamo prima fornirgli la *descrizione* del suono (la partitura, il testo musicale) e poi le *istruzioni per produrre* il suono (l'interpretazione del testo, i comandi ai dispositivi

che producono il suono).

Con M10, si può descrivere un suono coi parametri di frequenza e di estensione temporale; le caratteristiche spettrali e l'ampiezza delle vibrazioni sono fisse. Con l'istruzione:

**SOUND x, y**

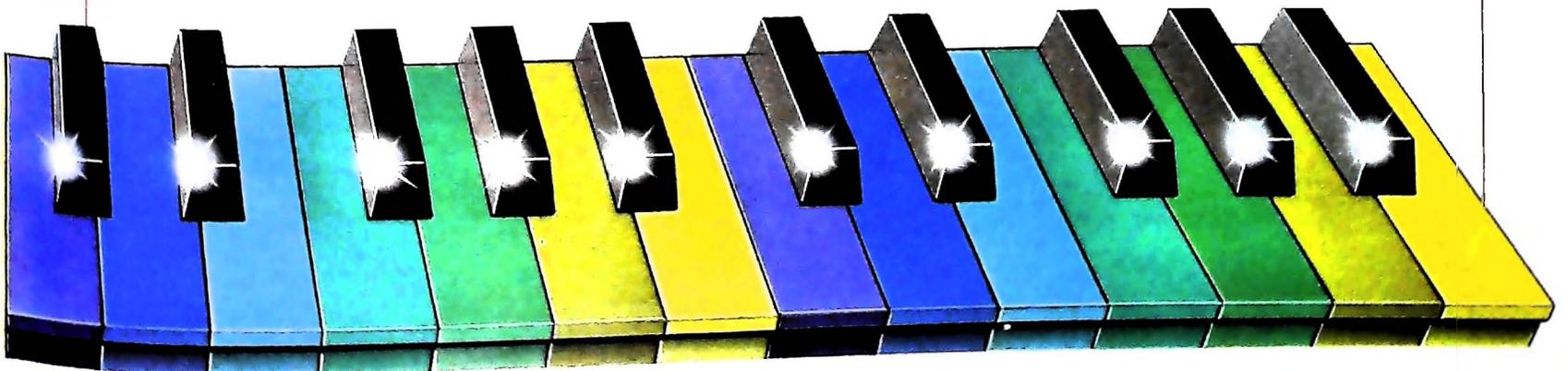
specifichiamo un suono di frequenza inversamente proporzionale a x e di estensione temporale proporzionale a y.

### La notazione dell'altezza

Nella musica del mondo occidentale si usa un insieme ben definito di valori di altezze a cui sono associati i nomi *do* (C, nella notazione anglosassone), *re* (D), *mi* (E), *fa* (F), *sol* (G), *la* (A), *si* (B); queste altezze hanno tra loro precisi rapporti di frequenza:  $9/8$  tra *do* e *re*,  $5/4$  tra *do* e *mi*,  $4/3$  tra *do* e *fa*,  $3/2$  tra *do* e *sol*,  $5/3$  tra *do* e *la*,  $15/8$  tra *do* e *si*. Questa serie di intervalli si ripete dal grave all'acuto per i diversi valori di frequenza all'interno delle diverse *ottave*; l'ottava è appunto l'intervallo di frequenza tra un suono e il suono di frequenza doppia del primo; ad esempio, il rapporto tra la frequenza di un *do* e il *do* dell'ottava successiva è 2. Così, dopo un *si* abbiamo nuovamente un *do*, seguito da un *re* ecc. Sulla tastiera del pianoforte le altezze qui menzionate corrispondono ai ta-



Sotto, tastiera di un pianoforte, con l'identificazione del nome dei tasti delle note intere. Nel pianoforte, l'altezza di una nota cresce da sinistra verso destra. Qui accanto, la rappresentazione di due ottave sul rigo musicale: le note sovrapposte hanno lo stesso nome, ma si trovano alla distanza di una ottava (la superiore ha frequenza doppia dell'inferiore). Il *do* rappresentato subito sotto il pentagramma nella chiave di violino è il *do* "centrale" del pianoforte.



sti bianchi; ogni sette tasti bianchi cambia ottava. Sul pentagramma la notazione dell'altezza corrisponde alla posizione verticale sul rigo. Il segno all'inizio del rigo è la *chiave di sol* (detta anche di *violino*); serve a fissare il valore di altezza di riferimento, il *sol*, sulla linea che attraversa il suo centro (la seconda del rigo partendo dal basso). Sul rigo sono rappresentate due ottave: le note sovrapposte hanno lo stesso nome, ma si trovano alla distanza di un'ottava.

Guardando la tastiera del pianoforte notiamo anche la presenza di tasti neri a cui corrispondono altezze intermedie rispetto alle altezze dei tasti bianchi adiacenti. L'insieme dei sette tasti bianchi e dei cinque tasti neri di ogni ottava costituisce l'insieme dei dodici *semitoni* di un'ottava. Il semitono è l'intervallo di altezza corrispondente numericamente alla radice dodicesima di 2, che esprime il rapporto tra le frequenze che determinano le altezze tra cui è definito l'intervallo. Due semitoni costituiscono l'intervallo di un *tono*; tra due tasti bianchi contigui c'è l'intervallo di un semitono (*mi-fa, si-do*), tra due tasti bianchi tra cui è inserito un tasto nero c'è l'intervallo di un tono; tra un tasto bianco e il tasto nero adiacente c'è l'intervallo di un semitono. La suddivisione dello spazio di frequenze secondo questo criterio è chiamata *temperamento equabile* (poiché l'ottava è suddivisa in dodici semitoni uguali, cioè dodici intervalli di altezza uguali).

La programmazione di questo parametro deve perciò prevedere dodici valori di altezza per ogni ottava del campo di frequenze producibili dal dispositivo per la sintesi del suono utilizzato. L'istruzione **SOUND** permette di generare differenti valori di frequenza in corrispondenza con il valore del

parametro *x* che può variare tra 0 (il suono più acuto) e 16383 (il suono più grave). In particolare, in corrispondenza con le altezze previste dal temperamento equabile sono producibili le altezze in un campo di sette ottave.

Nella tabella è data l'indicazione delle altezze *do diesis, re diesis, fa diesis, sol diesis, la diesis*: sono le altezze corrispondenti ai tasti neri del pianoforte. Nel gergo musicale, sono chiamate *alterazioni* e possono essere indicate anche in altro modo a seconda del contesto musicale; le alterazioni sono generalmente indicate dal *diesis* (viene indicato #) che alza il valore di altezza a cui si riferisce di un semitono, dal *bemolle* (viene indicato con b) che abbassa il valore di altezza a cui si riferisce di un semitono, dal *doppio diesis* (viene indicato con x o con # #) che alza di un tono, dal *doppio bemolle* (viene indicato con b b) che abbassa di un tono; inoltre, il *bequadro* (viene indicato con ♮) riporta il valore dell'altezza a cui si riferisce al valore iniziale; ad esempio, un *la* a cui è stato applicato un diesis, ritorna un *la* dopo l'applicazione di un bequadro. Certe altezze si indicano con diverse notazioni equivalenti:

*do diesis = re bemolle    la diesis = si bemolle*  
*re diesis = mi bemolle    si diesis = do*  
*mi diesis = fa                mi = fa bemolle*  
*fa diesis = sol bemolle    si = do bemolle*  
*sol diesis = la bemolle*

Analogamente, se applichiamo le alterazioni doppie (doppio diesis o bemolle).

A destra, la rappresentazione dei dodici semitoni in un'ottava do-do, sul pentagramma in chiave di violino. I valori della tabella sono identificati con il nome della nota corrispondente e il numero dell'ottava (l'ottava 1 è la più bassa nella gamma di M10, la 7 la più alta).



VALORI DI X NELL'ISTRUZIONE SOUND PER LA SINTESI DI FREQUENZE AUDIO							
nota/ottava	1	2	3	4	5	6	7
do		9394	4697	2348	1171	587	293
do diesis		8866	4433	2216	1103	554	277
re		8368	4184	2092	1043	523	261
re diesis	15800	7900	3950	1975	987	493	247
mi	14912	7456	3728	1864	932	466	233
fa	14064	7032	3516	1758	879	439	220
fa diesis	13284	6642	3321	1660	830	415	207
sol	12538	6269	3134	1567	783	392	196
sol diesis	11836	5918	2954	1479	739	370	185
la	11172	5586	2793	1396	693	349	174
la diesis	10544	5272	2636	1318	659	329	165
si	9952	4968	2484	1244	622	311	155

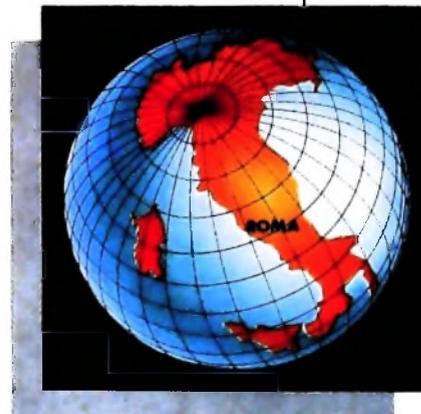
# Banca Aperta

## LE NUOVE RISPOSTE DEL BANCO DI ROMA.

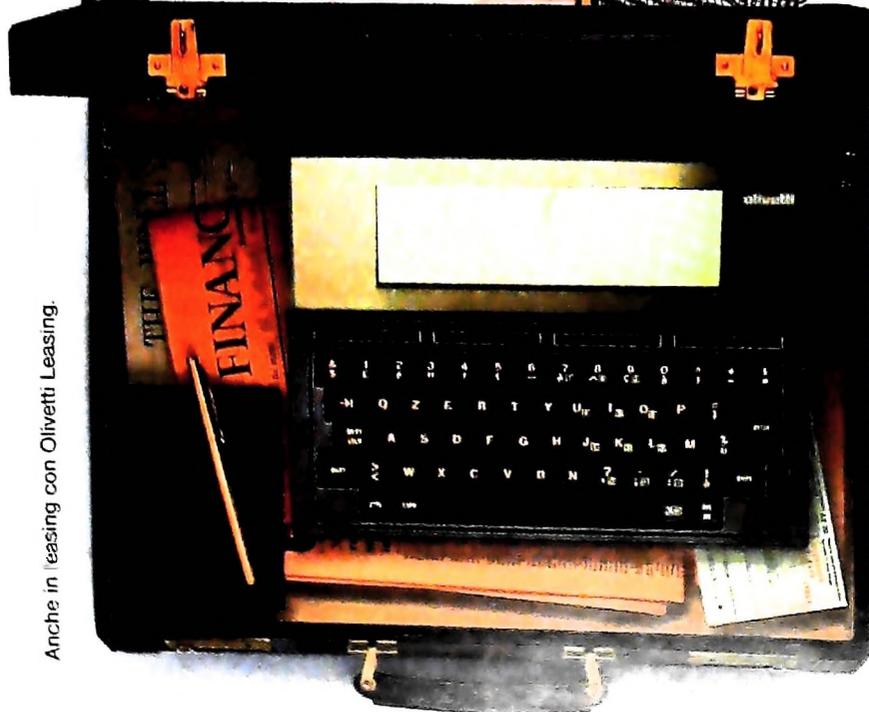
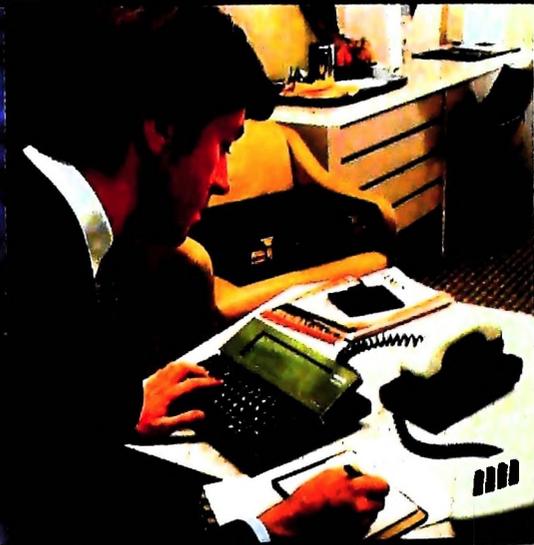
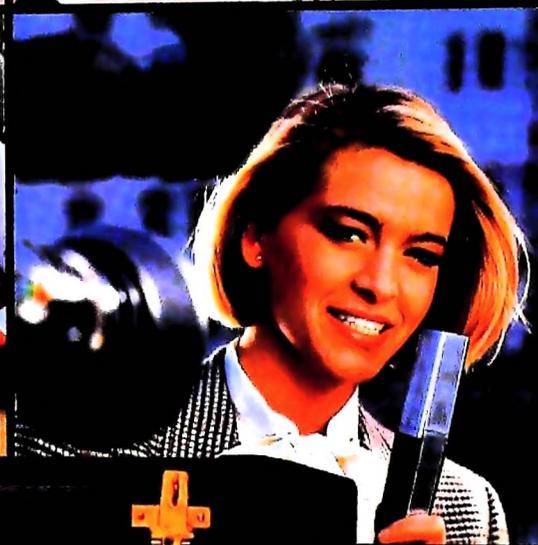
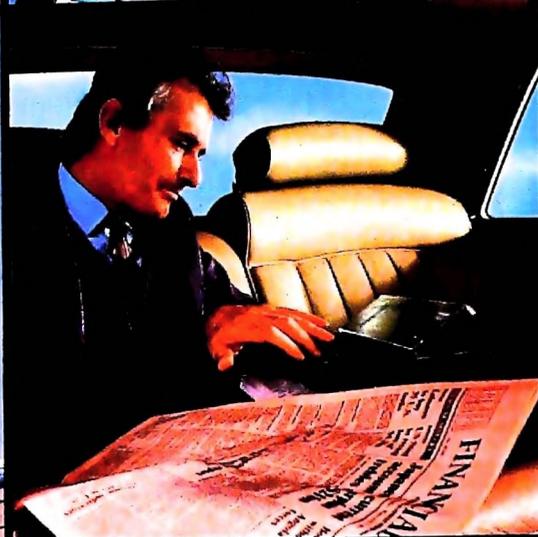
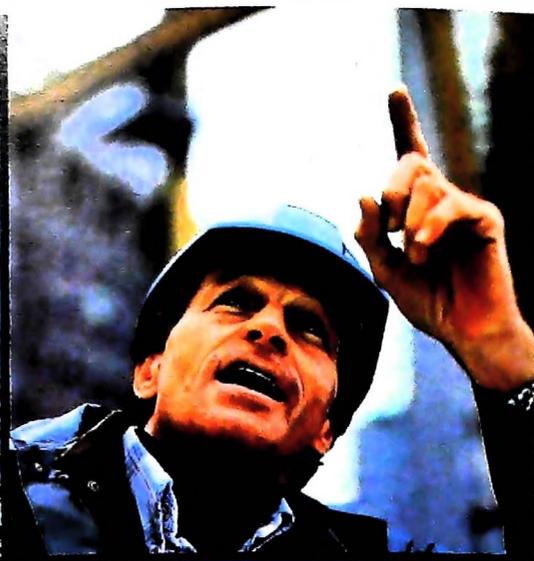


*Vorrei avere  
un rapporto più diretto  
con la mia banca...*

Anche le strutture bancarie si evolvono. Il Banco di Roma, primo in Italia, sta introducendo la struttura a "banca aperta", già attuata da molte sue filiali italiane. "Banca aperta": non il solito bancone, le lunghe file, ma un nuovo modo di essere banca, un rapporto più personalizzato, un clima più agevole, più professionale e una maggiore rapidità in ogni operazione. Un ulteriore passo avanti verso la completa consulenza finanziaria che il Banco di Roma intende mettere a disposizione dei propri clienti. Tra i numerosi servizi offerti ricordiamo: Prestito Personale. Prestito Casa, gestione dei patrimoni. Leasing, assistenza all'import-export, attraverso ben 60 sedi estere in 30 Paesi dei 5 continenti. Tutto questo perché il Gruppo Banco di Roma è in grado di gestire ogni servizio specifico con grande professionalità, fornendo anche informazioni dirette a domicilio attraverso i sistemi Videotel e Voxintesi.



 **BANCO DI ROMA**  
CONOSCIAMOCI MEGLIO.



## PERSONAL COMPUTER OLIVETTI M10 L'UFFICIO DA VIAGGIO

Olivetti M10 vuol dire disporre del proprio ufficio in una ventiquattre. Perché M10 non solo produce, elabora, stampa e memorizza dati, testi e disegni, ma è anche capace di collegarsi via telefono per spedire o ricevere informazioni.

Qualunque professione sia la vostra, M10 è in grado, dovunque vi troviate, di offrirvi delle capacità di soluzione davvero molto grandi. M10: il più piccolo di una grande famiglia di personal.

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

NOME/COGNOME

VIA/N°

CAP/CITTA'

TELEFONO

Anche in leasing con Olivetti Leasing.

# olivetti