

# MSX

# N 23

# DISK

£ 15.000

- Hole Golf
- Tracer
- Crystal
- Mortadelo
- Archivio Anagrafico
- Elettronica
- Finance Plus 6

giochi  
news  
utilità

L'UNICA DISK MAGAZINE  
DEDICATA ALLO STANDARD MSX



# MSX

## DISK

## SOMMARIO

- 2 Sommario – Sul disco  
Caricamento – Avvertenze
- 3 Editoriale – Abbonamenti
- 4 Hole Golf – Mortadelo
- 5 Tracer – Crystal
- 6 Archivio Anagrafico - Elettronica
- 7 Finance Plus 6
- 8 News Flash
- 9 Computer & Programmazione (Parte VI<sup>a</sup>)
- 10 Il Pascal (Parte IV<sup>a</sup>)
- 12 Dentro l'MSX (Parte V<sup>a</sup>)
- 13 Basic (Parte VIII<sup>a</sup>)

## SUL DISCO

- 1 Hole Golf
- 2 Tracer
- 3 Mortadelo
- 4 Crystal
- 5 Archivio Anagrafico
- 6 Elettronica
- 7 Finance Plus 6

## CARICAMENTO

A computer spento inserite il disco nel driver. Tenendo premuto il tasto CTRL accendete il computer e tenetelo inserito fino alla comparsa sul video del sommario. Per caricare un programma premete il numero corrispondente (dall'1 all'8). Il caricamento avverrà automaticamente.

## AVVERTENZE

Questo disco è stato registrato con cura e con i più alti standard di qualità. Leggete con attenzione le istruzioni per il caricamento. Nel caso in cui, per una ragione qualsiasi, trovaste difficoltà nel caricare i programmi, telefonate alla nostra redazione al numero (02) 89502256 oppure spedite il disco al seguente indirizzo:

**Gruppo Editoriale International Education srl - viale Famagosta, 75 - 20142 Milano.**

Testeremo il prodotto e, nel caso, lo sostituiamo con uno nuovo senza aggiunta di costi supplementari.

# EDITORIALE

**C**on un po' di fortuna siamo riusciti a sopravvivere e rieccoci di nuovo qui! Dopo tanti problemi occorsi e risolti, ci presentiamo nuovamente a voi utenti Msx come ultima spiaggia per questo standard abbandonato da tutti. Siamo in un nuovo anno e forse sarà il nostro ultimo, intanto noi cerchiamo di progredire comunque, certi del vostro fedele supporto. Altre nostre testate sono scomparse ormai dal mercato, vittime di una concorrenza sempre più varia e multiforme, ma Msx Disk resiste proprio grazie a voi. Continuate a credere in noi e non vi deluderemo! Non rimane altro da dire se non augurarvi buon lavoro e buon divertimento con questo nuovo numero di Msx Disk.

LA REDAZIONE



## ABBONAMENTI

### Comunicato importante

da oggi potrete abbonarvi alla rivista MSX DISK e riceverla comodamente a casa semplicemente sottoscrivendo uno speciale abbonamento per 10 numeri allo specialissimo prezzo di Lit. 128.000 invece di Lit. 140.000. Potrete così assicurarvi la vostra copia e risparmiare ben Lit. 12.000.

Desidero abbonarmi alla rivista MSX DISK allo speciale prezzo di Lit. 128.000 anziché Lit. 140.000 per 10 numeri.

COGNOME  NOME

VIA

CAP  CITTA'  PROV.

Allego    assegno     vaglia postale     intestato a **Gruppo Editoriale International Education.**

**Ritagliare e spedire a Gruppo Editoriale International Education – Viale Famagosta 75 – 20142 Milano.**

# HOLE GOLF



**T**ra tutti i programmi che simulano il gioco del golf, questo è sicuramente uno dei migliori da molti punti di vista. Innanzitutto è decisamente eccellente per l'effetto e per la potenza che si possono imprimere alla pallina quando si lancia, e grazie alle capacità grafiche dell'Msx 2, è anche molto curato nella grafica e nel disegno dei dettagli. Una volta caricato il gioco comparirà il messaggio WAIT che vi inviterà ad attendere alcuni secondi, dopodiché sullo schermo scorrerà la presentazione iniziale e quindi comparirà il menù del gioco. Tramite questo menù potrete selezionare nell'ordine: la capacità del giocatore, il tipo di torneo, il percorso. Per scegliere le diverse opzioni dovrete spostarvi in su e in giù sulle diverse opzioni premendo i tasti del cursore per muovere in alto o in basso, oppure muovendo in queste due direzioni la leva del joystick. Una volta sull'opzione da modificare dovrete usare i tasti dei cursori, o la leva del joystick, muovendo nelle direzioni orizzontali. Questo cambierà lo stato dell'opzione selezionata. Le opzioni sono tre:

**PLAYER:** Corrisponde alla capacità del giocatore e può essere: EXPERT = Esperto, PROFESSIONAL = Professionista o AVERAGE = Medio.

**GAME:** E' il tipo di torneo: 1-4 DAY TOURNAMENT = Torneo della durata di 1-4 giorni, STROKE PLAY o MATCH PLAY.

**COURSE:** Percorso del campo: EAST = Percorso ad Est oppure WEST = Percorso ad Ovest.

Per iniziare il gioco quando lo desiderate dovrete premere la barra spaziatrice. Lo spazio o il tasto fire del joystick devono essere usati anche per lanciare la pallina una volta selezionati potenza ed effetto da imprimirle.

## COMANDI

Tasti:

[CURSORI] = Movimenti / Scelte opzioni

[SPAZIO] = Lancio pallina / Inizio gioco

Joystick in porta 1

# MORTADELO



**Q**uesto è un gioco che sovverte ogni logica interattiva introducendo sul vostro Msx un simpatico personaggio di genere "fumettaro". Mortadelo è il nome del protagonista di questo gioco decisamente inconsueto. Il nostro Mortadelo vive solo in una grande casa invasa, non si sa come, da una flotta di fieri pollastri. Quale migliore idea di trasformarli in cena, pranzo e colazione?

Per questo, voi dovrete aiutare il nostro strano eroe a catturare i ruspanti amici per poi portarli, grazie all'ascensore, fino al terrazzo della casa dove, grazie all'amico cuoco, i polli salteranno in padella. Tutto qui! Sembra facile, ma non lo è affatto. Non dovrete fare altro che caricare il gioco e provare a catturare i ruspanti, che non sono mica tanto dei "pollastri".

Completato il caricamento del gioco grazie all'apposita opzione di Msx Disk, premendo la barra spaziatrice passerete dalla presentazione ai credits e poi al menù principale. Questo menù è composto da tre opzioni: la prima permette di scegliere tra tastiera e joystick, la seconda permette di ridefinire i tasti e la terza fa cominciare il gioco.

Per ridefinire i tasti una volta selezionata l'apposita opzione dovrete premere in sequenza i tasti scelti per le seguenti funzioni: Fuoco, Alto, Basso, Sinistra, Destra, Pausa.

Una volta finita la scelta dei tasti da usare per controllare il gioco, il programma torna al menù principale dopo alcuni istanti.

## COMANDI

Tasti:

[1] = Tastiera / Joystick

[2] = Ridefinizione tasti

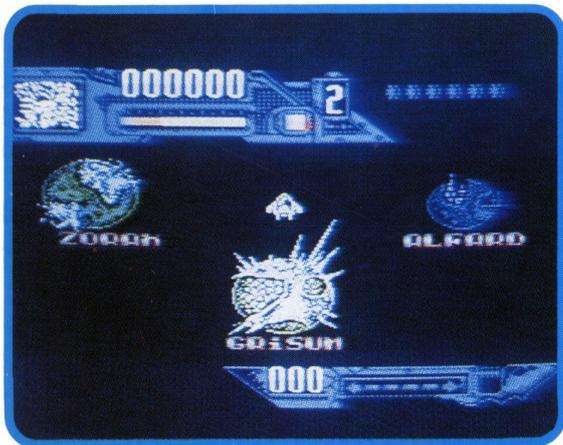
[0] = Inizio gioco

[CURSORI] = Movimenti

[SPAZIO] = Salto

Joystick in porta 1

# TRACER



**S**ulla traccia dei grandi giochi spaziali, Tracer offre una ricetta ben dosata di diversi classici del genere, da Defender a Skramble e così via. E' un tipo di spaziale a scorrimento orizzontale nelle due direzioni sinistra e destra, cioè lo schermo di gioco si sposta di qua e di là a seconda di dove muovete la vostra astronave.

Una volta caricato il gioco e dopo la presentazione grafica compariranno i credits del programma: a questo punto dovrete premere lo spazio per passare al menù principale. Questo menù sarà composto da quattro opzioni fondamentali corrispondenti, nell'ordine, ai tasti numerici dall'1

al 3 e al tasto numerico 0 (ultima opzione). Con le prime due opzioni si sceglie come controllare il gioco, cioè tastiera (1) o joystick (2). Usando la tastiera potete ridefinire i tasti di controllo a vostro piacimento tramite l'opzione 3. Per iniziare a giocare una volta scelti i controlli dovrete premere il tasto 0. Una volta cominciato il gioco compariranno sullo schermo tre pianeti tra cui scegliere la missione da affrontare: i primi due in alto sono accessibili subito e le due missioni corrispondenti devono essere necessariamente affrontate per poter passare all'ultimo pianeta che è quello in basso.

Per scegliere il pianeta, e quindi la missione, dovrete muovere l'astronave sul disegno corrispondente al pianeta scelto e quindi premere il tasto di fuoco. Dopo questa operazione vi troverete sul pianeta per affrontare la vostra misteriosa missione che vi porterà, infine, sul terzo pianeta abitato di quella galassia.

Scegliendo l'opzione di ridefinizione tasti dovrete premere, nell'ordine, i tasti corrispondenti alle seguenti direzioni: Sinistra, Destra, Alto, Basso, Fuoco, Reinizio, Pausa.

## COMANDI

Tasti:	[0] = Inizio gioco
[1] = Tastiera	[CURSORI] = Movimenti
[2] = Joystick	[SPAZIOA] = Fuoco
[3] = Ridefinizione tasti	Joystick in porta 1

# CRYSTAL



**E**cco un'altra storia tesa: per liberare la principessa, l'eroe della situazione si lancia alla ricerca dei preziosissimi cristalli necessari per sconfiggere il crudelissi-

mo signore dei ragni.

Ovviamente la nostra bella è prigioniera in un tetro castello popolato di simpatici amici con molte zampe e con il dente... avvelenato!

Sembra una cosa da poco, ma vedrete che correre su e giù inseguiti dai cattivissimi ragni non è poi il massimo!

Quando compaiono i credits - chi a fatto il programma, il titolo, etc. - potrete iniziare a giocare premendo il tasto di fuoco del joystick o la barra spaziatrice.

## COMANDI

[CURSORI] = Movimenti
[SPAZIO] = Inizio gioco / Fuoco
Joystick in porta 1

# ARCHIVIO ANAGRAFICO

**T**ra le ormai molte utility proposte, non poteva certo mancare un archivio di tipo anagrafico che potesse usare per crearvi una rubrica o semplicemente una lista di nominativi da gestire per le più svariate ragioni.

Il programma consente di inserire un numero massimo di nominativi dipendente dalla capacità dei dischi che usate; così un disco formattato a 360k conterrà meno dati di uno da 720k, ma nulla vieta di dividere l'archivio in più parti.

Comunque un disco da 360k è già sufficiente visto che permette di immagazzinare centinaia di nominativi.

Ogni nominativo corrisponde ad una scheda composta da 10 campi, cioè dieci dati informativi per ogni scheda.

Questi dati sono poi divisi in nome, cognome, etc.

I dati archiviati - le schede possono essere visualizzati e stampati su tabulato in diverse combinazioni.

Una volta caricato il programma, sullo schermo comparirà un messaggio che vi invita ad inserire un disco Archivio.

Questo deve essere un disco appositamente formattato

da usare solo per archiviare i dati tramite questo programma. E' bene fare ciò per evitare confusione, perdite di dati e soprattutto perchè sul disco di Msx Disk non c'è spazio sufficiente a contenere un archivio di questo tipo.

Premendo un tasto qualunque dopo aver inserito il disco archivio potrete accedere al seguente menù principale, in cui le opzioni corrispondono ai diversi tasti funzione:

- [F1] = Inserimento Nominativi;
- [F2] = Archiviazione Dati;
- [F3] = Visualizzazione e Stampa Dati;
- [F4] = Cancellazione Nominativo dall'archivio;
- [F5] = Ritorno in ambiente Basic.

Per uscire dal programma evitate di spegnere il computer poichè potreste perdere dei dati.

E' bene terminare l'uso del programma tramite l'ultima, apposita opzione.

# ELETRONICA



**C**hi studia elettronica, o chi se ne interessa come molti hobbisti, troverà decisamente interessante questo programma. Esso raccoglie alcune delle formule e procedure utilizzate nel calcolo della capacità dei condensatori, delle resistenze, dei circuiti LC, etc.

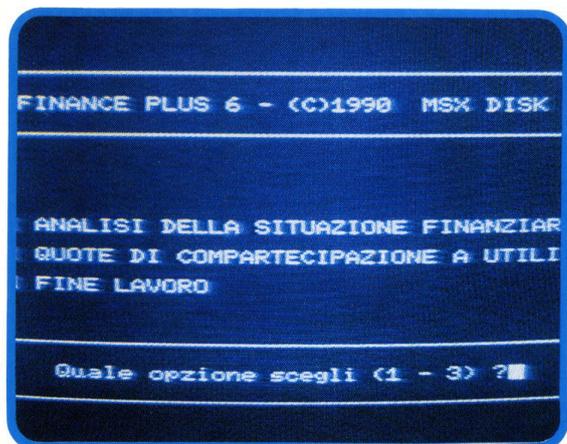
Una volta caricato il programma, vi verrà mostrato il menù

principale del programma che permette di scegliere tra le diverse possibilità di calcolo. Ad ogni calcolo di circuiti elettrici è associato un numero e per selezionare l'opzione desiderata sarà sufficiente premere il tasto numerico corrispondente.

Il menù del programma offre sette opzioni:

- [1] - CONDENSATORI
- [2] - RESISTENZE
- [3] - CALCOLO CIRCUITI LC
- [4] - CALCOLO CIRCUITI RC
- [5] - CALCOLO BOBINE
- [6] - CALCOLO RIFRAZIONE
- [7] - FINE LAVORO

Come è nostra consuetudine, l'opzione di fine lavoro riporta il sistema al Basic e potrete comodamente analizzare il programma usando l'istruzione LIST ed il tasto [STOP] per fermare lo scorrere delle linee sullo schermo. Ovviamente, è necessario conoscere le basi dell'elettronica teorica, ma se non si conoscono questo programma può rappresentare un valido supporto per l'apprendimento delle nozioni basilari di questa particolare ed affascinante materia.



**C**ome sempre, per chi segue questi programmi per la prima volta ricordiamo che essi hanno il duplice scopo di fornire un aiuto nella gestione delle proprie finanze e offrire uno spunto didattico ai principianti che entrano nel mondo della programmazione. Per utilizzare il programma sarà sufficiente premere il tasto [7] quando compare il menù principale di Msx Disk. Il file 7.BAS è il file sorgente pronto per voi, per essere esaminato.

Questa volta il programma è dedicato al metodo di deprezzamento accelerato e alla suddivisione proporzionale.

Il programma è, come sempre, diviso in tre parti e offre un menù principale con le seguenti opzioni:

- 1 - ANALISI DELLA SITUAZIONE FINANZIARIA;
- 2 - QUOTE DI COMPARTICIPAZIONE AGLI UTILI;
- 3 - RITORNO AL DOS.

Nel file sorgente potrete trovare che la prima opzione viene espletata dalla parte che va dalla linea 1000 alla linea 2840 mentre l'analisi degli investimenti comuni è compresa tra la linea 3000 e la linea 3999.

Invece che dalla linea 3000 come negli scorsi numeri, le routine di servizio trovano posto dalla linea 5000 in poi.

## ANALISI DELLA SITUAZIONE FINANZIARIA

Questo algoritmo esegue una serie di calcoli basati sui dati ricavati da una relazione finanziaria di una ditta e che possono risultare di estremo interesse per una persona che investa capitali.

Tali calcoli riguardano il profitto, la liquidità e la struttura del capitale della ditta. Si devono solo paragonare i tassi di una società con quelli di altre nello stesso campo di attività. Il confronto può anche essere fatto con i valori medi delle industrie operanti nello stesso campo.

Per usare questa parte del programma, è necessario introdurre il nominativo della ditta in esame, la data della dichiarazione finanziaria e i vari importi che compaiono sulla dichiarazione stessa.

E' inoltre necessario introdurre il numero dei titoli ordinari che non rientrano nel bilancio, il prezzo di mercato di una azione e i dividendi pagati per ogni azione.

**ESEMPIO:** Il signor Rossi ha intenzione di investire nei titoli ordinari emessi da un costruttore di computers. Dopo aver inserito i dati della dichiarazione finanziaria della ditta in esame e sapendo che ha emesso 7000 azioni ordinarie ad un prezzo di mercato di 1750 lire l'una e che agli azionisti verrà corrisposto un dividendo di 125 lire per azione, il programma indicherà i profitti, la liquidità, l'attività, l'indebitamento e l'equità.

## QUOTE DI COMPARTICIPAZIONE AGLI UTILI

Questo programma calcola la distribuzione delle quote di compartecipazione agli utili fino a 250 dipendenti. Alcuni piani di distribuzione non sono 'integrati' - cioè la quota distribuita ad ogni dipendente è esattamente proporzionale al suo stipendio. Se il compenso è pari al 5% del compenso totale di tutti i partecipanti, allora viene assegnato il 5% del contributo totale di quell'anno, e così via.

I piani integrati di distribuzione delle quote di compartecipazione sono meno diretti. In questo caso un livello di stipendio non più alto del minimo contrattuale viene scelto come il livello di integrazione. Ogni impiegato il cui stipendio supera il livello di integrazione percepisce una percentuale - non più del 7% - sulla differenza tra il suo stipendio e il livello di integrazione. Il rimanente contributo totale è distribuito proporzionalmente allo stipendio. Se la parte integrata dell'assegnazione totale supera la quota totale da distribuire, essa viene ridotta proporzionalmente. Se questo succede, i dipendenti i cui stipendi sono inferiori al livello di integrazione non ricevono niente.

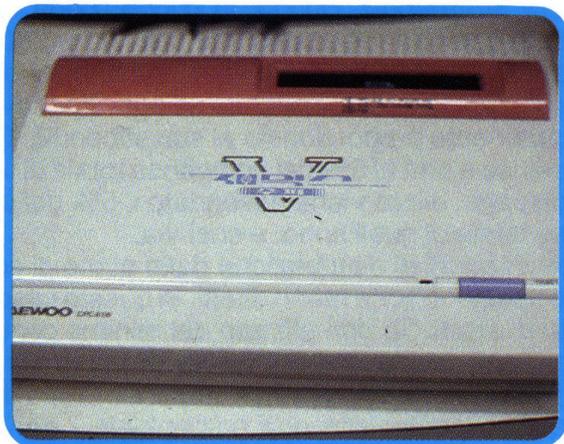
Questo programma tratta sia i piani integrati, che quelli non integrati fino a 250 partecipanti. Per prima cosa si introduce il nominativo e lo stipendio di ciascun impiegato partecipante. Gli importi degli stipendi vanno inseriti in migliaia di lire. Dopo aver inserito il nominativo e lo stipendio dell'ultimo impiegato, si deve inserire un nominativo a case, e -1 per lo stipendio per determinare l'introduzione dei dati quando il programma li richiede.

Il programma stampa il totale degli stipendi e l'usuale limite del 15% sui contributi. Allora si introduce il totale delle quote di compartecipazione come una frazione decimale del compenso totale. Viene richiesto se il piano è integrato, e se è così, quali sono il livello di integrazione e la percentuale.

Il programma mostra quindi una tabella che riporta il nominativo, lo stipendio di ogni impiegato e l'ammontare della sua assegnazione in migliaia di lire, divisa in parti integrate e non integrate; stampa il totale per tutti i dipendenti. Al termine viene data la possibilità di tornare indietro e cambiare una parte dei dati o tutti.

**E'** nato! Con un buon anticipo è cominciata sia in Giappone che in Europa dell'Msx 3 meglio noto come TURBO R. Del computer ancora non sappiamo molto e non si conoscono nemmeno i lineamenti generali di questo nuovo potente figlio dello standard Msx. Il computer, comunque, è prodotto dalla casa giapponese ASCII già nota agli utenti Msx per i numerosi prodotti software. In Europa il nuovo mostro viene importato da diverse case mentre ancora non è prevista la commercializzazione nel nostro paese.

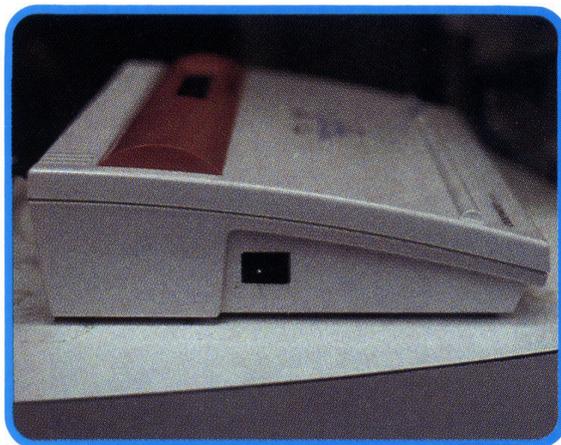
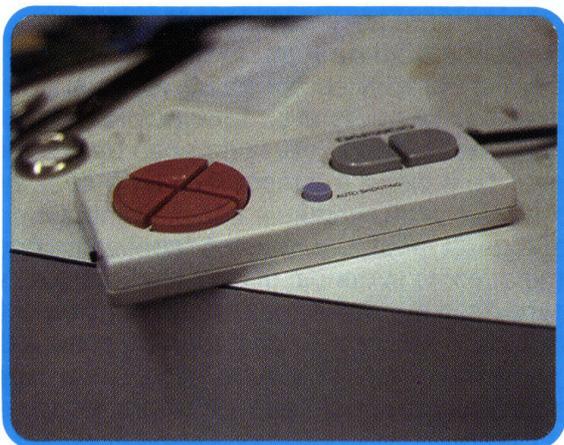
Se ne siete interessati potete scrivere alla: HSH Computervertrieb GmbH, Rombergstrasse 16, D-4715 Davensberg, West Germany - Tel. xx49-



2593/6168 - Fax. xx49-2593/7234, unico produttore europeo di Hard Disk e interfacce SCSI per il nostro standard.

Rivoluzione all'interno dell'MSX CLUB ITALIA. Dopo un tranquillo avvio iniziale, l'associazione di utenti Msx, che ormai conta centinaia di iscritti, ha dovuto riorganizzarsi per far fronte al grande numero di iscrizioni. E' stata una cosa imprevedibile ma che ha dato speranza a tutti gli utenti Msx, attivi più che mai.

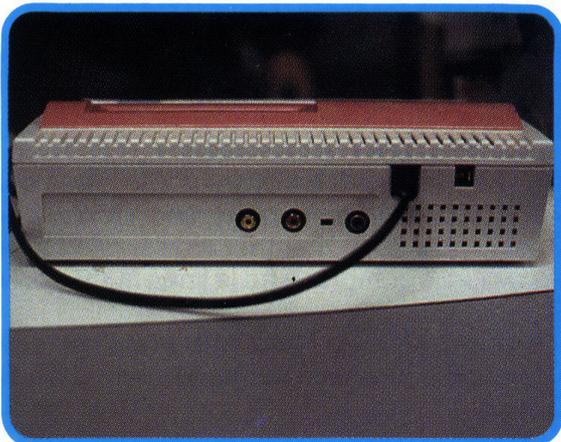
Il catalogo programmi è stato ampiamente arricchito con numerosissimi programmi per Msx 2 e utility di ogni tipo, ma purtroppo è stata abbandonata l'importazione di macchine e schede Msx per i grossi



problemi incontrati nell'approvvigionamento e nell'importazione.

Tra le novità offerte dall'MSX CLUB ITALIA c'è una linea diretta telefonica, attivata nel mese di Gennaio, che permette di contattare la direzione del Club. Per ora il servizio è disponibile un solo giorno alla settimana, ma al più presto sarà possibile contattare il Club in due appuntamenti settimanali. Di prossima uscita, inoltre, il primo numero dell'attesissimo MSX CLUB JOURNAL, che sarà ricco di trucchi, informazioni, e di tutto ciò che può rivelarsi utile all'utente Msx.

L'indirizzo del club è: MSX CLUB ITALIA - C.P. 34 - 20075 LODI CENTRO - MI.



Quell'aggeggio bianco e rosa che vedete nelle foto è un prodotto coreano destinato a far concorrenza alle console per videogiochi già affermate. Si tratta di un MSX 2 di cui si sfruttano la porta per i cartridge e le entrate dei joystick. Oltre a funzionare con le cartucce già commercializzate per i computer MSX, grazie ad uno speciale connettore è possibile collegare le cartucce del Nintendo.

Il costo è abbastanza limitato, pari a circa la metà di un Nintendo.

La macchina, prodotta dalla casa coreana DAEWOO è importata in Europa dall'MSX CENTRUM - Witte de Withstraat 27 - 1057 XG Amsterdam - Nederland - Tel. 020-167058 - Fax. 020-167058.

Eccoci giunti al nuovo appuntamento con il nostro corso di programmazione. Come nello scorso numero, continueremo a parlare di Programmazione o Notazione Lineare Strutturata, base di qualsiasi linguaggio per computer.

## ARRAY MULTIDIMENSIONALI

Nell'ultima lezione di programmazione abbiamo imparato che cosa sono gli array e abbiamo accennato al fatto che il tipo degli elementi di un array non è necessariamente il tipo degli interi e che deve pertanto essere specificato nella parte dichiarativa. In effetti conosciamo adesso un altro tipo, cioè gli array di interi. Consideriamo ora la seguente dichiarazione:

```
var A : array[1..N]of array[1..M]of integer
```

In altre parole A[I] diventa un array di M elementi e, se volessimo considerare lo J-esimo elemento di questi, dovremmo preoccuparci di scrivere: A[I][J]. In pratica la dichiarazione viene semplificata con la seguente:

```
var A : array[1..N,1..M]of integer
```

e la notazione A[I][J] diventa A[I,J]. In questo modo A[I,J], assume più chiaramente il significato di elemento all'incrocio della I-esima riga con la J-esima colonna di una matrice (o tabella) di N righe per M colonne. Si tratta dunque di un'estensione del concetto di array unidimensionale: le variabili hanno ora due indici (ordinati), e si dice che l'array è a due dimensioni. Le matrici costituiscono un comodo strumento per moltissime applicazioni, non solo di tipo numerico. ESEMPIO: se volessimo rappresentare lo stato di una scacchiera durante una partita a dama, potremmo usare il seguente metodo:

0= casella libera; 1= pedina bianca; 2= pedina nera; 3= dama bianca; 4= dama nera

la scacchiera potrebbe essere rappresentata da una matrice di 8X8 elementi come segue:

```
var SCACCHIERA : array[1..8,1..8]of integer;
```

Se volessimo pulire la scacchiera dovremmo scrivere un programmino del genere:

```
for I=1 to 8 do
for J=1 to 8 do SCACCHIERA[I,J]=0.
```

## INTRODUZIONE ALLA RICERCA E ALL'ORDINAMENTO

I problemi di ricerca e di ordinamento vestono un ruolo di particolare importanza in campo informatico. Il problema non è tanto quello di riuscire ad effettuare la ricerca o l'ordinamento quanto di farlo nel modo più opportuno e conveniente, vale a dire con il minor numero di confronti per l'esecutore.

### RICERCA

Il problema di ricerca può tranquillamente essere formulato in questo modo: avendo una sequenza A<sub>1</sub>,A<sub>2</sub>,...,A<sub>n</sub> di elementi, trovare se esiste, l'indice dell'elemento di valore K. Il metodo più semplice che ci potrebbe venire in mente, potrebbe essere il seguente:

```
program RICERCA-SEMPLICE,"metodo sequenziale".
N=...
var A : array[1..N]of integer;
leggi K
I=1;
repeat
if A[I]=K then exit;
I=I+1
until I>N;
if I<=N then scrivi 'Trovato nella posizione',I
else scrivi 'Non Trovato'.
```

Questo metodo è sicuramente efficace e funzionale, ma molto dispendioso, in quanto vengono eseguiti 2 confronti per ognuno degli N/2 cicli che si eseguono in media (N cicli in caso di insuccesso). Si può dimezzare il numero di confronti con un piccolo accorgimento:

```
program RICERCA-CON-SENTINELLA;
N=...
var A : array[1..N+1]of integer;
leggi K;
A[N+1]=K;"introduci K in fondo come sentinella"
I=1;
while A[I]<>K do I=I+1;
if I<=N then scrivi 'trovato nella posizione',I
else scrivi 'Non trovato'
```

L'accorgimento consiste nel fatto di evitare il controllo di fine vettore introducendo K dopo l'ultimo elemento. Si tratta, tuttavia, ancora di un metodo sequenziale che può andar bene per tabelle di piccole dimensioni. Per esempio nessuno di noi ricerca una parola su un dizionario con migliaia di voci. In questi casi, come nella ricerca binaria che vedremo tra breve, si sfrutta l'ipotesi che le voci siano ordinate. Il metodo riferito ad una ricerca su un dizionario, si formulerebbe come segue:

"Aprire a metà il dizionario: guardare se, per combinazione, la parola si trova sulla pagina aperta : se si trova lì, la ricerca si conclude con successo. Altrimenti, escludere la metà del vocabolario su cui non può certamente trovarsi ed applicare il procedimento di ricerca sulla metà buona fino a quando il numero delle pagine si riduce ad uno : o è lì oppure significa che non c'era". Il metodo può essere formalizzato con il seguente algoritmo:

```
program RICERCA-BINARIA-O-DICOTOMICA
N=...
var A : array[1..N]of integer;
leggi K;
INF=1;
SUP=N;
while INF<=SUP "finchè c'è spazio per ricercare"
do begin "K può trovarsi tra A[INF] e A[SUP]"
CENTRO=(INF+SUP)/2;"divisione intera"
if A[CENTRO]=K then exit; "trovato!!!"
if A[CENTRO]>K then INF=CENTRO+1
if A[CENTRO]<K then SUP=CENTRO-1
end;
if INF<=SUP then scrivi 'Trovato nella posizione',CENTRO
else scrivi 'Non trovato'.
```

La ricerca binaria rivela la sua sostanziale superiorità, rispetto a quella sequenziale, all'aumentare della lunghezza N del vettore. Per dare un'idea più precisa potremmo dire che se  $N=2^k$ , nel caso peggiore (ricerca con insuccesso) la ricerca sequenziale richiede  $2^k$  confronti, mentre quella binaria richiede  $k+1$  iterazioni ( $2(k+1)$  confronti). Si tenga presente che esistono dei metodi di ricerca ancor più veloci (per esempio tecnica di ricerca hash).

### ORDINAMENTO

Anche in questo caso, come nella ricerca, esistono numerosi algoritmi di ordinamento con prestazioni molto diverse fra loro. Ora ne esamineremo giusto un paio fra quelli a basse prestazioni per la loro semplicità. Quando si chiede ad uno studente di scrivere un algoritmo di ordinamento, nella maggior parte dei casi ci viene proposto il seguente:

```
program ORDINAMENTO-SEMPLICE;
N=...
var A : array[1..N]of integer;
for I=1 to N-1 do
for J=I+1 to N do
if A[J]<A[I] then scambia A[I] e A[J]
```

All'i-esimo ciclo del for più esterno si porta all'i-esimo posto l'elemento minimo fra quelli di indice compreso fra i e N, effettuando lo scambio ogni volta che se ne trova uno minore. Tuttavia non si tratta del miglior algoritmo. Ad esempio, se il vettore è già ordinato esegue sempre lo stesso numero di confronti, pari a  $N(N+1)/2$ . Un certo miglioramento può essere ottenuto esaminando una generalizzazione del metodo di ordinamento di tre numeri presentato in precedenti lezioni. L'idea di base è quella di spingere successivamente in fondo i numeri più grandi attraverso degli scambi fra coppie di elementi adiacenti. Si dice anche che gli elementi maggiori "salgono come bolle di sapone", da cui la denominazione di BUBBLE SORT per questo metodo di ordinamento. Durante il primo ciclo di scambi ci si assicura di portare in fondo il numero più grande. Al ciclo successivo si può evitare di spingersi fino al penultimo posto limitandosi ad arrivare all'ultimo elemento scambiato al ciclo precedente: alla sua destra il vettore è necessariamente già ordinato (il discorso vale anche per tutti i cicli successivi). Nell'algoritmo che segue si comprenderà il ruolo della variabile-indice ULTIMO-SCAMBIATO che viene azzerata all'inizio di ogni ciclo per potersi accorgere che non è stato fatto alcuno scambio, cioè per stabilire che il vettore è già ordinato.

```
program BUBBLE-SORT;"versione migliorata"
N=...
var A : array[1..N]of integer;
SUP=N;
while SUP>0 "finchè il vettore non è ordinato..."
do begin
ULTIMO-SCAMBIATO=0;
for I=1 to SUP-1 do
if A[I]>A[I+1] then begin
A[I]=A[I+1];
A[I+1]=A[I];
ULTIMO-SCAMBIATO=1
end;
SUP=ULTIMO-SCAMBIATO;
"il vettore è ordinato da ULTIMO-SCAMBIATO + 1 a N"
end
```

Si noti che l'algoritmo sfrutta il fatto che alcune sottosequenze sono già ordinate. Infatti, nel caso in cui il vettore sia ordinato vengono eseguiti solo N-1 confronti (si esce dopo il primo ciclo). Al contrario, il caso peggiore si ha quando il vettore è ordinato in senso inverso. In media è leggermente migliore del precedente. Tuttavia, per fare un sostanziale salto di qualità occorre servirsi di metodi più complessi noti come QUICKSORT, MERGESORT, e altri. Per dare un'idea, questi algoritmi consentono di ordinare vettori lunghi  $N=2^k$  con un numero di confronti pari a  $k^2/2$  contro i circa  $2^k/2$  dei metodi presentati.

Dopo aver concluso lo scorso appuntamento parlando dei tipi semplici, questa volta esamineremo il tipo enumerativo, i tipi predefiniti, integer, real, boolean e char.

## TIPO ENUMERATIVO

Il primo passo verso la definizione di un tipo di dato è stabilire quali sono i valori che lo costituiscono. Supponiamo di avere la necessità di un tipo di dato che mantenga l'informazione sui giorni della settimana. Quello che serve innanzitutto è enumerare i possibili valori dei giorni. Il programmatore ha la possibilità in Pascal di definire un TIPO ENUMERATIVO che esplicita direttamente i valori del tipo. Nel nostro caso possiamo definire ad esempio:

```
TYPE giorno = (lun, mar, mer, gio, ven, sab, dom);
```

Un tipo enumerativo è definito elencando gli identificatori con cui sono denotati i valori del tipo. I nomi introdotti dall'utente sono in pratica le costanti letterali che caratterizzano il tipo. Evidentemente una variabile dichiarata di quel tipo può assumere uno di quei valori e nessun altro valore. Alcuni altri oggetti appropriatamente descritti da tipi enumerativi sono:

```
TYPE colore = (bianco, rosso, blu, nero, viola);  
frutta = (limone, mela, pesca, ananas);  
VAR poligono : (triangolo, quadrato, pentagono);  
moneta : (dollaro, marco, yen, lira);
```

L'ordine con cui compaiono nella enumerazione gli identificatori definisce l'ORDINAMENTO dei valori denotati da quei nomi. In conseguenza delle convenzioni di ordinamento, nel caso precedente valgono le seguenti relazioni:

```
mar < gio  
dom > mer
```

In virtù dell'ordinamento stabilito, come vedremo successivamente, gli oggetti di questo tipo potranno essere confrontati mediante operazioni relazionali. Inoltre funzioni standard che ritornano il predecessore, il successore e la posizione di un valore sono anche fornite. La presenza dell'ordinamento dei valori potrebbe apparire una sovrapposizione inutile rispetto alle esigenze del programmatore. In realtà ciò può tornare utile in molte situazioni, ad esempio quando una iterazione ordinata sugli elementi dell'insieme è richiesta. In ogni caso, se usato o meno, l'ordinamento non comporta alcuna codifica aggiuntiva. Si noti infine che la costante enumerativa denota uno e un solo valore. Non è possibile quindi usare lo stesso identificatore in due definizioni di tipi enumerativi. La seguente definizione è illegale se aggiunta alle precedenti:

```
TYPE semaforo = (giallo, rosso, verde);
```

Un certo insieme di "funzioni" è disponibile sui tipi enumerativi; il fatto che essi siano ordinati permette infatti di avere una funzione PRED, che individua il predecessore di un valore nella sequenza dei simboli, e una funzione SUCC, che ne individua il successore; nella definizione del tipo giorno:

```
pred(mer) = mar
```

```
mentre
```

```
succ(mer) = gio
```

Inoltre la posizione del simbolo nella definizione è ottenibile con la funzione ORD; considerando numerato con zero il primo posto, si ha:

```
ord(lun) = 0  
ord(gio) = 3
```

Di fatto la funzione ORD fornisce in genere la codifica interna del simbolo che i compilatori Pascal fanno corrispondere alla posizione nella definizione.

## TIPI SUBRANGE

Supponiamo ora di aver bisogno di un tipo di dato che enumeri solamente i giorni lavorativi della settimana e quelli festivi. Questo può essere ottenuto definendo un sottoinsieme dei valori contigui del tipo giorno precedentemente definito. Un TIPO SUBRANGE è definito indicando i limiti inferiore e superiore dei valori del sottoinsieme del tipo enumerativo genitore. Nella nostra esemplificazione potremmo introdurre le definizioni:

```
TYPE feriale = lun...ven;  
festivo = sab...dom;
```

Il tipo "feriale" contiene nell'ordine i valori "lun", "mar", "mer", "gio" e "ven", mentre il tipo "festivo" solamente i valori "sab" e "dom". La relazione d'ordine introdotta nella definizione del tipo genitore continua a valere ancora per i due nuovi tipi. Di conseguenza la definizione di un tipo subrange è legale solo se:

```
limite inferiore <= limite superiore
```

E' scorretta quindi la definizione seguente:

```
TYPE crazyweek = dom..mar;
```

## TIPI PREDEFINITI

In entrambi i casi precedenti l'utente ha la possibilità di definire tipi di dati propri, ovvero orientati al particolare problema. Essi sono infatti anche chiamati TIPI DEFINITI dall'utente. Tuttavia il programmatore ha la possibilità di usare direttamente tipi di dati la cui definizione è implicitamente data dal linguaggio. Essi definiscono semplici e intuitive astrazioni sulla macchina reale che rendono invisibile al programmatore la rappresentazione fisica di tali dati. Il Pascal supporta quattro TIPI PREDEFINITI denotati da altrettanti identificatori standard.

## TIPO INTEGER

Il tipo predefinito INTEGER rappresenta l'insieme dei numeri interi rappresentabili nella parola di memoria della macchina. In conseguenza di tale restrizione possiamo meglio definire tale tipo standard come un subrange di un ipotetico tipo intero universale, cioè:

```
TYPE integer = -maxint..maxint;
```

dove maxint è un altro identificatore denotante il più grande numero intero rappresentabile sulla macchina. Tutti i tipi di dati che hanno una natura numerabile possono essere rappresentati in Pascal mediante il tipo INTEGER. Oltre alla natura enumerativa, questo tipo di dato è caratterizzato anche da altre operazioni specifiche del tipo. Come vedremo successivamente, le familiari operazioni aritmetiche possono essere applicate, caratterizzandolo

# IL PASCAL (Parte IV<sup>a</sup>)

rispetto ad un generico tipo enumerativo. Essendo predefinito anche le costanti letterali che denotano i suoi valori sono unità lessicali specificate direttamente dal linguaggio. Alcuni esempi di applicazioni di questo tipo di dato sono:

```
CONST dozzina = 12;  
TYPE lunghezza = integer;  
VAR perimetro : integer;  
contatore : integer;
```

Supponiamo ora di avere la necessità di un tipo di dato che numeri solamente le ore della giornata. In tal caso sappiamo a priori dell'esistenza di un limite inferiore e superiore ai valori permessi. Questo può essere riportato in Pascal introducendo un tipo subrange del tipo intero standard come di seguito specificato:

```
TYPE ora = 0..24;
```

Oltre a una maggiore leggibilità e aderenza al problema, il linguaggio inoltre utilizza tale informazione sul range dei valori per effettuare controlli automatici sia durante la traduzione, sia durante l'esecuzione, sul corretto uso di oggetti di tale tipo. Questo consente una più veloce messa a punto del programma e una maggiore affidabilità dello stesso.

## TIPO REAL

Supponiamo di voler misurare con massima precisione una grandezza fisica, ad esempio la pressione o la temperatura. In questo caso i valori discreti e numerabili del tipo integer introdurrebbero eccessiva approssimazione nel ricordare la misura. I numeri reali sono la soluzione a tali casi. L'insieme dei numeri reali è rappresentato in Pascal dal tipo standard REAL. In questo caso la limitazione fisica della parola di memoria della macchina introduce non solo dei limiti sul range dei valori permessi, ma anche sulla precisione degli stessi. Questo ha l'effetto di rendere in pratica "discreta" la natura "continua" dei numeri reali. Per questo motivo due distinti valori di tipo real, in senso matematico, possono essere rappresentati dallo stesso valore di tipo real sulla macchina. Alcuni esempi di definizione con il tipo real sono:

```
CONST g = 9.8;  
TYPE pressione = real;  
VAR ascissa, ordinata: real;
```

Analogamente al tipo integer, anche il tipo real è caratterizzato dalle note operazioni aritmetiche. Si noti infine che non essendo di utilità pratica il riconoscimento del tipo real come enumerativo (seppure possibile per le ragioni anzidette), il linguaggio non consente di esprimere subrange di questo tipo.

## TIPO BOOLEAN

Tutta la logica opera su un tipo di dato costituito da due soli e alternativi valori. A questo scopo il Pascal introduce il tipo predefinito BOOLEAN, con la seguente definizione:

```
TYPE boolean = (false, true);
```

I due valori di verità FALSE e TRUE sono nomi standard introdotti dal linguaggio. Evidentemente questa definizione introduce una convenzione di ordinamento sui due va-

lori, cioè:

```
FALSE < TRUE
```

Questo tipo è ovviamente caratterizzato da particolari operazioni logiche derivate dall'algebra booleana. Alcuni esempi di uso del tipo booleano sono:

```
CONST vero = true;  
TYPE bit = boolean;  
VAR flag : boolean;  
errore : boolean;
```

## TIPO CHAR

Normalmente la comunicazione e l'iterazione tra la macchina e il mondo esterno avviene non attraverso la codifica binaria (che è la rappresentazione interna della macchina e la modalità con cui avviene la maggior parte dell'elaborazione), ma con la forma letterale tipicamente usata dall'uomo.

Il tipo predefinito CHAR in Pascal è l'insieme finito e ordinato dei caratteri utilizzabili sulla macchina. Tutti i caratteri grafici sono presenti in questo insieme.

Come per i tipi integer e real, i valori sono insiti nel linguaggio stesso e non possono essere ridefiniti dall'utente. Le costanti letterali denotanti i valori di tale tipo sono rappresentate dalle stringhe di caratteri di lunghezza uguale a 1. Solo i caratteri grafici hanno corrispondenti valori letterali specificati dal linguaggio. Tutti gli altri caratteri riconosciuti dalla macchina (come quelli di controllo di formato) non sono denotabili con costanti letterali.

Alcune esemplificazioni sono date di seguito e utilizzano la carta sintattiche delle stringhe di caratteri:

```
CONST blank = ' ';  
kappa = 'k';  
quote = '"';  
TYPE lettera = char;  
VAR simbolo : char;
```

Sia i caratteri utilizzabili sia il loro ordinamento dipendono dalla specifica macchina.

Normalmente su tutte le macchine è vero che

```
'a' < 'b' < 'c' ..... < 'z'  
'0' < '1' < '2' ..... < '9'
```

In tal caso la natura enumerativa del tipo ci permette di scrivere i seguenti tipi subrange:

```
TYPE maiuscole = 'A'..'Z';  
VAR cifra : '0'..'9';
```

Non è ovviamente consentito dal linguaggio, per i motivi precedentemente detti, definire una nuova enumerazione costituita di caratteri, come la seguente:

```
TYPE vocali = "'a', 'e', 'i', 'o', 'u';
```

Ricordando quanto detto sulla funzione ORD, la sua applicazione a un carattere ci fornisce la codifica interna dello stesso.

(continua)

# DENTRO L'MSX (Parte V<sup>a</sup>)

Dopo aver concluso lo scorso numero con un programma per la definizione dei caratteri, questa volta continueremo a parlare della manipolazione della memoria video, iniziando a parlare degli sprite.

## GLI SPRITE

Le istruzioni e le variabili fondamentali del BASIC per la manipolazione degli sprite sono:

```
SPRITE$(Numero-modello)=XYZ$
PUT SPRITE n,(X,Y),Colore,Numero-modello
PUT SPRITE n,STEP(x,y),Colore,Numero-modello
SCREEN Modo,Dimensione-sprite
SPRITE ON/OFF/STOP
ON SPRITE GOSUB
```

La dimensione degli sprite può essere sia di 8\*8 pixel sia di 16\*16. Inoltre entrambe le suddette dimensioni possono essere raddoppiate. L'istruzione SCREEN determina la dimensione che verrà usata:

```
0=8*8          2=16*16
1=8*8 raddoppiato 3=16*16 raddoppiato
```

quindi, ad esempio, SCREEN 2,2 provoca l'attivazione della modalità ad alta risoluzione grafica (HRG) con gli sprite di dimensione 16\*16 pixel. Uno sprite di 8\*8 pixels è definito, come un carattere, per mezzo di otto byte. Uno di dimensioni 16\*16 viene definito usando quattro blocchi di otto byte:

```
1 17          25 9
| |          | |
8 24          32 16
```

Questi ultimi sono definiti tramite l'istruzione SPRITE\$. Ciascuna definizione di modello di sprite viene introdotta in forma di stringa di 8 o di 32 caratteri, dove ogni numero corrispondente al carattere rappresenta un byte del modello. Per esempio, per definire uno sprite 8\*8 che rappresenti il segno '-':

```
A$=CHR$(0):B$=CHR$(126)
SPRITE$(0)=A$+A$+A$+B$+B$+A$+A$+A$
```

Infatti la configurazione dei bit che rappresentano 126 in numerazione binaria è la seguente:

```
128 64 32 16 8 4 2 1
1 1 1 1 1 1
```

In alternativa all'uso dell'istruzione SPRITE\$ si potrebbe usare la VPOKE per scrivere i primi otto byte della tabella dei modelli per gli sprite:

```
FOR X=14336 TO 14343:VPOKE X,0:NEXT: VPOKE 14339,126
: VPOKE 14340,126
```

Se sono attualmente in uso gli sprite 8\*8 si possono definire fino a 256 modelli di sprite, mentre con quelli a 16\*16 pixel ne sono consentiti  $256/4=64$ . La possibilità di definire sprite di dimensioni così piccole facilita l'uso di un "insieme di caratteri" basato sugli sprite. E' infatti possibile ottenere testo di dimensioni doppie e a più colori usando il suddetto insieme di caratteri unitamente all'opzione di raddoppio delle dimensioni. Sebbene esista il limite di quattro lettere per ogni linea dello schermo, questo approccio evita i problemi associati con la giustificazione a sinistra dell'insieme di caratteri residente nella RAM di sistema. Uno sprite può essere visualizzato sullo schermo e fatto muovere usando una delle due forme dell'istruzione PUT SPRITE. Lo schermo è indirizzabile secondo coordinate che valgono 0,0 nell'angolo superiore sinistro, 0,190 sull'angolo inferiore sinistro e 255,0 nell'angolo superiore destro. Una qualsiasi coppia di coordinate si riferisce al pixel in alto a sinistra dello sprite. Per poter far apparire lentamente lo sprite stesso entrambe le coordinate assumono come estremo inferiore -32. In questo modo è possibile posizionare sprite "fuori" dallo schermo. In ogni caso valori delle coordinate fuori dello schermo, ma comprese tra -32768 e 32767 non provocano una situazione di errore.

Infatti lo sprite verrà posizionato, in tal caso, secondo le coordinate che si ottengono dai valori dati, dividendoli per 256 e considerando il resto della divisione intera. L'unica eccezione si ha quando la coordinata Y assume il valore 208. Infatti quando il VDP si trova in questa situazione provvede a togliere dallo schermo lo sprite in questione e tutti quelli di priorità inferiore. Si tratta di un metodo efficace per far lampeggiare gli sprite. Il formato standard delle istruzioni PUT SPRITE è il seguente:

```
PUT SPRITE piano,(X,Y),colore,numero del modello
```

Il piano 0 è il piano a più alta priorità: ne consegue che ogni parte non trasparente di uno sprite che si trovi nel piano 0 si sovrapporrà a qualsiasi cosa vi sia sullo schermo nella medesima posizione. Il piano a priorità più bassa è il 32. Per ogni piano è consentito un solo sprite. Si ha inoltre a disposizione la stessa gamma di colori che si usa nell'istruzione COLOR. Si noti che ogni sprite può essere di un solo colore. Se non lo si specifica, viene usato il colore corrente di primo piano. Se non vengono specificate le coordinate lo sprite viene posizionato alle coordinate 209,0. Un'eccezione a questo comportamento si ha nel caso in cui uno sprite sia stato posizionato in precedenza sul piano in questione: in questo caso si usano le precedenti coordinate. Ad esempio: PUT SPRITE 0,(40,40),10,0:PUT SPRITE 0,,10,2 provoca la sostituzione sul piano 0, in posizione 40,40 dello sprite modello 0 con quello modello 2. L'istruzione PUT SPRITE può assumere anche il seguente formato:

```
PUT SPRITE Piano,STEP(X,Y),colore,numero del modello
```

che consente di variare le coordinate dello sprite in relazione a quelle correnti. Gli sprite hanno priorità maggiore rispetto al piano di visualizzazione dei caratteri: ogni parte non trasparente di uno sprite si sovrapporrà quindi alla schermata sottostante. Così come la limitazione di un solo colore per ciascuno sprite, anche questa priorità è intrinseca nel VDP. L'istruzione PUT SPRITE aggiorna un'entrata di quattro byte nella tabella degli attributi degli sprite, nella VRAM. In tutte le modalità che consentono l'uso degli sprite, nella VRAM. In tutte le modalità che consentono l'uso degli sprite, la suddetta tabella inizia all'indirizzo 6912 ed è lunga  $32*4=128$  byte. I quattro byte di ciascuna entrata sono così allocati:

```
Byte 1 Coordinata Y 3 Numero del modello
      2 Coordinata X 4 Colore/"Clock early bit"
```

In BASIC, la coordinata Y deve essere compresa tra -32 e 209. Però, poiché un byte senza segno può assumere valori tra 0 e 255, nella tabella viene usata la notazione in complemento a 2 per le coordinate negative (troverete delle spiegazioni sulla notazione in complemento a due in una prossima puntata che riguarderà il linguaggio Assembly dello Z80). Diverso è il caso della coordinata X che viene spostata di 32 pixel a sinistra nel caso sia stato impostato a 1 il "clock early bit". Per esempio, per posizionare uno sprite blu sul piano 0 con numero di modello 1 alle coordinate 100,100, le istruzioni che seguono costituiscono una alternativa all'uso di PUT SPRITE:

```
VPOKE 6912,100:VPOKE 6913,100
VPOKE 6914,1:VPOKE 6915,4
```

Per spostare a sinistra lo sprite di 32 pixel si potrebbe usare sia:

```
VPOKE 6913,68
sia: VPOKE 6915,132.
```

Con la seguente si possono togliere tutti gli sprite dallo schermo:

```
VPOKE 6912,208
```

Questo è tutto per questa volta. Concluderemo il discorso sugli sprite nel prossimo numero, dove cominceremo a parlare di grafica ad alta risoluzione.

(continua)

I calcoli matematici sono semplificati dalle numerose funzioni disponibili nel linguaggio BASIC. In particolare, sono comprese nel linguaggio le principali funzioni trigonometriche.

## MATEMATICA AL COMPUTER

In campo matematico, il computer "gioca in casa": non dimentichiamo che le prime rudimentali macchine da calcolo nascevano proprio dall'esigenza di sveltire i lunghi e noiosi procedimenti di calcolo manuale, nei quali era facile commettere errori. Per questo, naturalmente, oggi è sufficiente una qualsiasi calcolatrice. Il computer, tuttavia, ha il vantaggio di poter ripetere automaticamente le operazioni richieste, una volta programmato. Ad esempio, un programma per la soluzione delle equazioni di secondo grado necessita di essere scritto una sola volta. Terminato questo lavoro, possiamo risolvere tutte le equazioni che vogliamo, senza più fare un solo calcolo a mano. Consideriamo la classica equazione:

$$ax^2 + bx + c = 0$$

A costo di diventare noiosi, lo ripetiamo ancora: un computer non può risolvere niente da solo, occorre che il programmatore conosca la soluzione del problema. Dunque, il computer non può aiutarci a trovare la procedura risolutiva (algoritmo); al contrario, siamo noi che dobbiamo studiarla ed insegnarla alla macchina. Nel nostro caso, la procedura sembra facile, basta applicare la nota formula per ottenere le soluzioni. Apparentemente, il solo problema per la traduzione in BASIC è la radice quadrata. Per fortuna, il BASIC ci offre una funzione apposita: la SQR. Ad esempio:

con PRINT SQR(4) si ottiene 2  
con PRINT SQR(2) si ottiene 1.41421356

Perciò, si potrebbe pensare di scrivere il programma così:

```
10 REM SECONDO GRADO
20 INPUT "COEFFICIENTI A, B, C? ";A,B,C
30 R1 = (-B + SQR(B^2 - 4*A*C))/(2*A)
40 R2 = (-B - SQR(B^2 - 4*A*C))/(2*A)
50 PRINT " LE SOLUZIONI SONO ";R1," E ";R2
```

Semplice, vero? Proviamo a risolvere l'equazione:

```
x^2 + 2x - 15 = 0
RUN
COEFFICIENTI A, B, C? 1, 2, -15
LE SOLUZIONI SONO 3 E -5
```

(In risposta alla INPUT abbiamo fornito i tre valori, separati da virgole). Fin qui, tutto bene. Proviamo adesso con l'equazione:

```
x^2 + 2x + 4 = 0
RUN
COEFFICIENTI A, B, C? 1, 2, 4
?ILLEGAL QUANTITY ERROR IN 30
```

Ahi, questo non era previsto! Il discriminante ( $B^2 - 4AC$ ) è negativo ed il computer non può estrarre la radice quadrata di un numero negativo. Proviamo un'ultima equazione:

```
2x + 3 = 0 o, se preferite, 0x^2 + 2x + 3 = 0
RUN
```

## COEFFICIENTI A, B, C? 0, 2, 3 ?DIVISION BY ZERO ERROR IN 30

Un altro problema! Se l'equazione si riduce al primo grado ( $A=0$ ) il denominatore ( $2A$ ) diventa zero e la formula non è più usabile. Conclusione? Abbiamo sbagliato tutto! NON BISOGNA METTERSI SUBITO ALLA TASTIERA E SCRIVERE IL PROGRAMMA. Anche il problema più semplice può essere pieno di trabocchetti. Soltanto dopo aver esaminato accuratamente il problema e scritto un algoritmo che tenga conto di tutte le condizioni, si può introdurre il programma nel calcolatore. Il nostro programma, oltretutto, esegue due volte gli stessi calcoli (linee 30 e 40), altro segno dell'assenza di uno studio razionale. Ricominciamo daccapo.

## EQUAZIONI BEN FATTE

Di seguito troverete il listato di un programma in cui abbiamo seguito delle condizioni necessarie:

abbiamo distinto subito il caso in cui  $A=0$ , in quanto l'equazione diventa di primo grado e la soluzione va calcolata usando una formula differente.

se l'equazione è di primo grado, non può essere zero anche B coefficiente di X, altrimenti tutto si riduce a  $C=0$ , che non è un'equazione valida.

se l'equazione è di secondo grado, osserviamo il discriminante.

Se è negativo, non ci sono soluzioni reali (per semplicità, non abbiamo calcolato le soluzioni immaginarie).

al termine del lavoro, diamo la possibilità di continuare a risolvere altre equazioni, invece di terminare comunque il programma.

```
100 REM EQUAZIONI DI SECONDO GRADO
140 PRINT:PRINT "RISOLUZIONE DI EQUAZIONI"
150 PRINT "DI SECONDO GRADO DEL TIPO"
160 PRINT:PRINT "A * X^2 + B * X + C = 0"
180 REM INPUT COEFFICIENTI
200 PRINT:PRINT 210 INPUT "COEFFICIENTI A,B,C?";
A,B,C
230 IF A=0 GOTO 430
250 REM E' DI SECONDO GRADO
270 D=B*B-4*A*C:REM DISCRIMINANTE
280 IF D<0 THEN PRINT "L'EQUAZIONE NON HA SOLUZIONI REALI.":GOTO 530
290 D=SQR(D):REM NON E' NEGATIVO
300 R1=(-B+D)/(2*A)
310 R2=(-B-D)/(2*A)
330 REM STAMPA SOLUZIONI (2)
350 PRINT "L'EQUAZIONE HA DUE":PRINT "SOLUZIONI ";
360 A$="DISTINTE. ":IF R1=R2 THEN A$="COINCIDENTI. "
370 PRINT A$
380 PRINT:PRINT R1,R2
390 GOTO 530
400 REM E' DI PRIMO GRADO
430 IF B=0 THEN PRINT "NON E' UN'EQUAZIONE!":GOTO 530
440 R1=-C/B
460 REM STAMPA SOLUZIONE (1)
480 PRINT "L'EQUAZIONE E' DI PRIMO GRADO."
```

```
490 PRINT "LA SOLUZIONE E' ";R1
510 REM FINE OPERAZIONI
530 PRINT
540 INPUT "UN'ALTRA EQUAZIONE ? ";A$
550 A$=LEFT$(A$+" ",1):REM PRIMO CARATTERE
560 IF A$="S" GOTO 200:REM ALTRA EQUAZIONE
570 IF A$<>"N" GOTO 540:REM NON CAPISCO
580 PRINT:PRINT "CIAO!":END
```

Come si può notare, abbiamo dato ampio spazio ai commenti. Anche se questi possono sembrare inutili quando si scrive il programma, si rivelano invece preziosi quando capita di rileggerlo a distanza di tempo o, a maggior ragione, quando il programma deve essere letto e compreso (e magari modificato) da qualcun'altro.

Dato che lo spazio in memoria non è ormai più un problema, nemmeno nei calcolatori economici, non è il caso di risparmiare commenti: meglio troppi che troppo pochi e, comunque, non sono mai troppi.

Le linee dalla 140 alla 160 stampano un'intestazione che spiega brevemente a cosa serve il programma ed illustra il tipo di equazione che è in grado di risolvere.

E' molto utile mettere qualche istruzione del genere all'inizio dei propri programmi: l'uso ne risulta semplificato e non occorre andare a cercare fogli di spiegazioni, o esaminare il programma per cercare di capire come funziona. Un programma dev'essere il più possibile AUTO-ESPLICATIVO, cioè l'operatore deve capire come comportarsi anche se lo usa per la prima volta.

Le linee dalla 200 alla 220 effettuano l'input dei valori dei tre coefficienti A, B e C. L'input potrebbe essere fatto in modo più raffinato, comunque funziona, è comprensibile e questa è la cosa importante.

La linea 230 controlla il valore di A. Se è zero, l'equazione è di primo grado e va trattata diversamente; salta quindi alla linea 430. Altrimenti prosegue normalmente. Poiché lo zero corrisponde al valore logico falso, si poteva anche scrivere `IF NOT A GOTO 430` (se A non è vero, dunque se è falso, cioè se è zero, allora salta), ma sarebbe stato meno chiaro. Si tratta, comunque, di una costruzione molto usata che è bene conoscere.

La linea 270 calcola il valore del DISCRIMINANTE (la parte sotto radice nella formula risolutiva) e la 280 controlla il segno del risultato ottenuto.

Se è negativo, viene stampato che l'equazione non ha soluzioni reali ed il programma prosegue alla linea 530, punto di riunione finale.

E' sempre buona prativa avere un unico punto finale, piuttosto che vari END in mezzo al programma. Se occorre poi aggiungere qualcosa in fondo (in questo caso, la scelta se smettere o continuare), la cosa risulta molto più facile.

Poiché il discriminante non è negativo, l'estrazione della radice quadrata (linea 290) non presenta problemi, come pure il calcolo delle due soluzioni reali dell'equazione (linee 300-310). Le linee 350-390 stampano i valori delle soluzioni con una raffinatezza: se le due soluzioni sono diverse viene stampato "L'EQUAZIONE HA DUE SOLUZIONI DISTINTE", altrimenti "L'EQUAZIONE HA DUE SOLUZIONI COINCIDENTI". Per ottenere questo, si stampa prima la parte comune "L'EQUAZIONE HA DUE SOLUZIONI " (notate lo spazio finale), che nel programma è scomposta in due righe per non "debordare" su uno schermo a 40 colonne. Poi bisognerebbe scrivere:

```
IF R1=R2 THEN PRINT "COINCIDENTI:" ELSE PRINT
"DISTINTE:"
```

Poiché molti BASIC non dispongono dell'ELSE, abbiamo usato un trucco.

La variabile A\$ è posta comunque uguale a "DISTINTE". Se R1 ed R2 sono uguali, il contenuto viene sostituito con "COINCIDENTI:"; A\$ viene poi stampata. E' un sistema utile per risparmiare un GOTO (o un altro IF).

Stampate le due soluzioni reali, la linea 390 rimanda al punto finale comune.

Se l'equazione è di primo grado, la linea 430 (chiamata dalla 230) controlla se anche il coefficiente di x (cioè B) è zero.

In questo caso non si tratta nemmeno di un'equazione di primo grado. Messaggio e salto al punto finale.

Se invece B non è zero, la linea 440 calcola la soluzione dell'equazione di primo grado e le linee 480-490 la stampano.

Siamo arrivati al punto comune, dove si riuniscono tutte le varie "strade" percorse dal programma (come mostra chiaramente anche il diagramma di flusso). La linea 540 chiede se si vuole risolvere un'altra equazione.

In caso affermativo, rimanda all'input (linea 200). E' interessante la tecnica usata per esaminare la risposta dell'operatore: poiché viene esaminato soltanto il primo carattere della risposta (per effetto della LEFT\$ alla linea 550), è possibile scrivere "S" oppure "SI" indifferentemente.

Prima di effettuare il LEFT\$, viene aggiunto uno spazio, per il caso in cui la risposta consista in un semplice CR (stringa nulla) e la LEFT\$ non accetti una stringa nulla come argomento.

Inoltre, se la risposta non è "SI" (o "S"), si verifica che sia "NO" (o "N"). In caso contrario, viene ripetuta la domanda alla linea 540.

Infine, se la risposta è "NO" (o "N"), il programma termina (linea 570). L'END è solo per chiarezza.

Questo era un esempio di un programma abbastanza ben scritto, progettato per tenere conto di tutti i casi possibili e mediamente commentato.

Si potrebbe fare di meglio, per esempio trattare anche il caso di soluzioni immaginarie, o commentare più a fondo, descivendo l'uso di ciascuna variabile.

Già così, comunque, il programma è notevolmente più lungo di quello (sbagliato) del primo esempio. Scrivere un programma e; semplice; scriverlo bene, lo è un po' meno (ma ne vale sempre la pena!).

(continua)



GRUPPO EDITORIALE  
INTERNATIONAL EDUCATION

Caro amico,

conosciamo il tuo interesse a quanto, di seguito, siamo in grado di offrirti in forma esclusiva ed inedita.

Abbiamo selezionato la più completa ed interessante raccolta di programmi gioco e utility per l'utilizzo del tuo Msx che possono soddisfare ogni particolare esigenza.

Questi ti verranno pubblicati con sequenza mensile, in elenchi da 100 pezzi, nei quali potrai scegliere e richiedere quelli che ti interessano particolarmente: otterrai così uno o più dischi che conterranno un minimo di 5 programmi ai seguenti prezzi: L. 7.000 per ogni gioco Msx 1, L. 9.000 per ogni gioco Msx 2, L. 10.000 per ogni utility, L. 2.500 per ogni dischetto usato per contenere i 5 programmi da te scelti. La spedizione sarà in contrassegno, cioè pagherai al postino quando riceverai i dischi. Il costo della spedizione sarà aggiunto a quello totale dei programmi da te scelti. Ti saluto cordialmente e attendo tue notizie.

GRUPPO EDITORIALE INTERNATIONAL EDUCATION

Viale Famagosta, 75 - 20142 Milano - Tel. 02 - 89502288 r.a. - Telefax: 02 - 8466834 - Telex: 352191 GRELIN I

---



CODICE	NOME PROGRAMMA	AUTORE	TIFO	Kb	MSX	CODICE	NOME PROGRAMMA	AUTORE	TIFO	Kb	MSX
2010	COPIA 5		COPY CASS.	32	1	2020	COPIATORE CASSETTE		COPIATORE	32	1
203188	COPY 1		DISKCOPY	32	1	2040	COPY DISK	PICAR	COPIATORE	32	1
2050	COPY V19		COPY CASS.	32	1	20635	CORAGEOUS PERSEUS	COSMOS	ARCADE/AVV	64	1
20730	COSA NOSTRA	OFERA	ARCADE/AVV	64	1	20827	COSMIC ABSORBER	MARTECH	ARCADE/AVV	64	1
20913	COSMO EXPLORER	ZAF	ARCADE	64	1	210142	CRAFTON & XUNK	ERE	ARCADE/AVV	64	2
2112	CRAZY BULLET	ASCII	ARCADE	64	1	21247	CRAZY TRAIN	SONY	ARCADE	64	1
2130	CREATIVE GRAPH EDDY2	HAL & SONY	GRAFICA	64	1	21450	CRUSADER	PONYCA	ARCADE/AVV	64	1
215188	CSBK 5		COPY CASS.	32	1	216103	CSO!	PONYCA	ARCADE	32	1
21723	CYBERUN	ACG	ARCADE/AVV	64	1	21847	CYRUS II	I.C.S	SCACCHI	32	1
21923	DAMBUSTER	OCEAN	SIM/GUERRA	64	1	22023	DANGER X 4	ASCII	ARCADE	64	1
221104	DANGERHOUSE	THORN EMI	AVV/GRAFIC	64	1	22274	DARTS	ORPHEUS	FRECCETTE	32	1
22340	DAWN PATROL		ARCADE	64	1	224145	DBASE 2	ASHTONTATE	DATABASE	64	1
2252	DBASE II	ASHTON TAT	DATABASE	64	1	226117	DEATH		ARCADE	32	1
22745	DEATHWISH 3	GREMLIN	ARCADE/AVV	64	1	22830	DECATHLON	ACTIVISION	SPORT	32	1
22934	DEMAND	MIRACLE	ARCADE	32	1	23042	DEMONIA L' INTRUSION	INFOGRAMES	ARCADE	64	1
2316	DESOLATOR	DOOLEY	SPAZIALE	64	1	23297	DESPERADO		ARCADE	64	1
23342	DESPRINTER		ARCADE	64	1	234117	DESTROYER	MIND GAMES	SPAZIALE	32	1
23540	DEUS EX MACHINA		ARCADE	64	1	236116	DEVORADOR	FICARSOFT	PAC MAN	64	1
23761	DIAMOND LUIS 1	IKESOFT	BOULDERDAS	32	1	23857	DIAMOND MINE 2	ELU RIBBON	ARCADE	64	1
2396	DIG DUG	NAMCO	ARCADE	64	1	240118	DIGIVOX	HARTARD	DIGIT VOICE	64	1
241146	DINAMIC PUBLISHER	REDRACSOFT	GRAFICA	64	2	24298	DINO SOURCERS	JALECO	ARCADE	32	1
24323	DIP OIP	INDESOFT	ARCADE	32	1	244147	DISCAIF	TECNOSOFT	ARCADE/AVV	64	2
2450	DISK EDITOR		MONITOR	32	1	2460	DISK HEADER		LETT.HEAD.	32	1
2470	DISK UTILITIES		PROG. VARI	32	1	24850	DISK WARRIOR	ALLIGATA	ARCADE 3D	64	1
24954	DIZZY BALL	PONYCA	ARCADE	32	1	25034	DIZZY BALLOON	PONYCA	ARCADE	32	1
251103	DIZZY DICE	PLAYERS	ARCADE	64	1	2527	DOG FIGHT		ARCADE	32	1
253112	DON QUIJOTE 1	DINAMIC	ARCADE/AVV	64	1	254107	DON QUIJOTE 2	DINAMIC	ARCADE/AVV	64	1
25534	DONKEY KONG	OCEAN	ARCADE	64	1	256103	DONFAN	COLPAX	ARCADE	32	1
25735	DOOR DOOR	ENIX	ARCADE	32	1	25822	DORODON	UPL	ARCADE	32	1
2597	DOS CHESS	PHILIPS	SCACCHI	64	1	26061	DOTA	SYSTEMSOFT	ARCADE/AVV	64	1
26126	DR JACKLE & MR WIDE	CURTIS	ADV/ARCADE	64	1	26258	DRACULA	GENESIS	ARCADE/AVV	32	1
26313	DRAGON ATTACK	HAL	ARCADE	64	1	26419	DRAUGHTS	ORPHEUS	DAMA	32	1
26583	DREAM RUNAWAY		ARCADE	32	1	26621	DRILLER TANK	HUDSON	ARCADE	32	1
267111	DROIDS THE WHITE W.	LUCASFILM	SPAZIALE	32	1	26840	DROME	BYTEBUSTER	ARCADE	64	1
26962	DUNGEON MASTER	ASCII	SIMUL/ADV	64	1	27065	DUNGEON MISTERY 2	SCORPION	ARCADE/AVV	64	1
27115	DUNKSHOT	HAL	SPORT	64	1	27241	DUSTIN	DINAMIC	ARCADE/AVV	64	1
2736	DYNAMITE DAN	MIRRORSOFT	ARCADE/AVV	64	1	27424	D-DAY	JALECO	ARCADE	32	1
27542	EAGLE	JULIET	ARCADE	64	1	2760	EASE	PHILIPS	INTEGRATO	64	2
27717	EFTAL	JAPAN	ARCADE	32	1	278163	EGGERLAND MISTERY 1	HAL	ARCADE/AVV	64	2
279178	EGGERLAND MISTERY 2	HAL	ARCADE	64	2	2806	EGGY	ROTHTEC	ARCADE	64	1
28189	EL CID		AVVENTURA	64	1	28272	EL MISTERIO DEL NILO	ZIGURAT	ARCADE/AVV	64	1
28379	EL MOTORISTA SID. 1		EDUCATIVO	32	1	28480	EL MOTORISTA SID. 2		EDUCATIVO	32	1
285116	EL PODERE OSCURO	ZIGURAT	SPAZIALE	64	1	286107	ELEVATOR ACTION	MSX INFO	ARCADE	32	1
28754	ELIDON	ORPHEUS	ARCADE/AVV	64	1	28886	ELITE	FIREBIRD	ARCADE	64	1
28987	ESVAY		ARCADE	32	1	29029	EUROGAME	TYNESOFT	SPORT	32	1
29199	EWOKS	LUCASFILM	ARCADE	64	1	29211	EX INNOVA	PROGRAMMER	ARCADE	64	1
2932	EXCHANGER	ASCII	ARCADE	32	1	29488	EXECUTIVE BASKETBALL	DINAMIC	SPORT	64	1
2953	EXERION	JALECO	PAZIALE	64	1	29619	EXERION 2		ARCADE	64	1
29788	EXOIDE-Z	CASIO	SPAZIALE	32	1	29890	EXPERT	KNIGHTS	ARCADE	32	1
29961	EXTERMINATOR	BYTEBUSTER	ARCADE	64	1	300162	F1 SFIRIT	KONAMI	FORMULA 1	64	2

## TIENI QUESTO ELENCO

INSIEME AI SUCCESSIVI CONTRIBUIRA' A FORMARE UNA COMPLETA ED INTERESSANTE RACCOLTA DI PROGRAMMI GIOCO E UTILITY CHE POTRANNO SODDISFARE OGNI TUA ESIGENZA

DESIDERO RICEVERE I SEGUENTI PROGRAMMI (MINIMO 5) DI CUI TRASCRIVO N° CODICE:

---



---



---



---



---



---



---



---



---



---

Tenendo conto che ogni gioco per MSX 1 costa £ 7.000, per MSX 2 £ 9.000, per ogni UTILITY £ 10.000, per ogni dischetto usato per contenere i programmi £ 2.500 cad., per cui dovrò riconoscervi la spesa totale di £ ..... più spese postali

NOME ..... COGNOME ..... VIA .....

..... C.A.P. .... CITTÀ ..... PROV. ( ..... )

TEL. ....

Inviare in busta chiusa a:

GRUPPO EDITORIALE INTERNATIONAL EDUCATION  
Viale Famagosta 75 - 20124 Milano - Tel. 02 / 89502288 r.a. - Telefax 02 / 8466834