

CADTEL

Spediz. in abbonamento postale GR II/70 L. 2.200
(...)

54 CORSO PRATICO COL COMPUTER

422063

F4 F5 F6 F7
diretto da GIANNI DEGLI ANTONI

è una iniziativa
FABBRI EDITORI

in collaborazione con
BANCO DI ROMA

e **OLIVETTI**



BANCO DI ROMA

Aut. Min. Conc.

IN EDICOLA
DAL 19 APRILE

CLASSICA
I MUSICISTI - LE OPERE - LA SOCIETÀ

Con il primo fascicolo
e la prima cassetta dedicata a

Bach

in REGALO il primo raccoglitore

L. 7.000

FABBRI EDITORI

FABBRI
EDITORI

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

Acquisto per contanti

È la formula di acquisto tradizionale.

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

Il servizio di finanziamento bancario

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

Banca Centro Sud
Banco di Perugia

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

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

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

I valori del credito

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

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

al "prime rate ABI (Associazione Bancaria Italiana) + 1,5 punti percentuali".

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

Le diverse forme di pagamento del finanziamento bancario

Il pagamento potrà avvenire:

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

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

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

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

Testi
DANIELE MARINI,
Eidos (TIZIANO BRUGNETTI), ANDREA GRANELLI, ALDO GRASSO,
CLAUDIO PARMELLI Etnoteam (ADRIANA BICEGO)

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

Direttore Editoriale
ORSOLA FENGHI

Redazione
CARLA VERGANI
LOGICAL STUDIO COMMUNICATION

Art Director
CESARE BARONI

Impaginazione
BRUNO DE CHECCHI
PAOLA ROZZA

Programmazione Editoriale
ROSANNA ZERBARINI
GIOVANNA BREGGÈ

Segretarie di Redazione
RENATA FRIGOLI
LUCIA MONTANARI

Corso Pratico col Computer - Copyright © sul fascicolo 1985 Gruppo Editoriale Fabbri, Bompiani, 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 - Distribuzione per l'Italia Gruppo Editoriale Fabbri S.p.A., via Mecenate, 91 - Milano - tel. 02/50951 - Pubblicazione periodica settimanale - Anno II - n. 54 - 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 cosueto da mutate condizioni di mercato.

 **BANCO DI ROMA**
CONOSCIAMOCI MEGLIO.

IL SISTEMA OPERATIVO CP/M (II)

Dopo aver esaminato le caratteristiche fondamentali del sistema analizziamone i singoli componenti.

Come è organizzata la memoria.

La memoria primaria "vista" dal CP/M viene riservata in parte a contenere il sistema operativo stesso e i suoi dati e in parte ai programmi utenti (figura a lato).

In figura si può vedere come è suddivisa la memoria di lavoro. I primi 256 byte sono riservati al sistema operativo per proprio uso interno; quindi all'indirizzo 100H, cioè 100 in esadecimale, comincia la porzione di memoria dedicata ai programmi utente. In corrispondenza, invece, degli indirizzi più alti della memoria trova la sua collocazione il sistema operativo.

A seconda delle dimensioni fisiche della memoria centrale la porzione che è riservata agli utenti potrà essere più o meno ampia.

Va ricordato, infatti, che il CP/M può operare in configurazioni di memoria variabili a partire da un minimo di 16 kbyte fino a un massimo di 64 kbyte.

Uno dei comandi messi a disposizione dal sistema operativo (MOVCPM) serve appunto a riconfigurare una versione del CP/M per adattarsi ad altri requisiti di memoria.

Come abbiamo detto i programmi utente vengono eseguiti nella zona di memoria chiamata TPA (Transient Program Area) e quindi, in linea di massima, le loro dimensioni debbono tener conto dell'ampiezza di quest'area. È possibile però per questi programmi "invadere" la zona riservata al sistema operativo, rinunciando ovviamente totalmente o parzialmente alle sue funzioni; sarà poi cura di questi stessi programmi ricaricare in memoria da disco il CP/M prima di terminare la propria esecuzione.

Periferiche logiche - periferiche fisiche

L'utente di CP/M dispone di quattro nomi di periferiche, che diremo "logiche" perché non sono collegate in modo stabile a dispositivi di ingresso e uscita di dati.

L'utilizzo delle periferiche "logiche" consente di evitare l'esigenza di una conoscenza dettagliata dell'hardware sottostante e di presentare all'utente una visione semplificata e a un maggiore livello di astrazione.

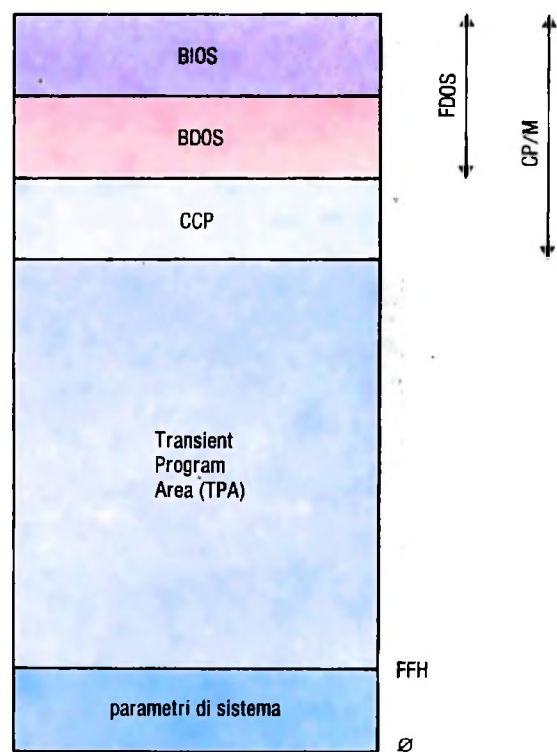
Tutti i comandi di input-output impostati da tastiera faranno quindi riferimento a queste periferiche "simboliche".

In realtà il sistema operativo deve indirizzare le operazioni di

input/output verso periferiche reali. L'utente non ha visibilità degli aspetti fisici di queste operazioni, che sono interamente a carico del CP/M.

La corrispondenza tra periferiche "logiche" e "fisiche" è risolta dal contenuto di una particolare cella di memoria situata a un indirizzo fisso, che è chiamata in linguaggio CP/M IOBYTE.

Il sistema operativo fa quindi riferimento al contenuto di IOBYTE per sapere a quale dispositivo fisico corrisponde una certa periferica logica.



Nella figura è schematizzata la struttura della memoria "principale". Si noti come il sistema operativo trova la sua collocazione in corrispondenza degli indirizzi più alti della memoria.

Se un utente vuole modificare le regole di assegnamento delle, ovviamente, alterare il contenuto di questa locazione di memoria.

Il CP/M riconosce un certo numero di periferiche fisiche. In questa pagina, sotto, ne è riportata una tabella.

Inoltre riconosce quattro periferiche logiche. Esse sono, per la precisione, le seguenti:

- CON: "console" o terminale per la comunicazione tra CP/M e utente;
- RDR: lettore di nastro o di schede; questa è una periferica che consente solo l'ingresso dei dati;
- PUN: perforatore di nastro o di scheda; questa è una periferica che consente solo l'uscita dei dati;
- LST: stampante.

Organizzazione dei file

Il CP/M, per mezzo delle sue componenti CCP e BDOS, traduce la visione logica dei file che ha l'utente in un trattamento fisico. Quest'ultimo dipenderà dal tipo particolare di supporto magnetico su cui è memorizzato il file; l'utente non avrà, comunque, nessuna visibilità di queste differenze e continuerà a gestire i file usando lo stesso tipo di comandi. Consideriamo ora, a titolo di esempio, un dischetto flessibile

a 8 pollici e a singola densità con un formato del tipo IBM 3740, del quale è riportata una rappresentazione nel disegno della pagina a fianco.

Questo disco è suddiviso in 77 fasce concentriche che si chiamano tracce. Ciascuna traccia è ulteriormente suddivisa in 26 settori di 128 byte ciascuno. Nel linguaggio CP/M questi settori vengono chiamati record.

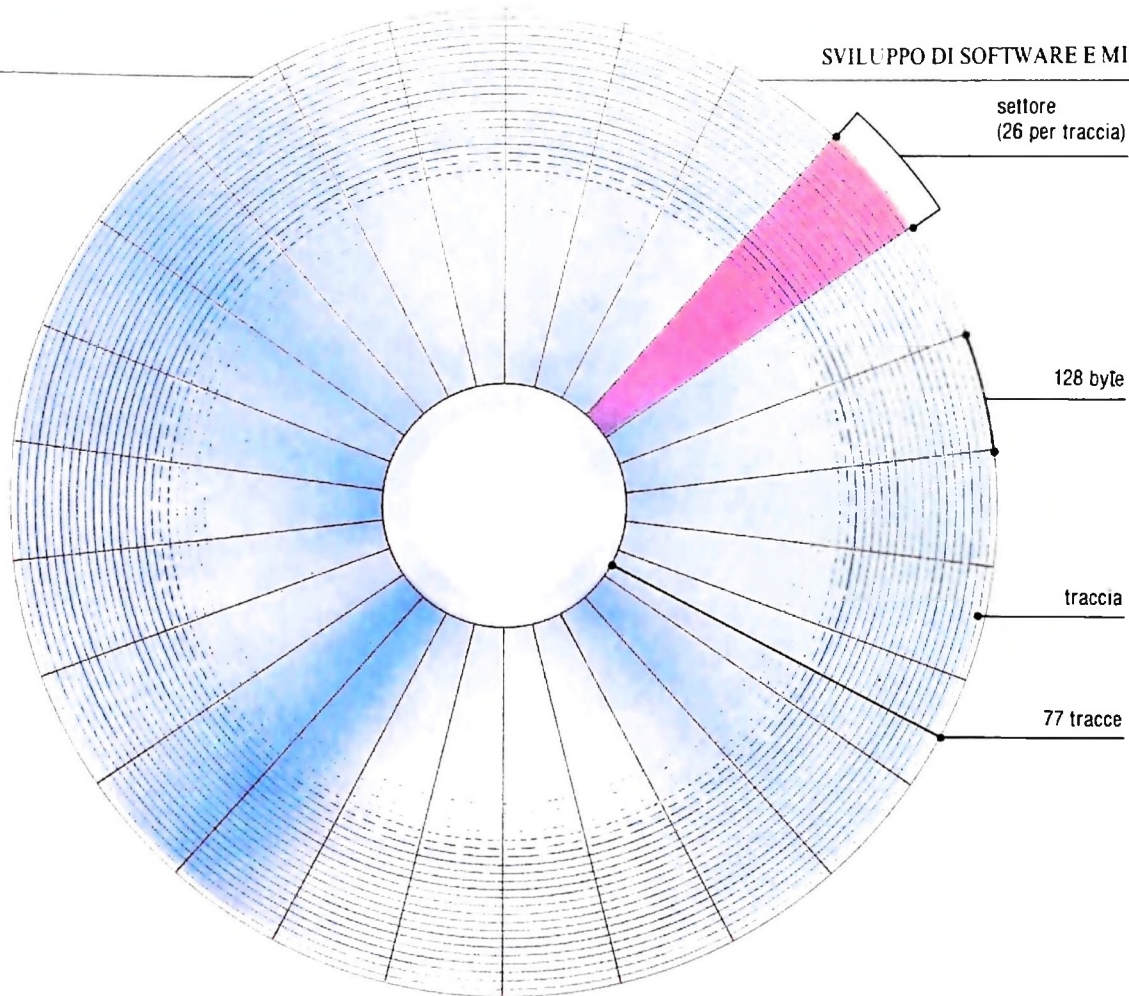
Come si può vedere nella figura in basso della pagina a fianco le prime due tracce del disco (tracce 0 e 1) sono destinate a contenere:

- 1) un programma capace di leggere il disco stesso e caricare in memoria il sistema operativo;
- 2) il CP/M stesso.

Una terza traccia (traccia 2) invece contiene informazioni riassuntive sui file memorizzati su disco, in altre parole una specie di indice del contenuto del disco. Infine tutto lo spazio rimanente contiene i file dell'utente e i file che contengono i comandi del sistema operativo non destinati a restare permanentemente nella memoria centrale.

Un file è costituito da un insieme di record che non sono, generalmente, fisicamente contigui. Questo per motivi di efficienza nella lettura o scrittura del file stesso; infatti dopo ogni settore che contiene dati utente una parte dello spazio su disco è usata per scrivere informazioni relative alla correttezza dei dati stessi o in generale alla gestione dei dati da

Periferiche fisiche	
TTY:	per una console o un terminale, un lettore, un perforatore o un dispositivo di stampa (teletype).
CRT:	una console, un terminale o un dispositivo di stampa (Tubo a raggi catodici).
PTR:	un dispositivo di lettura di nastro di carta o di schede.
PTP:	un dispositivo di perforazione di nastro di carta o di schede.
LPT:	un dispositivo di stampa (stampante).
UC1:	una console o un terminale definito dall'utente.
UR1:	un lettore definito dall'utente.
UR2:	un secondo lettore definito dall'utente.
UP1:	un dispositivo di uscita (perforatore) definito dall'utente.
UP2:	un secondo dispositivo di uscita (perforatore) definito dall'utente.
UL1:	un dispositivo di stampa definito dall'utente.



parte del sistema operativo.

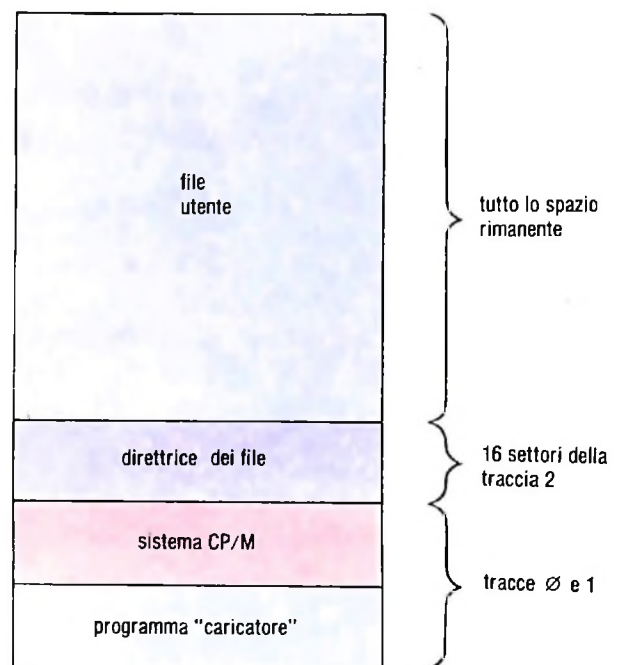
Dopo aver letto un settore, quindi, il calcolatore è impegnato a elaborare queste informazioni "di contorno" e perde la possibilità di leggere il record che segue immediatamente il precedente, che nel frattempo è passato sotto la testina di lettura del disco. Non dobbiamo dimenticarci, infatti, del fatto che un disco è in continua rotazione. Per poter leggere record contigui sarebbe quindi necessario riposizionare continuamente la testina di lettura, e questo comporterebbe un notevole spreco di tempo. Il CP/M sceglie invece di considerare "logicamente" contigui settori posizionati a una distanza fisica fissa, di solito pari a 6 record.

Data la non contiguità fisica dei record che costituiscono un file, il sistema operativo CP/M deve mantenere traccia della loro posizione effettiva.

Queste informazioni sono tutte contenute in un descrittore del file chiamato File Control Block (FCB), insieme al nome simbolico del file e ad altri dati che consentono al CP/M la gestione del file stesso.

I descrittori di tutti i file esistenti risiedono nella zona che ospita la direttrice del disco. Quando il sistema sta utilizzando un file, copia il suo FCB nella memoria centrale (nella Transient Program Area) per renderne più rapido l'accesso. La ricopierà su disco quando il file verrà chiuso.

Il CP/M riserva spazio ai file in modo dinamico. L'utente, quindi, non deve specificare a priori la dimensione di un file, come invece succede con altri sistemi operativi.



Nella pagina a fianco è riportata una tabella di periferiche fisiche riconosciute dal sistema operativo CP/M. In questa pagina, nella figura in alto, struttura fisica di un dischetto flessibile (formato del tipo IBM). Nel linguaggio CP/M i

settori in cui è suddiviso il disco vengono chiamati record. Nella figura in basso è indicata in modo schematico la struttura di un dischetto con sistema operativo CP/M. Sono indicati i contenuti delle tracce del disco.

I LABIRINTI

Haze, ovvero: l'importante è non perdersi in qualche vicolo cieco...

La figura del labirinto, il mito di una costruzione così complessa da potersi smarrire, ritrovarsi per poi perdersi nuovamente, ha da sempre emanato un fascino misterioso, tanto che, nel linguaggio comune, "labirinto" indica una situazione complessa, ingarbugliata, spesso inestricabile, dalla quale non si riesce a intravedere una via d'uscita.

La leggenda di Dedalo, artefice del famoso labirinto dove Minosse, re di Creta, lo rinchiusse insieme con il figlio Icaro per aver fornito ad Arianna il filo che guidò Teseo nella sfida con il terribile Minotauro, è una storia densa di significati e di simboli, di implicazioni magiche e misteriose; da allora, la figura del labirinto si è presentata sulla scena della storia del pensiero con una incredibile tenacia e persistenza, evolvendo come un organismo vivente. Dai graffiti rupestri ai pavimenti di cattedrali, dalle raffigurazioni pompeiane ai giardini di Chantilly, dalla danza all'alchimia, dalla letteratura alla pittura, dalla filosofia al cinema sarebbe impresa davvero ardua dar conto di tutti gli innumerevoli labirinti comparsi, citati o rappresentati nelle varie discipline.

Anche l'universo dei giochi conosce molto bene la figura del labirinto: il gioco dell'oca, che ha nei suoi cerchi concentrici un andamento labirintico e non è governato da nessun'altra forza se non dal cieco capriccio dei dadi; gli intrichi dei giochi enigmistici, i molti giocattoli meccanici con palline che devono raggiungere un centro; il flipper medesimo, con le sue pareti e i suoi percorsi elettromeccanici. Ma è attraverso i videogame che il labirinto conosce un poderoso rilancio ludico. Tunnel da attraversare, stanze e corridoi da percorrere, travi, fantasmi e spettri da evitare, una lunga serie di prove da superare alla ricerca del "bandolo della matassa" o del "percorso della sopravvivenza". Non sempre lo scopo finale è l'uscita dal labirinto, spesso il fine del gioco è interno all'intreccio stesso e consiste nell'abilità di schivare le numerose minacce e gli insidiosi trabocchetti per non perire durante il vagabondaggio in meandri interminabili e totalizzare il punteggio più elevato.

Il grande successo internazionale del gioco a labirinto è stato decretato da "Pac Man", il più blasonato e sfruttato personaggio del video, l'ingordo cacciatore di vitamine, il capostipite indiscusso della serie dei Tesei elettronici. "Pac Man" è in assoluto il gioco elettronico che ha avuto più successo, tanto da creare una vera e propria mania collettiva: la "Pac Man fever" che ha prodotto magliette, visiere, biglietti d'auguri, palloncini, cuscini, cestini, thermos, pupazzi, carte da

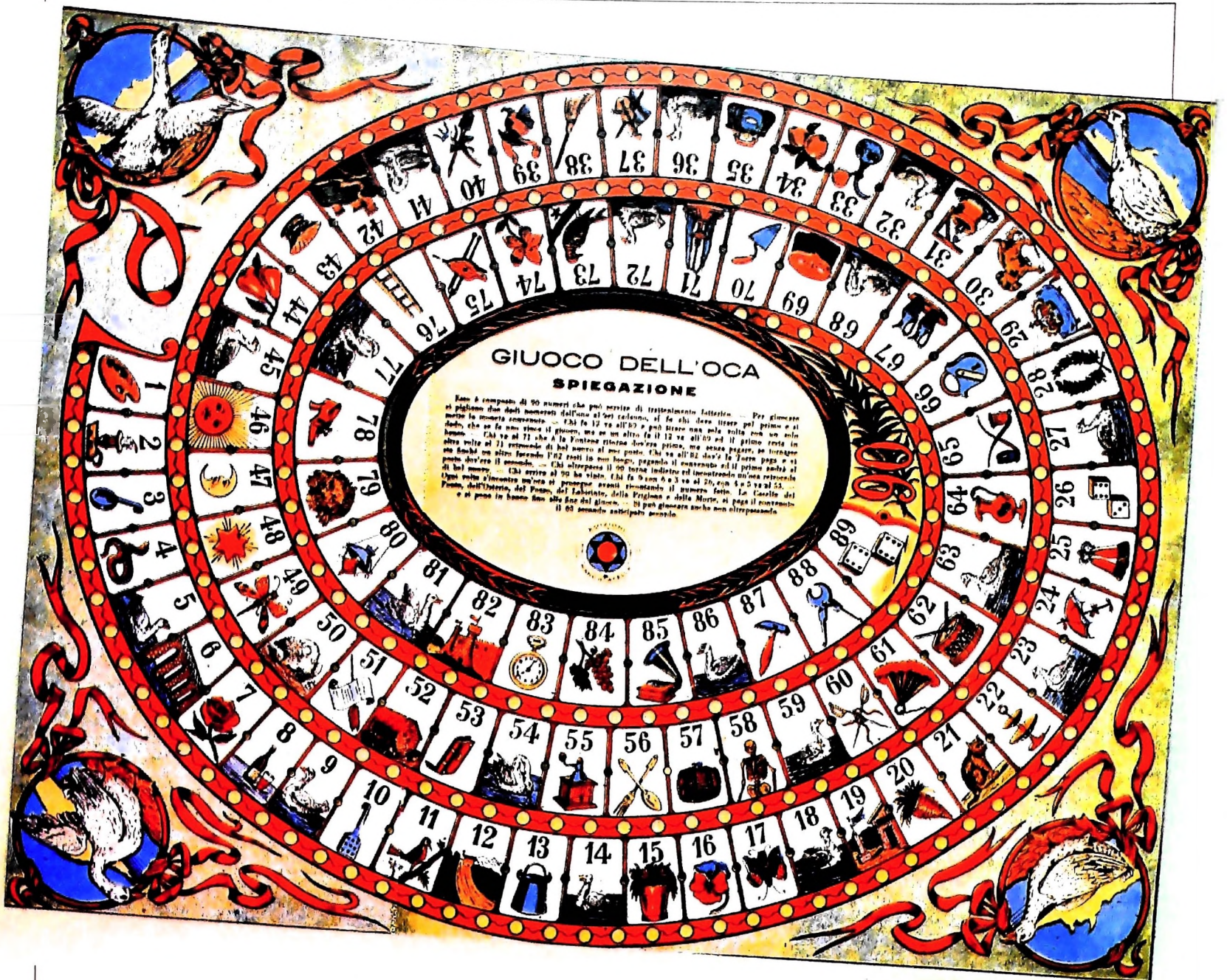
gioco e tanti altri gadget riproducti la celebre pallina gialla dalla bocca sempre spalancata. È persino nata una serie televisiva di cartoni animati, prodotta da Hanna e Barbera. E per rinnovare i fasti del più famoso mangiatore di pillole è stata anche realizzata la versione femminile, "Ms. Pac Man". Più aggraziata e smorfiosetta l'immagine, più complessi gli itinerari che seguono i terribili fantasmi (che, per l'occasione, sono stati "randomizzati", cioè affidati elettronicamente al caso) ma immutato l'obiettivo: ingoiare tutto ciò che Ms. Pac Man trova sul suo cammino: frutta, ciambelle, pillole d'energia. Un altro grande successo è stato "Burgertime": sullo schema fondamentale del labirinto è stata infatti disegnata una situazione curiosa: un cuoco muovendosi a perduto negli anfratti di una cucina deve confezionare dei burger, "raccolgendo" gli ingredienti classici: pane, insalata, pomodori e carne. Ma guai a offendere la già povera sapienza gastronomica del sandwich fumigante: il giocatore deve evitare wurstel, cetrioli, uova. Gioco che esalta la funzione esorcistica nei confronti della fame e disegna una prima normativa per il gusto gourmand.

"Happy Trails" mescola con rara sapienza il fascino del puzzle — ogni percorso va continuamente reinventato con delle tessere mobili, versione sofisticata del "gioco del 15" — con quello del labirinto: è un gioco metafora per evocare la situazione del garbuglio e dell'inestricabile.

Il labirinto non è un'espressione soltanto spaziale, bensì anche temporale. La durata del gioco in mezzo agli intrichi è sempre una funzione di due componenti: la complicatezza del disegno e il grado del nostro intuito. Sarà la fusione delle due componenti a poter fornire il tempo necessario per raggiungere la camera segreta (labirinto centripeto tipo "Lock'n Chase") o per tornare alla libertà (labirinto centrifugo tipo "Lady Bug"). L'importante è non perdersi in qualche vicolo cieco.

Il programma

Questo programma implementa la routine di base per costruire un gioco di labirinto; permette infatti la costruzione del labirinto (ovviamente molto semplice date le dimensioni dello schermo dell'M10) ed il movimento, nei suoi meandri, della preda (che è pilotata coi tasti "A", "D", "W", "Z" dal giocatore) e dall'inseguitore.



Il gioco del labirinto è l'eterna metafora della vita, e come la vita è perciò ricco di ostacoli, imprevisti, rinvii ed anche, qualche volta, d'un po' di fortuna. Anche il vecchio gioco dell'oca è, in fondo, un labirinto concentrico dominato dal Caso. Non stupisce quindi che nel fantastico mondo dei videogame la dinamica del gioco prenda in prestito parecchi elementi dalle enigmatiche situazioni dei classici dedali.

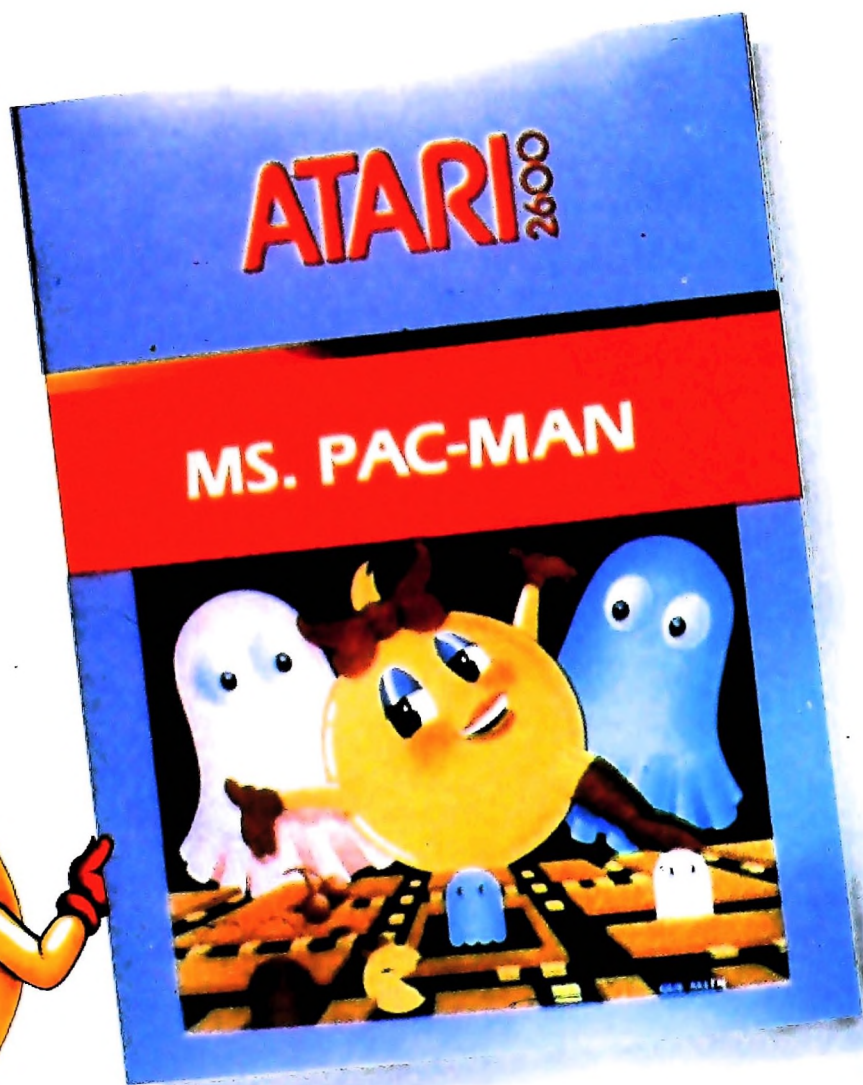



```

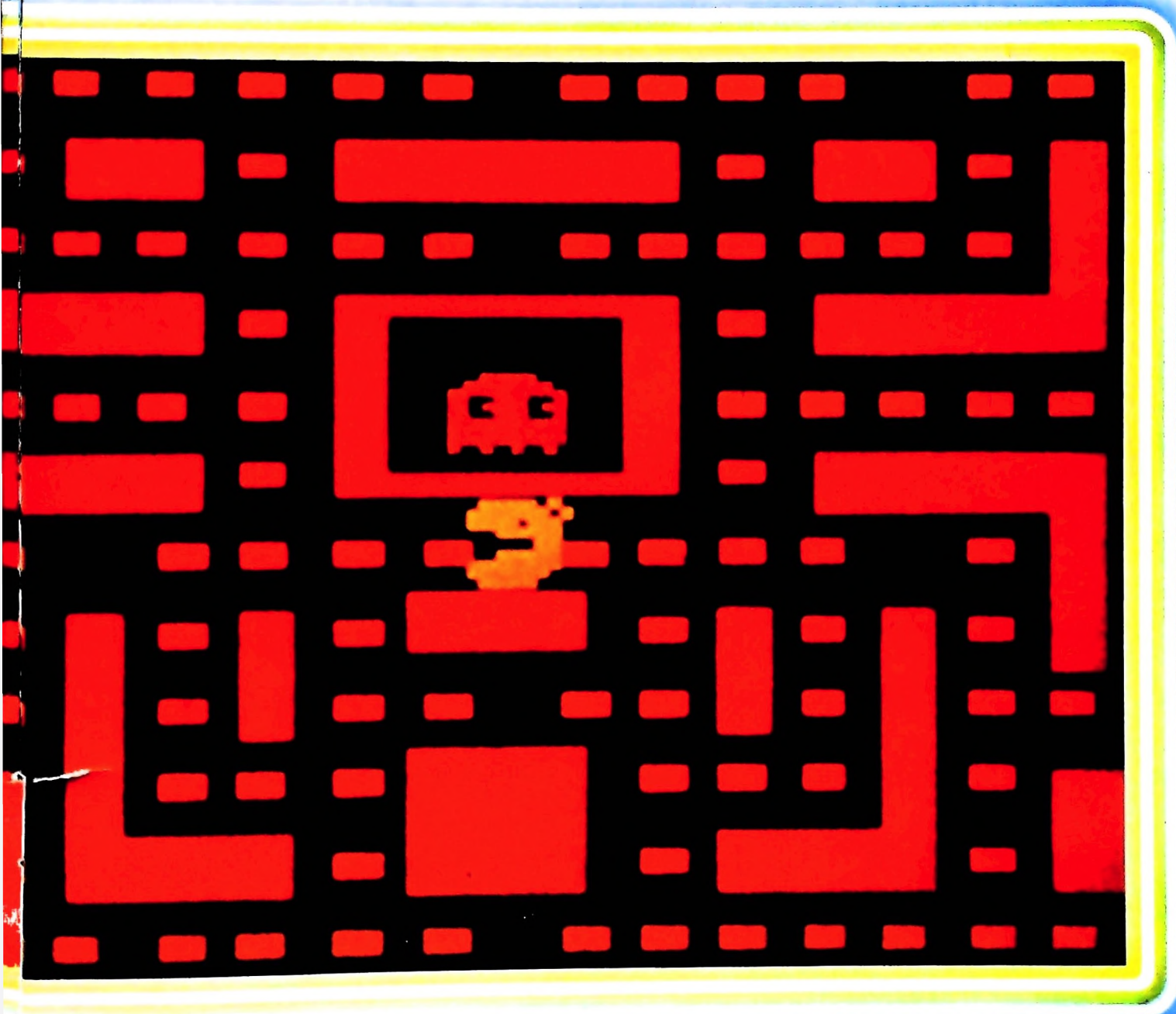
1   REM Esempio base di LABIRINTO
20  ESC$ = CHR$(27) : te = 0
30  ax = 2 : ay = 2 : cx = 29 : cy = 6
50  acn$ = "A" : cch$ = "O"
80  dx = 1 : dy = 0 : nd = 1
100 DIM la%(7,30)
110 FOR i=1 TO 30
120  la%(1,1) = 1 : la%(3,1) = 1
130  la%(5,1) = 1 : la%(7,1) = 1
140 NEXT i
150 la%(2,1) = 1 : la%(2,30) = 1
160 la%(4,1) = 1 : la%(4,30) = 1
170 la%(6,1) = 1 : la%(6,30) = 1
180 la%(3,2) = 0 : la%(3,29) = 0
190 la%(5,2) = 0 : la%(5,29) = 0
900 GOTO 9000 : REM inizia il programma
1000 REM crea il disegno sul video

```

Ecco qui la versione femminile di Pac Man: si tratta di Ms. Pac Man, prodotta dall'Atari. Il meccanismo del gioco è sempre quello (ad "ingoio"), ma questa signora Pac Man è senz'altro più svolazzante e irrequieta del suo compagno!




```
1005 CLS : PRINT
1010 FOR i=1 TO 7
1015 PRINT " ";
1020 FOR j=1 TO 30
1030 IF la%(1,j) = 0 THEN GOTO 1045
1040 PRINT "*"; : GOTO 1050
1045 PRINT " ";
1050 NEXT j
1060 IF i <> 7 THEN PRINT
1070 NEXT i
1075 ch$ = cch$
1080 x = cx : y = cy : GOSUB 8000
1090 ch$ = ach$
1100 x = ax : y = ay : GOSUB 8000
1110 RETURN
2000 REM esegue le azioni corrispondenti al tasto premuto
2010 IF kv$ = "A" THEN GOTO 2100 : REM a sinistra
```




```

2020 IF ky$ = "Z" THEN GOTO 2200 : REM in alto
2030 IF ky$ = "D" THEN GOTO 2300 : REM a destra
2040 IF ky$ = "W" THEN GOTO 2400 : REM in basso
2090 GOTO 2999
2100 REM premuto tasto "A"
2110 tx = cx - 1 : ty = cy
2199 GOTO 2900
2200 REM premuto tasto "Z"
2210 tx = cx : ty = cy - 1
2299 GOTO 2900
2300 REM premuto tasto "D"
2310 tx = cx + 1 : ty = cy
2399 GOTO 2900
2400 REM premuto tasto "W"
2410 tx = cx : ty = cy + 1
2900 IF la%(ty,tx) = 1 THEN GOTO 2999
2905 ch$ = " "
2910 x = cx : y = cy : GOSUB 8000
2920 ch$ = cch$
2930 x = tx : y = ty : GOSUB 8000
2940 cx = tx : cy = ty
2999 RETURN
3000 REM non e' stato premuto nessun tasto
3005 te = te+1 : IF te MOD 50 <> 0 THEN GOTO 3500
3010 tx = ax + dx : ty = ay + dy
3020 IF la%(ty,tx) = 0 THEN GOTO 3400
3030 GOSUB 7000
3035 tx = ax + dx : ty = ay + dy
3040 IF la%(ty,tx) = 1 THEN GOTO 3030
3400 ch$ = " "
3410 x = ax : y = ay : GOSUB 8000
3420 ch$ = ach$
3430 x = tx : y = ty : GOSUB 8000
3440 ax = tx : ay = ty
3500 RETURN
7000 REM cambia la direzione
7010 nd = INT(RND(te)*4) + 1
7020 IF nd > 4 THEN nd = 1
7030 ON nd GOTO 7050,7060,7070,7080
7050 dx = 1 : dy = 0 : GOTO 7100
7060 dx = -1 : dy = 0 : GOTO 7100
7070 dx = 0 : dy = 1 : GOTO 7100
7080 dx = 0 : dy = -1
7100 RETURN
8000 REM scrive "ch$" in (x,y)
8010 IF x < 0 OR x > 39 THEN GOTO 8100
8020 IF y < 0 OR y > 7 THEN GOTO 8100
8030 PRINT ESC$+"Y"+CHR$(y+32)+CHR$(x+32);
8040 PRINT ch$;
8100 RETURN
9000 REM programma principale
9010 GOSUB 1000
9030 ky$ = INKEY$
9040 IF ky$ = "" THEN GOSUB 3000 ELSE GOSUB 2000
9050 GOTO 9030
9999 END.

```


Lezione 53

La robustezza dei programmi

Nella precedente lezione abbiamo esaminato il concetto di "robustezza" di un programma, segnalando che, per far fronte con controlli e messaggi d'errore a un uso scorretto del programma, è necessario leggere i valori numerici attesi dal programma come stringhe di caratteri, ed effettuare su di esse controlli e conversioni. Avevamo assunto come esempio per effettuare interventi di "irrobustimento" un programma per calcolare il numero di piastrelle necessarie per rivestire una stanza.

Il primo problema che ci troviamo di fronte, quindi, è quello di costruire un sottoprogramma (utilizzabile più volte anche in altri programmi), che, data una stringa da interpretare come un valore numerico (eventualmente con parte decimale), ne controlli la correttezza e, se possibile, fornisca il valore numerico corrispondente.

Supponiamo che tale sottoprogramma riceva la stringa da controllare nella variabile stringa A\$ e fornisca il risultato nella variabile A, indicando nella variabile R la presenza di un errore. Nel programma useremo un certo insieme di funzioni capaci di operare sulle stringhe di caratteri, e precisamente:

- la funzione LEN: accetta come argomento una stringa, e ne fornisce la lunghezza, ovvero il numero di caratteri che la compongono; per esempio, l'istruzione

```
10 PRINT LEN("STRINGA")
```

visualizza 7, numero di caratteri che compone "STRINGA";

- la funzione LEFT\$: essa, applicata a una stringa, estrae un certo numero di caratteri a partire da sinistra; per esempio, l'istruzione

```
10 PRINT LEFT$("STRINGA", 3)
```

visualizza i primi tre caratteri da sinistra di "STRINGA" e cioè "STR";

- la funzione RIGHT\$: estrae il numero indicato di caratteri a partire dalla destra della stringa, cioè gli ultimi n caratteri della stessa; per esempio, l'istruzione

```
10 PRINT RIGHT$("STRINGA", 3)
```

visualizza gli ultimi tre caratteri di "STRINGA", e cioè "NGA";

- la funzione VAL: essa, applicata a una stringa di caratteri numerici, ci fornisce il corrispondente valore: per esempio visualizza il valore 1.
Potremo allora costruire il sottoprogramma come segue:

```
1000 'Accetta una stringa in A$ come numero
1002 ' Se e' corretta, A=valore e R=0
```



```

1004 ' Altrimenti A=0 e R=1
1010 L9=LEN(A$) 'Lung. di A$ in L9
1015 A=0 'Val. iniz. del risult
1020 I9=0 'Val. iniz. parte intera
1025 D9=0 'Val. iniz. parte decim.
1030 S9=1 'Segno: inizialm. +
1031 'Scarta blank iniziali
1032 'While primo crt=" " do
1033 Z$=LEFT$(A$,1) 'Primo crt in Z$
1034 IF Z$=" " THEN A$=RIGHT$(A$,L9-1):L9=L9-1
      :GOTO 1032
1035 'Endwhile
1036 'Cerca + o - iniz e registra segno
1037 IF Z$="+" THEN A$=RIGHT$(A$,L9-1):L9=L9-1:
      GOTO 1040
1038 IF Z$="-" THEN S9=-1:A$=RIGHT$(A$,L9-1):
      L9=L9-1 'Segno -
1040 'While ci sono crt(">".) do
1050 Z$=LEFT$(A$,1):A$=RIGHT$(A$,L9-1):L9=L9-1
1060 IF Z$=" " THEN 1120 'Elimina blank
1085 IF Z$="." THEN 1130 'A calcolo decimale
1090 IF Z$<="9" AND Z$>="0" THEN I9=I9*10+VAL(
      Z$):GOTO 1120 'Costruz. valore
1100 R=1: GOTO 1230 'Errore
1120 IF L9>0 THEN 1040
1125 'Endwhile
1130 'Calcolo parte decimale
1150 L9=LEN(A$) 'Lung. parte decim.
1155 E9=1 'Posiz. decimale
1157 'While ci sono crt do
1160 IF L9=0 THEN 1210
1170 LET Z$=LEFT$(A$,1):A$=RIGHT$(A$,L9-1):L9
      =L9-1
1175 IF Z$=" " GOTO 1200 'Elimina blank
1180 IF Z$<"0" OR Z$>"9" THEN R=1:GOTO 1230
      'Errore
1190 LET D9=D9+VAL(Z$)/10^E9:E9=E9+1 'Calcolo e
      increm. posiz. decimale
1200 GOTO 1160
1205 'Endwhile
1210 LET A=S9*(I9+D9) 'Risultato finale
1220 LET R=0
1230 RETURN

```

Il sottoprogramma è organizzato come segue:

- le linee fino alla 1030 hanno lo scopo di inizializzare le variabili usate (si noti che queste hanno nomi non mnemonici, e terminanti con 9, in modo da evitare il più possibile di collidere con eventuali nomi usati da un utente del sottoprogramma nel suo programma principale; fanno eccezione le variabili di "interfaccia" A e R); il valore verrà costruito calcolando separatamente la parte intera (se presente), la parte decimale (se presente) e tenendo conto del segno; le istruzioni 1210 e 1220 costruiranno il risultato finale, in caso di esito positivo della scansione;

- il ciclo da 1032 a 1035 elimina eventuali caratteri “ ” presenti all’inizio della stringa;
- le linee da 1036 a 1038 controllano se il primo carattere significativo sia il segno “ + ” o “ - ”; in quest’ultimo caso registrano un 1 negativo in S9;
- le linee da 1040 a 1125 valutano la parte intera: esse ignorano 1 “ ”, rinviano al calcolo della parte decimale appena viene incontrato il “.”, segnalano errore per caratteri non ammessi; nel caso in cui tutto vada bene, calcolano il valore della parte intera: se per esempio si incontra il numero 123, al primo ciclo di iterazione (con 19 inizialmente a 0), a 19 viene assegnato il valore $0*10+1$, cioè 1; alla seconda iterazione, operando cioè sulla seconda cifra, a 19 viene assegnato il valore $1*10+2$, cioè 12; al terzo giro, $12*10+3$, cioè 123;
- le linee dalla 1130 alla 1205 calcolano il valore della parte decimale; il procedimento è simile a quello per la parte intera: qui è necessario tenere conto che ogni cifra che si incontra ha un peso pari a $1/10$, $1/100$, $1/1000$, e così via, e cioè $1/10^E9$, ove E9 contiene la posizione della cifra dopo il punto decimale.

Prima ancora di inserire il sottoprogramma nel nostro programma varrà la pena di provarlo: costruiamo allora un programma principale allo scopo, come il seguente:

```
10 'Prova
20 LINEINPUT A$
25 GOSUB 1000
30 PRINT "Esito:"; R, "Valore:"; A
40 GOTO 20
```

La sua esecuzione permette di verificare i vari casi significativi; l’istruzione LINEINPUT permette di leggere nella variabile A\$ TUTTA LA STRINGA che viene fornita sulla linea, senza tenere conto della presenza di eventuali separatori, come il carattere “,”.

Prima di iniziare le prove con il sottoprogramma definiamo quali prove fare: allo scopo costruiamo una CHECKLIST, cioè una lista di tutti i valori che dovremo verificare per convincerci del corretto comportamento del sottoprogramma. Alcune prove saranno POSITIVE, cioè per la verifica del corretto funzionamento del programma a fronte di un suo corretto uso; altre saranno NEGATIVE, per verificare che il programma si comporti correttamente anche se lo usiamo in modo scorretto.

Ciò corrisponde a una buona pratica di sviluppo di programmi.

L’insieme delle prove che possiamo ritenere necessarie è il seguente:

- un valore con sola parte intera e uno con sola parte decimale
- un valore con parte intera e parte decimale
- un valore con il segno “ + ” davanti, e un disegno
- un valore con spazi inframmischiati
- un valore con caratteri frammischiati (prova negativa)
- un valore con più “.” decimali
- un valore con la “,” come separatore
- un valore con più segni

Quindi:

```
numero? 1
Esito: 0      Valore: 1
```



```

numero? 12345
Esito: 0      Valore: 12345

numero? +13
Esito: 0      Valore: 13

numero? -35
Esito: 0      Valore: -35

numero? 123.456
Esito: 0      Valore: 123.456

numero? .1
Esito: 0      Valore: .1

numero? 1 2 3
Esito: 0      Valore: 123

numero? 1 2 3 . 5
Esito: 0      Valore: 123.5

numero? A
Esito: 1      Valore: 0

numero? 1.1Q
Esito: 1      Valore: 0

numero? 1.1.1
Esito: 1      Valore: 0

numero? 1,2
Esito: 1      Valore: 0

numero? +-34
Esito: 1      Valore: 0

numero? +3.-4
Esito: 1      Valore: 0

```

Il nostro sottoprogramma si comporta bene. Nella prossima lezione integreremo il sottoprogramma con il programma principale, per vedere tutto l'insieme in azione.

Cosa abbiamo imparato

In questa lezione abbiamo visto:

- la funzione `LEN (A$)`, che fornisce il numero di caratteri di cui `A$` è composta;
- la funzione `LEFT$ (A$,I)`, che fornisce i primi `I` caratteri da sinistra di `A$`;
- la funzione `RIGHT$ (A$,I)`, che fornisce gli ultimi `I` caratteri a destra della stringa `A$`;
- la funzione `VAL`, che applicata a una stringa di cifre, ci fornisce il corrispondente valore numerico;
- il concetto di `CHECKLIST`, come lista di tutte le prove da effettuare per convincerci che un programma funziona correttamente;
- il concetto di prova `POSITIVA` e prova `NEGATIVA`, come prove per verificare il corretto funzionamento di un programma a fronte, rispettivamente, di un suo uso corretto o scorretto;
- l'istruzione `LINEINPUT`, che legge una stringa di una intera riga, senza tenere conto dei caratteri di separazione.

LE TRASFORMAZIONI PROSPETTICHE

Con l'algebra delle matrici si realizza un sistema software che fornisce la visualizzazione prospettica dell'oggetto.

Facciamo innanzitutto un'importante classificazione delle proiezioni prospettiche in base al numero dei punti di fuga presenti, ovvero dei punti in cui convergono le linee parallele della figura originale. Le prospettive a un punto di fuga vengono anche chiamate prospettive centrali, mentre quelle a due e tre punti di fuga vengono chiamate prospettive accidentali. Il numero dei punti di fuga corrisponde al numero di intersezioni che il piano di proiezione forma con gli assi del sistema di riferimento associato alla figura originale (chiamato anche riferimento principale). Nella pagina seguente riportiamo una classificazione dei vari tipi di proiezioni geometriche piane che aiuta a creare un quadro di riferimento per questo problema generale.

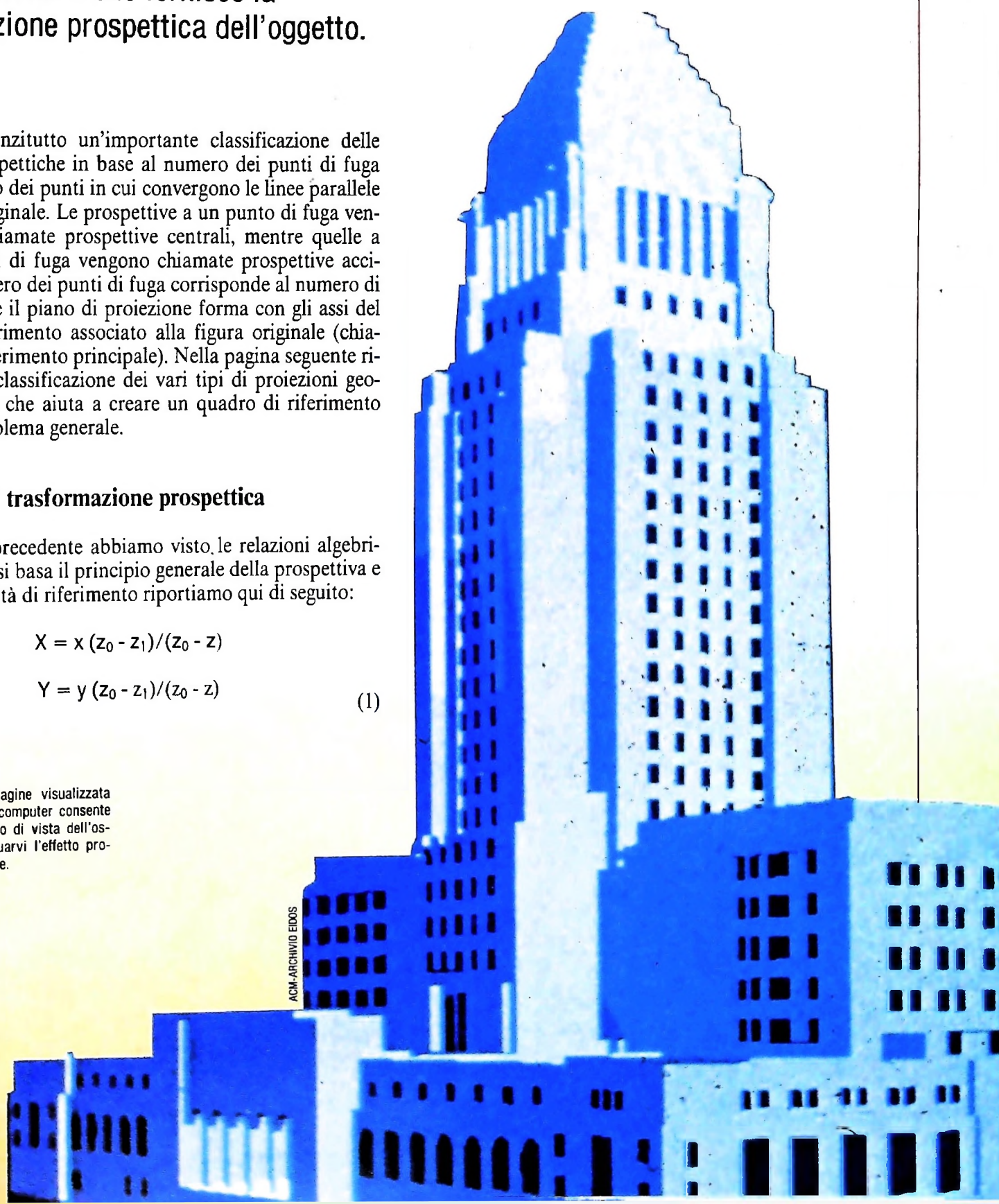
La matrice di trasformazione prospettica

Nella lezione precedente abbiamo visto, le relazioni algebriche sulle quali si basa il principio generale della prospettiva e che per comodità di riferimento riportiamo qui di seguito:

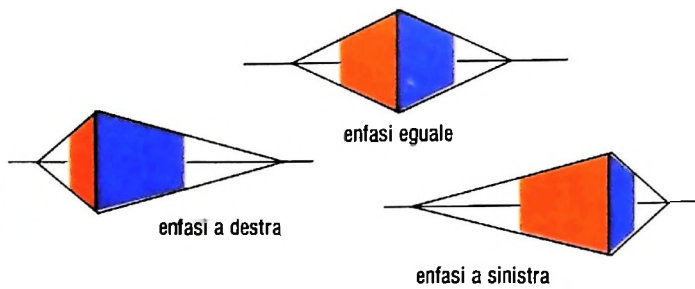
$$\begin{aligned} X &= x(z_0 - z_1)/(z_0 - z) \\ Y &= y(z_0 - z_1)/(z_0 - z) \end{aligned} \quad (1)$$

Un esempio di immagine visualizzata prospetticamente. Il computer consente di modificare il punto di vista dell'osservatore e di adeguarvi l'effetto prospettico dell'immagine.

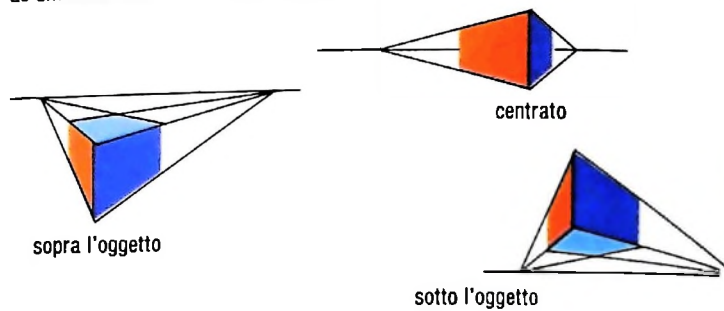
ACM-ARCHIVIO EDDOS



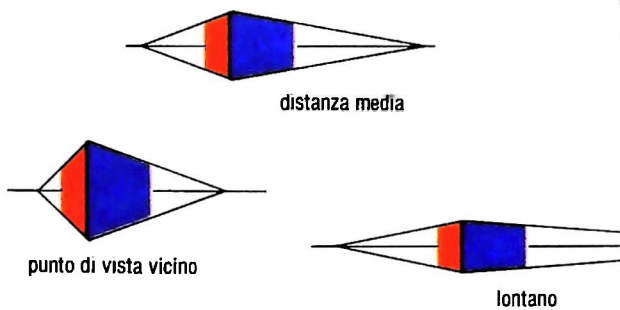
I diversi orientamenti del piano di proiezione e della posizione del punto di vista



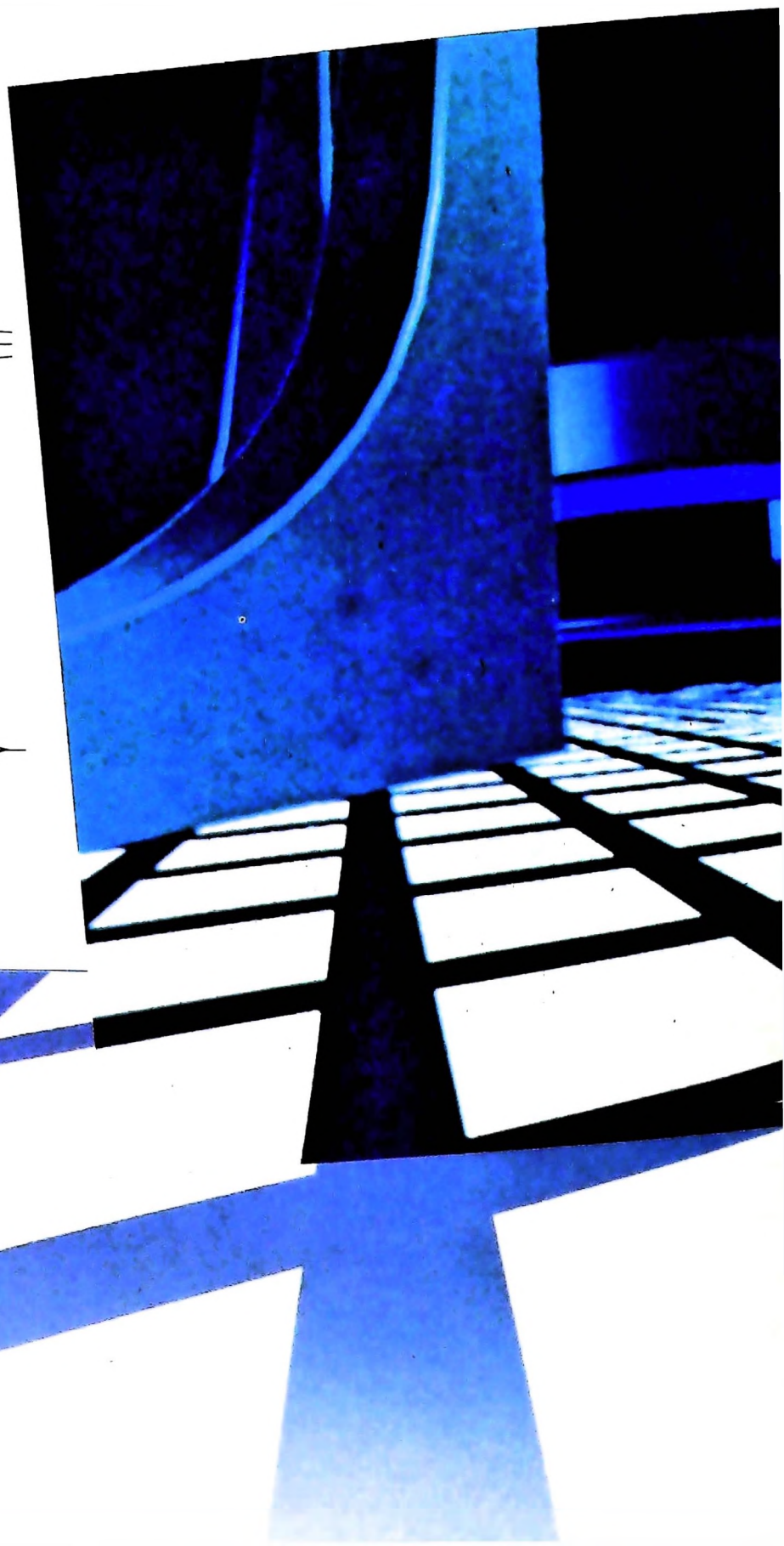
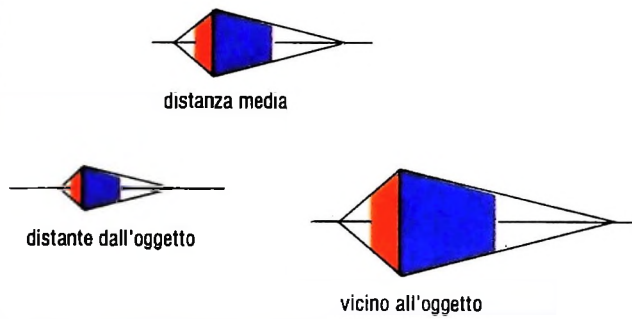
Le differenti altezze del punto di vista



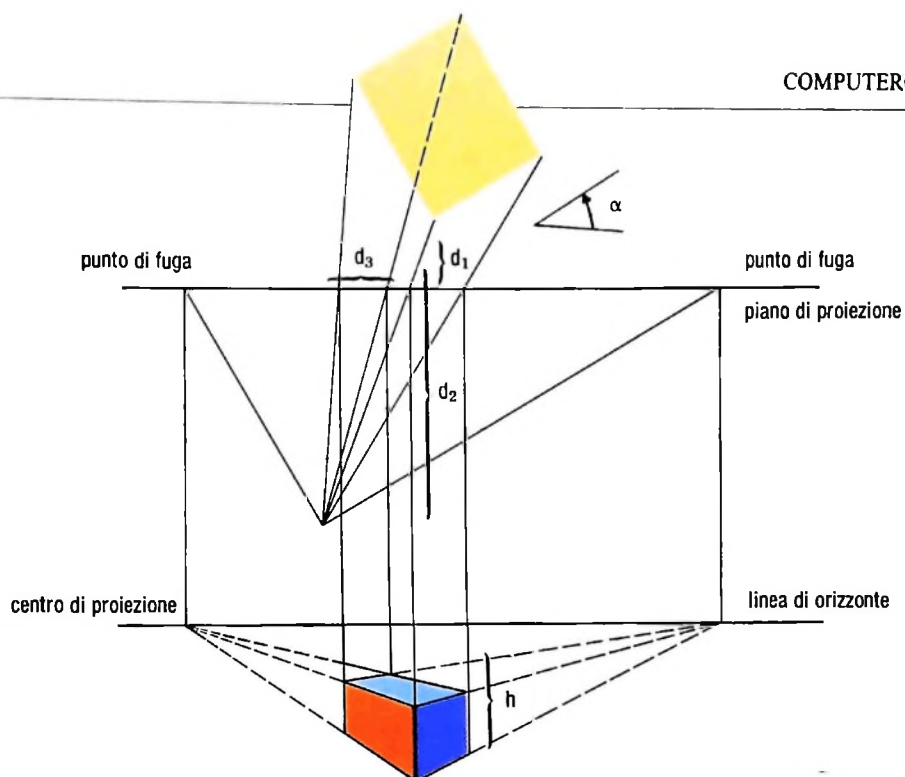
L'effetto della distanza del punto di vista



L'effetto della distanza del piano di proiezione



Sopra: lo schema illustra l'influenza dei diversi parametri sulla restituzione prospettica con due punti di fuga. A lato: un altro esempio di immagine visualizzata prospettivamente. Le tecniche prospettiche sono di grande utilizzo per la simulazione di architetture di interni.



Questo disegno rappresenta la costruzione di una proiezione accidentale. La corretta lettura dello schema impone di considerare due punti di vista: uno per la lettura del parallelepipedo e l'altra per l'esatta interpretazione del rettangolo che va letto come la faccia superiore del parallelepipedo stesso.

ACM-ARCHIVIO EIDOS

Consideriamo ora una proiezione prospettica centrale, tale cioè che il piano di proiezione sia ortogonale all'asse z e coincida inoltre col piano xy. Le relazioni (1) possono essere ricavate se supponiamo di rappresentare la trasformazione mediante la matrice 4x4:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Se $(x,y,z,1)$ sono le coordinate di un punto, le coordinate trasformate mediante la matrice P saranno: $(x,y,0,1-z/z_0)$, da cui ricaviamo le coordinate cartesiane del punto trasformato sul piano di proiezione: $X=x/(1-z/z_0)$, $Y=y/(1-z/z_0)$ che sono appunto le relazioni (1) qualora si ponga $z_1 = 0$, ovvero la distanza tra piano di proiezione e assi principali nulla.

Una proiezione prospettica non è soltanto il risultato della trasformazione prospettica descritta dalla matrice P; soprattutto se la rappresentazione deve avere un carattere di "fedeltà" all'oggetto originale, essa sarà il risultato della composizione di numerose trasformazioni: rotazioni, traslazioni e trasformazioni di scala. Pertanto si può pensare di realizzare una visualizzazione prospettica di una figura decidendo come deve essere ruotata rispetto al riferimento principale, come deve essere traslata, in che rapporto di scala deve venir prodotta la visualizzazione e infine effettuando la proiezione vera e propria applicando la trasformazione P. La trasforma-

zione, come si vede dalla matrice che la rappresenta, non si limita a generare i valori delle coordinate x e y trasformate, ma effettua anche lo "schiacciamento" della figura sul piano di proiezione, annullando la coordinata z della figura originale. Un problema interessante riguarda ancora l'ordine con cui le varie trasformazioni vengono eseguite e soprattutto se alcune delle trasformazioni possano essere eseguite dopo aver effettuato la proiezione. Questa possibilità è particolarmente interessante perché consente di ridurre la complessità di calcolo delle trasformazioni, che possono essere eseguite, anziché con quattro coordinate, solamente con le due coordinate x e y sul piano di proiezione. L'unica trasformazione che può venir trattata in questo modo è però la trasformazione di scala, che definisce il rapporto con cui si desidera riprodurre, sul video o sul foglio del plotter, l'immagine proiettata sul piano di proiezione; infatti la rotazione e la traslazione della figura originale devono essere eseguite prima, in quanto consentono di definire l'orientamento dell'oggetto rispetto al punto di vista. La matrice generale di trasformazione prospettica (a proiezione centrale) è la seguente:

$$\begin{bmatrix} A & B & C & 0 \\ D & E & F & 0 \\ G & H & I & -1/z_0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

dove $A, B, C, D, E, F, G, H, I$ rappresentano i fattori della rotazione. Come si vede questa trasformazione dipende da 7 parametri indipendenti (i tre angoli delle rotazioni, i tre incrementi di traslazione e il valore di z_0), indubbiamente troppi per la generazione di prospettive "fedeli". Si ricorre in genere ad una notevole riduzione del numero di questi parametri. I parametri significativi nella costruzione di una prospettiva accidentale fedele sono per esempio soltanto 5: l'angolo di rotazione dell'oggetto rispetto al riferimento principale attorno all'asse y , la distanza d_1 dell'oggetto dal piano di proiezione, la distanza d_2 del centro di proiezione dall'oggetto, lo spostamento d_3 del centro di proiezione lungo l'asse x , l'altezza h del centro di proiezione sul piano dell'oggetto. Questi 5 parametri vengono combinati, con alcuni passaggi che per brevità qui omettiamo, per costruire la matrice di trasformazione P_1 :

$$P_1 = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & r_2 \\ \sin\alpha & 0 & \cos\alpha & r_1 \\ -d_3 & -h & 0 & 1 \end{bmatrix}$$

dove r_1 e r_2 sono due parametri che dipendono da d_2 e dal seno dell'angolo tra gli assi principali e il piano di proiezione. Nel caso della prospettiva centrale, un punto di fuga tende all'infinito e il parametro r_1 è uguale a $-1/d_2$. Per trattare la prospettiva accidentale con tre punti di fuga occorrerà introdurre nella prima riga della quarta colonna della matrice un parametro r_3 , dipendente ancora dalla distanza d_2 e dall'angolo con cui il piano di proiezione è inclinato rispetto al piano xy del riferimento principale.

Per completare la visualizzazione prospettica è infine sufficiente trasformare le coordinate ottenute dall'applicazione della matrice P_1 mediante una proiezione parallela sul piano xy , ovvero "schiacciando" la figura mediante l'annullamento della coordinata z .

La trattazione svolta per la proiezione prospettica indica quali sono le caratteristiche delle proiezioni parallele piane, le quali dipenderanno ancora dalle posizioni relative del piano di proiezione e dell'oggetto, mentre saranno ovviamente indipendenti dalla distanza del punto di vista, il quale sarà stato portato all'infinito. Anche queste trasformazioni possono essere rappresentate da matrici a 4×4 elementi, e risulteranno dalla combinazione di rotazioni, traslazioni e matrici di proiezione sui differenti piani desiderati.

Aspetti computazionali

Come si è visto nella trattazione fatta, la possibilità di rappresentare le trasformazioni mediante l'algebra delle matrici presenta l'enorme vantaggio dell'estrema semplicità di programmazione. Infatti si può facilmente realizzare un sistema software che, prima di visualizzare un qualsiasi punto, applica su di esso la trasformazione desiderata. È evidentemente importante che un tale sistema consenta di definire interattivamente tutti i parametri della trasformazione desiderata, ma, nello stesso tempo, qualora si voglia una visualizzazione prospettica, ne fornisca automaticamente una, ricavando dalle coordinate originali dell'oggetto dei parametri standard per l'orientamento e la localizzazione del punto di vista. Per esempio, nel caso di visualizzazioni prospettiche di edifici in architettura, una regola semplice può essere quella di assumere, come altezza del punto di vista, 1,70 m (l'altezza degli occhi di una persona di media statura), e come distanza una misura che consenta di vedere gli spigoli estremi dell'edificio all'interno dell'angolo di visione naturale, che è compreso tra i 45 e i 60 gradi. Con una tale scelta si può ottenere, senza dover definire alcun parametro, una visione prospettica priva di distorsioni.

Un requisito che un sistema di trasformazioni deve possedere è indubbiamente quello della velocità di calcolo, sia per poter ottenere in breve tempo molteplici rappresentazioni dell'oggetto, sia, in alcuni casi, per poter realizzare effetti di animazione. In quest'ultimo caso la soluzione di realizzare via software il sistema di trasformazione non è la migliore; molto più efficace sarebbe poter disporre di un sistema di trasformazione hardware.

In ogni caso un notevole vantaggio in termini di complessità lo si raggiunge evitando il calcolo del prodotto matriciale. Infatti nel caso generale il prodotto fra un vettore di 4 componenti e una matrice 4×4 comporta 18 tra moltiplicazioni e divisioni e 12 somme; se programmiamo il sistema in modo da evitare le operazioni sulle coordinate che vengono trascurate alla fine (in particolare la z), si può realizzare un algoritmo che comporta soltanto 8 fra moltiplicazioni e divisioni e 5 somme. È evidente che per un grande numero di punti il vantaggio che si ottiene è notevole.

FINGEST (II)

Alcuni esempi di come si possano risolvere semplici problemi gestionali.

Problemi gestionali

A) Deprezzamento a quote costanti

Il Signor X possiede una macchina per scrivere del valore di lire 2 500 000. Supponendo di doverla utilizzare per 5 anni e ipotizzando un prezzo di vendita (al termine del quinquennio) di lire 800 000, il Signor X desidera conoscere la relativa tabella di deprezzamento.

DATI DA INSERIRE:

valore contabile attuale2 500 000
valore di realizzo800 000
vita (in anni).....5

B) Deprezzamento a rimanenza decrescente

Basandoci sul quesito precedente, si richiede la tabella di deprezzamento applicando il metodo a rimanenza decrescente, attribuendo un fattore d'accelerazione del 120%.

DATI DA INSERIRE:

valore contabile attuale2 500 000
vita (in anni).....5
fattore d'accelerazione.....120

C) Deprezzamento per somma dei singoli anni

Il quesito è lo stesso dei punti precedenti; in questo caso però nel conteggio viene utilizzato il metodo di somma dei singoli anni.

DATI DA INSERIRE: come punto A.

D) Punto di breakeven

Il Signor X acquista dei prodotti al prezzo di lire 2 000 cadauno ed intende rivenderli a lire 8 000.

Per il lancio pubblicitario si prevede di spendere lire 30 000 000. Si desidera conoscere quante unità deve vendere per raggiungere il pareggio tra costi e ricavi (punto di Break-even).

DATI DA INSERIRE:

costi fissi30 000 000
costo unitario2 000
prezzo di vendita8 000

E) Quantità economica di un ordinativo

Gli esperti in tecnica operativa hanno sviluppato un metodo che consente di calcolare la quantità di un dato articolo che risulti la più economica o vantaggiosa da ordinare.

Nel caso in esame, il Signor X, responsabile di una piccola azienda, vuole calcolare la quantità economica di un ordinativo di microprocessori utilizzati nelle piastre che la ditta produce.

Il fabbisogno annuo è di 100 000 chip con un costo d'immobilizzo annuo di lire 1 000 ciascuno e con un costo per ogni ordinativo di lire 90 000.

DATI DA INSERIRE:

spese per ordinativo.....90 000
fabbisogno annuo (unità).....100 000
costo d'immobilizzo unitario1 000

F) Commissione per un venditore

Il Signor X riceve uno stipendio di lire 1 300 000 e una commissione del 5% sulle vendite di computer da lui effettuate.

Quale sarà la commissione dovutagli il mese in cui ha realizzato un fatturato di 55 000 000?

DATI DA INSERIRE:

stipendio mensile1 300 000
volume vendite mensili.....55 000 000
commissione.....5

G) Stipendio con straordinario

Il Signor X lavora come programmatore e guadagna 40 000 lire all'ora.

Un mese, per rispettare le consegne, lavora 15 ore in più rispetto alle normali, e tali ore straordinarie gli vengono retribuite una volta e mezza la normale paga oraria.

Quanto sarà il suo stipendio globale?

DATI DA INSERIRE:

paga oraria.....40 000
fattore multipl. per ore straord.1.5
ore lavorative normali.....35
ore straordinarie15

```

3000 * *****
3001 M$=      "Gestionali"
3020 * *****
3030 CLS
      :PRINTQ14,A1$,"":M$:"":A2$
      :PRINT"
      :PRINTA1$;" A ":"A2$;" deprezzamento a quote costanti"
      :PRINTA1$;" B ":"A2$;" deprezz.a riman.decrescente"
      :PRINTA1$;" C ":"A2$;" deprez.per somma sing.anni"
      :PRINTA1$;" D ":"A2$;" punto di breakeven"
3040 PRINTA1$;" E ":"A2$;" q.ta econom.d'un ordinativo"

```



```

:PRINT " ";A1$;" seleziona - (0) per proseguire ";A2$;
3050 A$=INKEY$
:IFA$=""THEN3050
:IFA$="0"THEN3060ELSE3090
3060 CLS
:PRINTA1$;" G ";A2$;" commissione per un venditore"
:PRINTA1$;" H ";A2$;" stipendio con straordinario"
3070 PRINTA1$;" I ";A2$;" ritorno menu precedente"
:PRINT""
:PRINTTAB(14);A1$;" seleziona ";A2$;
3080 A$=INKEY$
:IFA$=""THEN3080
3090 ON(ASC(A$)-64)GOTO3500,3550,3600,3650,3700,3740,3750,3800,3000
3500 '*****
3501 'deprezz. a quote costanti
3502 '*****
3503 RESTORE3513
:N=3
:FORI=1TON
:READD$(I)
:NEXTI
:CLS
3504 X=1
:PRINTTAB(5);A1$;" deprezzamento a quote costanti ";A2$
:FORI=1TON
:X=X+40
:
:PRINT@X,D$(I);STRING$(26-LEN(D$(I)),".")
:NEXTI
3505 X=26
:FORI=1TON
:X=X+40
:PRINT@X,"";
:INPUTPH(I)
:NEXTI
3506 KH=(PH(1)-PH(2))/PH(3)
3507 CLS
:PRINTA1$;" ANNO          DEPREZZAM.          VALORE ";A2$
3508 FORI=1TOPH(3)
:PRINTTAB(2);I;TAB(9);
:PRINTUSING"#####";KH;
:PRINTTAB(23);
:PRINTUSING"#####";PH(1)-KH*I
3509 IF1/6(<)INT(I/6)THEN3511
3510 C$=INKEY$
:IFC$=""THEN3510ELSECLS
:PRINTA1$;" ANNO          DEPREZZAM.          VALORE ";A2$
3511 NEXTI
3512 C$=INKEY$
:IFC$=""THEN3512ELSE3030
3513 DATA"valore contabile attuale","valore di realizzo"
3514 DATA"durata (in anni)"
3550 '*****
3551 'deprezz. a rimanenza decrescente
3552 '*****
3553 RESTORE3564
:N=3
:FORI=1TON
:READD$(I)
:NEXTI
:CLS
3554 X=1
:PRINTTAB(0);A1$;" deprezzamento a rimanenza decrescente ";A2$
:FORI=1TON
:X=X+40
:
:PRINT@X,D$(I);STRING$(26-LEN(D$(I)),".")
:NEXTI
3555 X=26
:FORI=1TON
:X=X+40
:PRINT@X,"";
:INPUTPH(I)
:NEXTI
3556 CLS
:PRINTA1$;" ANNO          DEPREZZAM.          VALORE ";A2$
3557 FORI=1TOPH(2)
:KH=PH(1)*PH(3)/100/PH(2)
3558 PRINTTAB(2);I;TAB(9);
:PRINTUSING"#####";KH;
:PRINTTAB(23);
:PRINTUSING"#####";PH(1)-KH
3559 PH(1)=PH(1)-KH

```

```

3560 IF I/6 < INT(I/6) THEN 3562
3561 C$=INKEY$
      :IFC$="" THEN 3561 ELSE CLS
      :PRINT A1$;" ANNO      DEPREZZAM.      VALORE ";A2$
3562 NEXT I
3563 C$=INKEY$
      :IFC$="" THEN 3563 ELSE 3030
3564 DATA "valore contabile attuale","durata (in anni)"
3565 DATA "fattore di accel. (%)"
3600 '*****
3601 'deprez.per somma dei singoli anni
3602 '*****
3603 RESTORE 3615
      :N=3
      :FOR I=1 TO N
      :READ D$(I)
      :NEXT I
      :CLS
3604 X=1
      :PRINT TAB(0);A1$;" deprezzamento per somma singoli anni ";A2$
      :FOR I=1 TO N
      :X=X+40
      :
      :PRINT @X,D$(I);STRING$(26-LEN(D$(I)),".")
      :NEXT I
3605 X=26
      :FOR I=1 TO N
      :X=X+40
      :PRINT @X," ";
      :INPUT PH(I)
      :NEXT I
3606 KH=PH(1)-PH(2)
3607 CLS
      :PRINT A1$;" ANNO      DEPREZZAM.      VALORE ";A2$
3608 FOR J=1 TO PH(3)
      :K1H=KH*(PH(3)+1-I)/(PH(3)*(PH(3)+1)/2)
3609 PRINT TAB(2);I;TAB(4);
      :PRINT USING "#####";K1H;
      :PRINT TAB(23);
      :PRINT USING "#####";PH(1)-K1H
3610 PH(1)=PH(1)-K1H
3611 IF I/6 < INT(I/6) THEN 3613
3612 C$=INKEY$
      :IFC$="" THEN 3612 ELSE CLS
      :PRINT A1$;" ANNO      DEPREZZAM.      VALORE ";A2$
3613 NEXT J
3614 C$=INKEY$
      :IFC$="" THEN 3614 ELSE 3030
3615 DATA "valore contabile attuale","valore di realizzo"
3616 DATA "durata (in anni)"
3650 '*****
3651 ' punto di breakeven
3652 '*****
3653 RESTORE 3659
      :N=3
      :FOR I=1 TO N
      :READ D$(I)
      :NEXT I
      :CLS
3654 X=1
      :PRINT TAB(6);A1$;" punto di breakeven ";A2$
      :FOR I=1 TO N
      :X=X+40
      :
      :PRINT @X,D$(I);STRING$(26-LEN(D$(I)),".")
      :NEXT I
3655 X=26
      :FOR I=1 TO N
      :X=X+40
      :PRINT @X," ";
      :INPUT PH(I)
      :NEXT I
3656 KX=INT(PH(1)/(PH(3)-PH(2)))
3657 PRINT @200,A1$;" punto di breakeven ";A2$;
      :PRINT USING "#####";KX
3658 C$=INKEY$
      :IFC$="" THEN 3658 ELSE 3030
3659 DATA "costi fissi","costo unitario"
3660 DATA "prezzo di vendita"
3700 '*****
3701 ' q.ta economica di un ordinativo
3702 '*****
3703 RESTORE 3710

```



```

:N=3
:FORI=1TON
:READD$(I)
:NEXTI
:CLS
3704 X=1
:PRINTTAB(6);A1$;" q.ta economica ordinativo ";A2$
:FORI=1TON
:X=X+40
:
:PRINT@X,D$(I);STRING$(26-LEN(D$(I)),".")
:NEXTI
3705 X=26
:FORI=1TON
:X=X+40
:PRINT@X,"";
:INPUTPH(I)
:NEXTI
3706 KZ=INT(SQR(2*PH(1)*PH(2)/PH(3)))
3708 PRINT@200,A1$;" q.ta economica (unita') ";A2$;
:PRINTUSING"#####";KZ
3709 C$=INKEY$
:IFC$=""THEN3709ELSE3030
3710 DATA'spese per ordinativo","fabbisogno annuo unita'
3711 DATA'costo immobil.unitario"
3740 GOT03030
3750 '#####*
3751 ' commissione per un venditore
3752 '#####*
3753 RESTORE3759
:N=3
:FORI=1TON
:READD$(I)
:NEXTI
:CLS
3754 X=1
:PRINTTAB(6);A1$;" commissione per un venditore ";A2$
:FORI=1TON
:X=X+40
:
:PRINT@X,D$(I);STRING$(26-LEN(D$(I)),".")
:NEXTI
3755 X=26
:FORI=1TON
:X=X+40
:PRINT@X,"";
:INPUTPH(I)
:NEXTI
3757 PRINT@200,A1$;" competenza ";A2$;
:PRINTUSING"#####";PH(2)*PH(3)/100
:PRINT@240,A1$;"tot.a pagare (al netto stip.)";A2$;
:PRINTUSING"#####";PH(2)*PH(3)/100-PH(1);
3758 C$=INKEY$
:IFC$=""THEN3758ELSE3030
3759 DATA'stipendio mensile","volume vendite mensili"
3760 DATA'commissione (%)
3800 '#####*
3801 ' stipendio con straordinario
3802 '#####*
3803 RESTORE3808
:N=4
:FORI=1TON
:READD$(I)
:NEXTI
:CLS
3804 X=1
:PRINTTAB(6);A1$;" stipendio con straordinario ";A2$
:FORI=1TON
:X=X+40
:PRINT@X,D$(I);STRING$(26-LEN(D$(I)),".")
:NEXTI
3805 X=26
:FORI=1TON
:X=X+40
:PRINT@X,"";
:INPUTPH(I)
:NEXTI
3806 PRINT@200,A1$;" paga base ";A2$;
:PRINTUSING"#####";PH(1)*PH(3)
:PRINT@240,A1$;" straordinario ";A2$;
:PRINTUSING"#####";PH(1)*PH(2)*PH(4);
3807 C$=INKEY$
:IFC$=""THEN3807ELSE3030
3808 DATA'paga oraria","fatt.molt.per ore str."
3809 DATA'ore lav. normali","ore lav. straordinarie"

```


— UN NUOVO MODO DI USARE LA BANCA. —

Conto corrente
più

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

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

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

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

Inoltre un servizio utilissimo, soprattutto per imprenditori e commercianti denominato "esito incassi", consente di avere comunicazione dell'eventuale insolvenza entro solo cinque giorni dalla scadenza. Una opportunità veramente speciale.

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

Veniteci a trovare, ci conosceremo meglio.

 **BANCO DI ROMA**
CONOSCIAMOCI MEGLIO.



Olivetti M10 vuol dire disporre del proprio ufficio in una ventiquattrore. 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 microplotter 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 rivolgetevi al negozio contrassegna: Olivetti M10 Punto di Vendita*
o al distributore Olivetti & Olivetti, Div. Sicme Personal Computer, Via Meravigli 12,
20123 Milano

NOME COGNOME _____
 VIA _____
 CAP/CI/TÀ _____
 TELEFONO _____