

FEBBRAIO 1991
ANNO IV

MSX N 24

DISK

£ 15.000

- Spirit
- Gunfright
- Satan
- Bosconian
- Music Writer
- Ingegneria
- Finance Plus 7

giochi
news
utilità

L'UNICA DISK MAGAZINE
DEDICATA ALLO STANDARD MSX



DoMAEK

PRESENTS

WISDOM TO KILL



Small vertical text on the right side of the cover, likely a list of contents or contributors.

MSX DISK

SOMMARIO

- 2 Sommario – Sul disco
- Caricamento – Avvertenze
- 3 Editoriale – Abbonamenti
- 4 Spirit – Gunfright
- 5 Satan – Bosconian
- 6 Music Writer - Ingegneria
- 7 Finance Plus 7
- 8 Basic (Parte IX^a)
- 10 Dentro l'MSX (Parte VI^a)
- 12 Computer & Programmazione (Parte VII^a)
- 14 Il libro del mese

SUL DISCO

- 1 Spirit
- 2 Gunfright
- 3 Satan
- 4 Bosconian
- 5 Music Writer
- 6 Ingegneria
- 7 Finance Plus 7

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

Eccoci ad un altro numero, ed eccoci ad altre novità. Come nostra abitudine, anche questo mese vi proponiamo dei cambiamenti e, vedrete, sono anche abbastanza importanti.

Innanzitutto, sparisce il corso dedicato al Pascal e gli altri articoli, istruzioni comprese, sono ridotte e meno prolisse rispetto a ciò che vi abbiamo proposto negli scorsi numeri. Il motivo è uno solo: dobbiamo continuare a vivere! Sì, dobbiamo ridurre i costi per continuare a vivere e così ecco ridotte le pagine di Msx Disk. Non è bello, ma è necessario e siamo sicuri che ci capite e che continuerete a sostenerci come avete sempre fatto.

Dopo questo inizio quasi drammatico, parliamo di Club. Negli scorsi numeri vi abbiamo sempre dato notizie di un Msx Club nato, finalmente, in Italia. Su questo è necessario fare un'importante precisazione: MSX DISK e l'MSX CLUB ITALIA sono due entità distinte e non collegate in alcun modo.

Sì, abbiamo pubblicizzato questa associazione per darvi nuovi sbocchi nello sterile panorama Msx italiano, ma non rivolgetevi in redazione cercando notizie del club o per mettervi in contatto con esso. Tutto ciò che sappiamo su questo club ve lo diciamo in queste pagine, quindi è inutile che vi rivolgiate a noi. Se volete mettervi in contatto con questa associazione dovete scrivere direttamente a loro. L'indirizzo è: MSX CLUB ITALIA - C.P. 34 - 20075 LODI CENTRO (MI).

Chiuse queste parentesi, non resta altro che augurarvi buon lavoro e buon divertimento con 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.

SPIRIT

SPIRIT

Anche se dal nome non si capisce, questo è un gioco sportivo che vi porta nell'affascinante mondo dell'automobilismo. Una volta terminato il caricamento, vedrete scorrere sullo schermo la presentazione e quindi la demo. Per poter giocare dovrete premere la barra spaziatrice.

Premendo lo spazio accederete al primo di una serie di menù le cui opzioni si selezionano tramite i cursori e lo spazio.

Il primo menù offre due opzioni: GAME e COMMAND. Con la prima potete passare ad una serie di menù secondari e quindi al gioco vero e proprio, con la seconda opzione si accede ad un menù secondario che, tramite una password, permette di saltare direttamente ai livelli di gioco superiori.

Il secondo menù dopo Game/Command è composto da tre opzioni: 1 GIOCATORE, 2 GIOCATORI, MODALITA' BATTAGLIA.

Da questo menù si salta al successivo che permette di scegliere tra: GARA DI STOCK CARS, RALLY o FORMULA 3.

Infine, l'ultimo menù consente di scegliere tra AUTO GIA' PRONTE e AUTO DESIGNATE DAL GIOCATORE.

(Solo per MSX 2)

COMANDI

Tasti:

[SPAZIO] = Inizio gioco / Acceleratore

[CURSORI Su & Giù] = Selezione opzioni
/ Cambio marce

[CURSORI Destra & Sinistra] = Sterzo

GUNFRIGHT

GUNFRIGHT

Nel selvaggio west un nuovo eroe sta per scrivere il suo nome nella storia: voi!

Nei panni di un buffo sceriffo dovrete aggirarvi per la cittadina alla ricerca dei rapinatori che compaiono nel bando.

WANTED!

I ricercati vivi o morti - meglio morti! - vi frutteranno denaro e fama, ma i pericoli sono tanti e quindi dovrete avere mira precisa e velocità di estrazione.

(Per MSX 1 e 2)

COMANDI

Tasti:

[1] = Selezione numero di giocatori

[2] = Selezione Tastiera / Joystick

[3] = Inizio gioco

[I] = Fuoco

[Q] = Alto

[A] = Basso

[B] = Sinistra

[N] = Destra

SATAN

SATAN

Nei panni di un mitologico e muscolosissimo eroe dovete affrontare il male nella sua veste peggiore: il Diavolo in persona, SATANA! Con la vostra spada e col vostro coraggio - e anche

un po' di pazzia - dovrete aggirarvi nell'antro infernale alla ricerca di sua maestà il Diavolo, Re delle Tenebre.

La spada, ovviamente, è magica e dotata di potere divino così da consentirvi di decapitare il demone.

(Per MSX 1 e 2)

COMANDI

Tasti:

[1] = Inizio gioco

[2] = Selezione Joystick

[3] = Selezione tastiera / Definizione tasti di controllo

[Q] = Salto

[A] = Basso

[O] = Sinistra

[P] = Destra

[SPAZIO] = Lancia freccia magica & Usa spada

BOSCONIAN

BOSCONIAN

Tra i giochi spaziali, Bosconian è uno dei mitici e classicissimi che ci riportano alla storia degli arcade da bar.

La missione del giocatore è semplice ma difficile (ma che bel gioco di parole!): a bordo di un potente caccia interstellare si devono distruggere le basi spaziali nemiche presenti nel settore di galassia assegnato al caccia.

Le insidie non sono poche: rocce e mine stellari, missili automatici e astrocaccia nemici vi daranno la caccia.

Per distruggere le basi è possibile colpire direttamente il nucleo centrale oppure colpire tutte le componenti circolari che fanno parte della stazione stellare.

(Solo per MSX 1)

COMANDI

[CURSORI] = Movimenti

[SPAZIO] = Inizio gioco / Fuoco

MUSIC WRITER

MUSIC WRITER

Da diversi numeri non vi proponiamo un programma in grado di farvi sfruttare le ottime caratteristiche sonore del nostro standard, così ecco colmata questa lacuna.

- Come è possibile capire dal nome, il programma permette di scrivere la musica.

Grazie ad uno spartito simulato e ai comandi associati ai tasti funzione è possibile scrivere in note un pezzo per poi registrarlo, modificarlo, archivarlo o eseguirlo.

Per sfruttare al meglio il programma è consigliabile collegare il computer ad un amplificatore tramite l'uscita audio per il monitor.

Per controllare il programma dovrete usare i tasti funzione e i tasti numerici.

Sullo schermo di lavoro, nella parte bassa, troverete sempre le diverse funzione associate ai tasti da utilizzare.

(Per MSX 1 e 2)

COMANDI

Tastiera.

INGEGNERIA

INGEGNERIA

Questo è un programma dedicato all'ingegneria edile.

Una volta caricato il programma offre un menù principale composto da cinque punti:

- 1) CALCOLO CEMENTO ARMATO
- 2) CALCOLO TRAVI APPARTAMENTO
- 3) CALCOLO PLINTI
- 4) CALCOLO TRAVI APPOGGIATE
- 5) HELP

Premendo il tasto numerico corrispondente si seleziona l'opzione scelta che corrisponde ad un vero e proprio programma a se stante.

Il programma è corredato da una accurata parte di aiuto ed istruzione grazie alla quale è possibile ridurre notevolmente queste istruzioni.

Per qualunque istruzione relativa al programma fate riferimento all'help inserito nel programma.

Usando l'opzione di aiuto dovrete premere la combinazione di tasti [CTRL] ed [S] per fermare lo scorrimento del testo sullo schermo.

Per riattivare lo scorrimento sarà poi sufficiente premere la barra spaziatrice.

(Per MSX 1 e 2)

COMANDI

Tastiera.

FINANCE PLUS 7

FINANCE PLUS 7

Come 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 le volte scorse, diviso in tre parti e offre un menù principale con le seguenti opzioni:

- 1 - VERIFICA DELL'ESTRATTO CONTO;
- 2 - CRITICAL PATH METHOD;
- 3 - RITORNO AL DOS.

Nel file sorgente potrete trovare che la prima opzione viene espletata dalla parte che va dalla linea 1000 alla linea 1760 mentre l'analisi degli investimenti comuni è compresa tra la linea 2000 e la linea 2860. Come (quasi) sempre, dalla linea 3000 in poi trovano posto le routine di servizio.

VERIFICA DELL'ESTRATTO CONTO

Questo programma può rimuovere una considerevole mole di lavoro ogni volta che si renda necessario ricalcolare un conto corrente bancario. Se è il computer ad eseguire le addizioni e le sottrazioni, la possibilità che si verifichino degli errori di calcolo è praticamente inesistente.

Per utilizzare il programma è necessario introdurre il saldo finale del conto e l'ammontare di ogni deposito o prelievo effettuato dalla data del saldo. Dopo aver introdotto i dati relativi a tutti i depositi, si introduca uno zero; questo segnala al programma di passare alla sezione successiva, cioè alla introduzione dei dati riguardanti i prelievi e i pagamenti tramite assegno. Si introducono gli importi relativi agli assegni emessi e ai prelievi effettuati, quindi si batte ancora uno zero quando i dati da introdurre sotto questa voce sono terminati. Gli importi devono essere inseriti come numeri positivi; fa eccezione il saldo finale del conto che può essere anche un numero negativo. Se alla fine il conto non dovesse tornare, si controllino tutti gli importi introdotti per assicurarsi di non aver commesso qualche errore di battitura. Sono stati inseriti tutti gli importi degli assegni, dei depositi, dei prelievi effettuati dopo la data del saldo? E gli importi introdotti sono esatti? Se non si trova alcun errore, allora è consigliabile consultare l'agenzia presso cui è aperto il conto.

Esempio - Giulia sta controllando l'estratto conto inviatole dalla sua banca. In tale estratto conto non compare lo stipendio di 600.000 lire che la banca dovrebbe aver già ricevuto. Giulia inoltre ha compilato due assegni dopo la data di emis-

sione dell'estratto, uno di 87.320 lire ed uno di 250.000 lire. Se il saldo finale che compare sull'estratto conto è di 348.550 lire, il libretto degli assegni di Giulia riporta un saldo di 614.540 lire e le spese di servizio sul conto corrente sono di 3.310 lire, qual'è il saldo effettivo? L'estratto conto di Giulia è esatto? Risposta: il saldo effettivo a tutt'oggi del conto di Giulia è di 611.230. L'estratto conto è esatto.

CRITICAL PATH METHOD

Il Critical Path Method, altrimenti detto Metodo del Percorso Critico, è una tecnica usata per pianificare l'esecuzione di un progetto. Il programma qui presentato è basato su tale tecnica. Prima di utilizzarlo è consigliabile stendere un reticolo CPM e una tabella di priorità. Al momento in cui tracciate il reticolo, assicuratevi di includere anche le cosiddette 'attività fittizie'; tali attività non hanno una propria durata ma possono essere necessarie per indicare la precedenza di alcune attività su altre nell'esecuzione del progetto. I dati richiesti dal programma sono: il numero di attività previste dal progetto, la sequenza nodale, nonché durata e costi previsti per ciascuna attività. Si tenga presente che è consigliabile inserire i costi in migliaia di lire per problemi di spazio nella visualizzazione della tabella riassuntiva.

Al termine dell'introduzione dei dati il programma calcolerà l'istante iniziale, l'istante finale e il margine libero relativi a ciascuna attività, nonché la lunghezza del percorso critico e il costo totale dell'operazione. Una caratteristica di questo programma è la possibilità di rivedere il progetto iniziale modificando la durata e i costi delle varie attività. In questo modo avrete la possibilità di valutare le variazioni eventuali del percorso critico. Così come è strutturato questo programma consente di introdurre un massimo di 100 attività. Per cambiare tale numero massimo, è sufficiente modificare la linea 135 del programma come segue:

```
135 DIM A(n,2),S(n),F(n),E(n,2)
```

sostituendo alla lettera n il numero massimo di attività che intendete introdurre.

Esempio - La ditta Bianchi Valvole sta per iniziare la sua campagna di vendita nazionale, una operazione in cui i tre venditori della compagnia viaggiano attraverso la regione Lombardia cercando di effettuare il maggior numero di vendite nelle zone loro assegnate. Al termine dell'operazione i tre commessi si recano a Milano per un raduno di riepilogo. Nadia, direttrice delle vendite, vorrebbe sapere quando ognuno dei tre venditori dovrebbe partire, quanto tempo occorre a ciascuno per i viaggi di trasferimento e per le vendite e per quando prevedere il ritorno di ciascuno a Milano. Nella sua tabella di priorità figurano i rimborsi giornalieri per aiutarla a calcolare le spese di trasferimento.

Attività	Sequenza Nodale	Tempo (ore)	Costo
1. Giorgio va a Bergamo	1-2	2	30
2. Nadia va a Sondrio	1-3	3	40
3. Laura va a Brescia	1-4	3,5	49
4. Vendita a Bergamo	2-5	36	125
5. Vendita a Sondrio	3-6	48	320
6. Vendita a Brescia	4-7	48	125
7. Giorgio va a Pavia	5-8	3	40
8. Nadia va a Mantova	6-11	5	35
9. Laura va a Milano	7-10	2	30
10. Vendita a Pavia	8-9	16	90
11. Giorgio va a Milano	10-11	4	52
12. Laura va a Milano	9-11	2	30

Risposta: il tempo minimo richiesto per portare a termine la vendita è di 61 ore (lunghezza del percorso critico), e il costo totale in spese di viaggio è di 966 mila lire.

Riprendiamo là dove avevamo interrotto lo scorso numero, continuando a parlare di matematica al computer e, in particolare, delle funzioni più comuni disponibili anche con i BASIC meno potenti.

FUNZIONI TRIGONOMETRICHE SIN, COS, TAN, ATN

Il BASIC dispone di alcune funzioni specializzate per i calcoli trigonometrici: SIN, COS e TAN, che restituiscono rispettivamente il valore del seno, del coseno e della tangente dell'argomento specificato, ed almeno una delle funzioni inverse (di solito ATN, arcotangente), dalla quale è possibile ricavare tutte le altre.

Le funzioni trigonometriche del BASIC accettano l'argomento in radianti, non nei comuni gradi sessagesimali cui siamo abituati. Se scriviamo:

```
PRINT SIN(30) -> .988031623
```

Abbiamo calcolato il seno di un angolo di 30 radianti non di 30 gradi!

Poiché i 360 gradi dell'intera circonferenza corrispondono a 2 PIGRECO radianti, è possibile definire una costante di conversione tra gradi e radianti:

```
CC=(2*PI)/360
```

o, più semplicemente

```
CC=PI/180
```

dove PI sta per il valore di PIGRECO. In alcuni calcolatori, la variabile PI è RISERVATA e contiene già il valore di pi-greco. In altri, come il Commodore, esiste l'apposito simbolo sulla tastiera. Nei rimanenti casi, occorre definirla:

```
PI=3.14159265
```

Una volta definita la costante CC (naturalmente, si può usare qualunque nome di variabile) è possibile trasformare i gradi in radianti:

```
PRINT 30*CC -> .523598775
```

Abbiamo ottenuto il valore, in radianti, corrispondente a 30 gradi sessagesimali. Ora possiamo utilizzarlo per un'operazione trigonometrica:

```
PRINT SIN (.523598775) -> .5
```

o, saltando il passaggio intermedio:

```
PRINT SIN(30*CC) -> .5
```

Effettivamente il valore .5 (lo zero iniziale non viene stampato) corrisponde al seno di un'angolo di 30 gradi. Con l'aiuto di una costante di conversione possiamo dunque usare le funzioni SIN, COS e TAN con gli angoli sessagesimali, che ci sono più familiari (a parte i lettori matematici, che probabilmente preferiscono impiegare direttamente i radianti).

UN PRIMO PROGRAMMA GRAFICO

Approfittiamo delle funzioni trigonometriche per studiare un programmino che disegna sinusoidi sul video. Poiché

praticamente tutti i calcolatori impiegano la grafica in maniera differente, faremo ricorso ad una forma di grafica limitata, ma sicuramente accettabile da ogni computer: disegnare con i caratteri.

Più precisamente, disegneremo una sinusoide per mezzo di asterischi.

Ecco il programma:

```
100 REM --- SINUSOIDI ---
120 REM PARAMETRI
140 CI=5:REM CICLI
150 PA=30:REM NUMERO PASSI
160 AM=30:REM AMPIEZZA (MAX. 30)
180 REM COSTANTI
200 PI=3.14159265
210 CC=(2*PI)/360:REM COSTANTE CONVERSIONE
220 CS=20:REM CENTRO SCHERMO
230 AM=AM/2
250 REM DISEGNO
270 FOR C=1 TO CI:REM PER CI VOLTE
280 IN=360/PA:REM INCREMENTO
290 FOR A=0 TO 360-IN STEP IN
300 P=CS+AM*SIN(A*CC):REM POSIZIONE
310 PRINT TAB(INT(P+.5));"*"
320 NEXT A:REM PROSSIMO ANGOLO
330 NEXT C:REM PROSSIMO CICLO
```

Se avete un calcolatore a disposizione, provate ad introdurre il programma ed osservate la sinusoide disegnarsi sul video (a 90 gradi rispetto al disegno classico, cioè con l'immaginario asse delle ascisse porto verticalmente al centro dello schermo).

La DEFINIZIONE non è eccezionale, ma è il massimo ottenibile senza fare ricorso ad istruzioni grafiche specializzate (o caratteri grafici speciali).

Le linee dalla 140 alla 160 definiscono alcuni parametri. Il concetto di PARAMETRO, molto importante nello studio e nella pratica della programmazione, ha vari aspetti. In questo caso, un parametro è una variabile il cui valore non cambia durante un'esecuzione del programma. Cosa lo distingue, allora, da una costante (intendendo con questo termine anche una variabile a cui è stata assegnata una costante, come PI nella linea 200)?

La differenza è che il valore di un parametro può essere cambiato da un'esecuzione all'altra del programma per ottenerne un diverso comportamento (in questo caso, occorre cambiare delle linee di programma). In effetti, sarebbe stato corretto parlare di parametri anche se i valori delle tre variabili CI, PA ed AM fossero stati introdotti per mezzo di istruzioni di INPUT. Il valore di una costante, invece, non può mai essere alterato (avrebbe senso cambiare il valore di PI?).

Dunque, i tre parametri stabiliscono rispettivamente quanti cicli di sinusoidi devono essere disegnati (CI), da quanti asterischi è composta ciascuna sinusoide (PA) e la sua ampiezza massima, da picco a picco (AM). Il programma si comporta diversamente, a seconda del valore di questi parametri. Provate a cambiarli ed osservate i risultati.

Le linee dalla 200 alla 230 definiscono le COSTANTI usate nel programma. Nei computer che possiedono già il valore di PI, la linea 200 è inutile e la 210 può essere modificata con l'eventuale simbolo già pronto.

Come abbiamo già visto, la linea 210 può essere semplificata (dividendo per due numeratore e denominatore);

l'abbiamo lasciata solo per chiarezza.

La linea 220 definisce la posizione di centro schermo; in realtà, poichè il numero di caratteri è pari, si tratta della posizione appena a sinistra del centro. Se disponete di un computer con video ad 80 colonne, potete aumentare il calore di CS a 40 (o ridurlo, nel caso contrario).

La 230 non calcola una costante, ma modifica il valore del parametro AM, in modo che indichi l'ampiezza di un solo picco della sinusoide, e non quella da picco a picco, per semplice comodità di calcolo. Non possiamo dire di aver modificato il valore di un parametro, in realtà avremmo potuto usare un'altra variabile e lasciare inalterato AM.

La linea 270 apre il ciclo più esterno, che specifica quante sinusoidi disegnare (in base al parametro CI). La 280 calcola l'INCREMENTO, in gradi, tra un passo e l'altro, cioè ogni quanti gradi va calcolato e disegnato un nuovo asterisco. Il valore si ottiene facilmente, dividendo l'intero ciclo (360 gradi) in PA parti.

La linea 290 apre il ciclo più interno: in ciascuna ripetizione di questo viene disegnato un asterisco. A è l'angolo (in gradi), che parte da 0 ed arriva ad un passo dalla chiusura del ciclo (per non sovrapporsi con la sinusoide seguente). L'incremento, specificato dalla STEP, è appunto quello calcolato dalla linea 280. Ad ogni ciclo, dunque, l'angolo A viene aumentato di IN gradi.

La linea 300 calcola la posizione orizzontale, sul video, dove dovrà essere stampato l'asterisco. Tale posizione è ottenuta partendo dal valore di centro schermo (CS, corrispondente all'asse delle ascisse) ed aggiungendo il valore calcolato con la funzione SIN in corrispondenza del corrente angolo A.

Poichè l'angolo è espresso in gradi, occorre convertirlo in radianti (moltiplicarlo per CC) prima di usare la funzione SIN. Inoltre, poichè il seno di un angolo è compreso tra -1 e +1, il valore ottenuto va moltiplicato per l'ampiezza massima AM (espressa in caratteri). Se AM vale 19 (ricordate che l'abbiamo divisa per due alla linea 230), il valore calcolato varierà tra 19 e +19.

Aggiungendo questo valore alla posizione di centro schermo (CS), otteniamo la posizione orizzontale (cioè l'ordinata, visto che siamo ruotati di 90 gradi rispetto al classico piano cartesiano) dove stampare l'asterisco, espressa da un numero variante tra CS-19 e CS+19. Ad esempio tra 1 e 39 se CS vale 20 e AM 10.

La linea 310, finalmente, stampa l'asterisco alla posizione specificata, utilizzando la funzione TAB. Teoricamente la TAB è in grado di accettare anche valori frazionari (considerando solo la parte intera), ma è meglio usare un valore intero, dato che stiamo parlando della posizione di una colonna di caratteri sul video.

Per ottenere un valore intero abbiamo impiegato la funzione INT, che restituisce la parte intera dell'argomento. L'aggiunta di 0.5 merita un discorso a parte, per cui ne parleremo nel prossimo paragrafo.

Intanto, provate a toglierla e guardate cosa succede (con AM=30, la cosa è ancora più evidente).

Le linee 320 e 330 chiudono, rispettivamente, il ciclo interno e quello esterno.

APPROSSIMAZIONI

Se avete provato a modificare la linea 130, togliendo la somma con .5, avrete notato come la sinusoide diventi più "brutta", meno simile ad una sinusoide corretta e per di più asimmetrica.

Il primo di questi problemi è connesso alla tecnica che ab-

biamo usato per rappresentare la sinusoide: poichè non è possibile impiegare una curva continua (nessun calcolatore digitale è in grado di farlo), abbiamo usato una serie di punti (nel nostro caso, piuttosto grossolani: gli asterischi). Abbiamo cioè CAMPIONATO (ricavato) il valore della sinusoide per determinati angoli, cercando di rappresentarne quanto più fedelmente possibile il valore così ottenuto. Questa tecnica è affine a quella impiegata per convertire un segnale elettrico analogico (come l'uscita di un rivelatore di temperatura, o di un microfono) in un numero digitale, o in una serie di numeri ricavati ad intervalli regolari di tempo (CONVERSIONE A/D, o Analogico/Digitale). In tutti i casi di CAMPIONAMENTO DIGITALE si pone un problema.

Avendo a disposizione solo un insieme finito di possibili numeri (nel caso della nostra sinusoide, da 1 a 39, corrispondenti alle relative colonne dello schermo video dove può essere stampato l'asterisco), come ci si deve comportare con i valori non interi (per esempio 1.1 o 17.436752)? Qualunque valore intero venga impiegato si tratta sempre di un'APPROSSIMAZIONE, dunque di un errore introdotto, per quanto inevitabile. I nostri asterischi, infatti, non disegnano esattamente una sinusoide reale, non potendo stare in posizioni intermedie tra le colonne video. Per ridurre al minimo questo

ERRORE DI CAMPIONAMENTO,

o

ERRORE DI DIGITALIZZAZIONE,

bisogna che un valore non intero venga approssimato AL VALORE INTERO PIU' VICINO.

Ad esempio, i numero da 3.5 (incluso) fino a 4.5 (escluso) vanno approssimati a 4. In questo caso, l'errore non supera mai la metà dell'intervallo di campionamento (cioè la distanza di 1/2 carattere).

Se impieghiamo semplicemente la funzione INT per convertire il numero reale (cioè non intero) corrispondente alla posizione esatta in cui dovrebbe essere stampato l'asterisco, troviamo che l'errore massimo raddoppia.

Infatti, poichè INT approssima all'intero inferiore, e non all'intero più vicino, il valore 3.999999999 diventa 3, anzichè 4.

Per di più, poichè INT approssima all'intero inferiore anche nel caso di numeri negativi, succede ad esempio che -.000000001 diventa -1 (e questa è la causa dell'evidente asimmetria al centro).

Aggiungendo 1/2 (cioè .5) al valore reale prima di eseguire l'INT, tutto va a posto. Se non ci credete, provate con gli esempi citati.

Quindi l'operazione INT(X+.5) approssima il valore reale X all'intero più vicino.

(continua)

DENTRO L'MSX (Parte VI^a)

Nel nostro prossimo esempio viene illustrato come copiare l'insieme dei caratteri maiuscoli degli sprite per ottenere un testo in formato non standard:

```
10 SCREEN1,1:REM sprite espansi 8*8
20 FOR X=0 TO 207
30 VPOKE 14856 + X,VPEEK(520+X)
40 NEXT
```

I numeri di modello dal 65 in su sono stati usati deliberatamente allo scopo di mantenere una corrispondenza numerica tra il numero del modello di sprite e il codice del carattere. Ciò permette infatti di visualizzare con facilità delle stringhe:

```
10 A$="MSX":FOR X=1 TO 3
20 PUT SPRITE X,(18*X,40),11,ASC(MID$(A$,X,1))
30 NEXT
```

I due restanti comandi per la manipolazione degli sprite sono:

1. ON SPRITE GOSUB
2. SPRITE ON

consentono di rilevare e gestire collisioni tra sprite. Una collisione si verifica quando si sovrappongono parti non trasparenti di due sprite. Vi sono due limitazioni sull'informazione che è possibile dedurre da ogni collisione:

1. Si possono rilevare solo collisioni sprite-sprite e non tra sprite e fondo.

2. Non è disponibile né una funzione BASIC né un registro del VDP che indichi quali sono i piani degli sprites coinvolti nella collisione.

L'istruzione SPRITE ON attiva la procedura di controllo che quindi viene eseguita dopo ogni istruzione BASIC. Nessun controllo viene effettuato nella modalità ed esecuzione diretta del BASIC. Quando avviene una collisione, viene eseguita un'istruzione GOSUB alla routine che è stata specificata nell'istruzione ON SPRITE GOSUB.

L'istruzione SPRITE OFF annulla ogni controllo riferito alle collisioni tra sprite. L'istruzione SPRITE STOP sospende l'effettiva attivazione della routine di gestione della collisione, ma il controllo viene mantenuto attivo. Infatti se avviene una collisione, essa viene "ricordata" fino a quando viene eseguita un'istruzione di SPRITE ON nella chiamata della subroutine.

Un'istruzione di SPRITE STOP viene automaticamente eseguita ogni volta che si inizia l'esecuzione della routine di gestione delle collisioni. Analogamente quando si giunge al RETURN, viene eseguita una SPRITE ON- a meno che la routine stessa non contenga un'istruzione di SPRITE OFF. E' importante ricordare che l'istruzione CLEAR effettua una SPRITE OFF.

Accade spesso che il movimento degli sprite sia di natura fissata e ripetitiva. Vale a dire che la loro posizione deve solo essere incrementata e decrementata ad intervalli di tempo prefissati. IL BASIC MSX comprende un potente insieme di comandi per far sì che, durante l'esecuzione di un programma, ad intervalli di tempo regolari, venga effettuato un salto ad una determinata routine.

L'intervallo di tempo viene specificato in multipli di 1/50 di secondo (cioè la frequenza tipica di interruzione del VDP).

Il formato dei suddetti comandi è simile a quello dei comandi per la gestione delle collisioni:

1. INTERVAL ON
2. ON INTERVAL=X GOSUB

Anche in questo caso esistono le varianti INTERVAL OFF e INTERVAL STOP: quest'ultima viene eseguita all'ingresso della routine di gestione.

Se si specificano intervalli molto brevi, può verificarsi un leggero allungamento dell'intervallo effettivo a causa del controllo che viene effettuato dopo l'esecuzione di ogni istruzione BASIC.

Due gruppi di comandi simili a quelli fin qui illustrati sono:

1. ON STOP GOSUB e STOP ON/OFF/STOP
2. ON KEY GOSUB linea x, line y ... e KEY(x) ON/OFF/STOP

La struttura ON STOP GOSUB/STOP ON va usata con una certa cautela poichè può impedire il ritorno dalla modalità BASIC di esecuzione di un programma a quella di esecuzione diretta dei comandi.

Per esempio:

```
10 ON STOP GOSUB 40:STOP ON
20 PRINT "NOSTOP":GOTO20
40 RETURN
```

è un programma che non può essere interrotto. Infatti quando si vuole che l'esecuzione di un programma non possa essere interrotta, è sufficiente che la linea 10 appaia dopo un'istruzione CLEAR.

GRAFICA AD ALTA RISOLUZIONE

Il BASIC MSX ha sette istruzioni che possono essere utilizzate solo nelle due modalità grafiche:

- 1) CIRCLE
- 2) DRAW
- 3) LINE
- 4) PAINT
- 5) PSET
- 6) PRESET
- 7) POINT

E' ovvio però che queste istruzioni producono un risultato abbastanza scadente nella modalità multicolore dove la massima risoluzione è costituita da blocchi 4 * 4 pixel.

Non è possibile visualizzare caratteri sullo schermo grafico usando le normali istruzioni. Viceversa i caratteri possono essere visualizzati come dati da file:

1. OPEN "GRP:"FOR OUTPUT AS#1
2. PRESET (40,40)
3. PRINT# 1, "Testo grafico"

Si noti che, a meno che si sia previsto un numero maggiore di file con l'istruzione MAXFILES, il numero di file deve essere 1. Nella modalità multicolore ogni lettera è approssimativamente di 2,5 cm quadrati.

Il "cavallo di battaglia" delle istruzioni per la grafica è la DRAW nel seguente formato:

DRAW "comandi a lettera singola"

I 13 comandi disponibili costituiscono quello che la Microsoft definisce un "micro linguaggio per la grafica". Tramite essi si possono tracciare linee secondo uno specifico passo a sinistra, a destra, in alto, in basso o in diagonale dal punto di riferimento corrente.

Può essere utilizzato uno qualsiasi dei 16 colori disponibili. L'istruzione in questione è più comprensibile se si considera un esempio:

```
DRAW "D40R40U40L40"
```

Si ottiene un quadrato tracciando dapprima una linea di 40 unità di lunghezza verso il basso (D40), poi una verso destra (R40), poi una verso l'alto (U40) ed infine verso sinistra (L40).

I principali comandi per determinare la direzione sono:

```
U  
H E  
L + R  
G F  
D
```

e "Cx" viene usata per impostare il colore.

Per tracciare una linea fino ad una posizione specificandone le coordinate, si deve usare M X,Y. In alternativa per muoversi fino ad una posizione di cui siano note le coordinate relative al punto di riferimento corrente, è possibile far precedere alla X e alla Y il segno "+" o il segno "-".

L'unità di misura dei movimenti può essere impostata dal comando Sn dove n è compreso nell'intervallo 0 - 255 (default = 4).

Dividendo questo numero per quattro si ottiene il numero dei pixel per ogni unità di spostamento.

Se in un comando viene specificata una posizione sullo schermo, esterna all'intervallo 32768 + 32767, non viene segnalato alcun errore e viene assunta la posizione più esterna possibile.

Esistono due varianti per ciascun comando di movimento:

1. Usando come prefisso al comando una lettera "B" si provoca lo spostamento del cursore, senza che però venga tracciata alcuna linea.

2. Usando come prefisso "N" si riporta il cursore grafico alla posizione iniziale, dopo il completamento dell'esecuzione del sottocomando.

Le direzioni effettive possono essere ruotate in senso antiorario per multipli di 90 gradi con il comando An, dove n deve essere compreso tra 0 e 3, come viene illustrato di seguito:

```
0  
1 + 3  
2
```

Quindi il comando DRAW "A3L40" serve per tracciare una linea verso destra per quaranta unità di spostamento.

Il sottocomando X consente d'includere una variabile stringa nella linea di comando.

La X precede il nome della variabile che viene seguito da un punto e virgola: per esempio:

```
ST="M-20,+40";DRAW"XST$;"
```

Questo sottocomando non è però necessario qualora non si usino i delimitatori: DRAW ST\$. Inoltre anche una variabile numerica può essere usata in una linea di comando: è necessario farla precedere da un segno "=" e farla seguire da un ";":

```
DRAW "U=Y;".
```

Il formato del comando LINE è il seguente:

```
LINE(X1,Y1)-(X2,Y2),colore
```

e provoca il tracciamento di una linea tra le due coordinate indicate; ciascuna coordinata può essere specificata in modo alternativo usando la variante STEP dell'istruzione. Se il comando è fatto seguire da "B" si ottiene il disegno di un rettangolo e se si usa "BF" il rettangolo viene riempito.

Il comando CIRCLE consente di ottenere diagrammi piuttosto sofisticati in modo semplice.

Una qualsiasi ellisse può essere specificata e disegnata, interamente o parzialmente.

Il formato del comando di base è:

```
CIRCLE (X,Y),raggio
```

con le opzioni:

colore, angolo di partenza, angolo finale, raggio apparente.

Le coordinate X e Y specificano il centro del cerchio; usando il formato STEP (X,Y) è possibile definire tale punto relativamente al riferimento corrente.

Quando si usano l'angolo di partenza e quello finale devono assumere valori tra 0 e 2 pi greco. Viene accettato anche un valore negativo, ma provoca la visualizzazione di ulteriori linee tra il centro dell'ellisse ed il punto di partenza e di arrivo.

Il raggio apparente è il rapporto tra i raggi orizzontale e verticale.

Il comando PAINT è sufficientemente auto esplicativo; il suo formato è:

```
PAINT (X,Y) oppure PAINT (X,Y), colore
```

provoca il riempimento in colore di una qualsiasi figura che racchiuda le coordinate in questione. Se non viene specificato il colore, viene usato il colore corrente in primo piano.

Si noti che nella modalità HRG, il colore del bordo e quello specificato in PAINT devono essere i medesimi, altrimenti viene ricolorato l'intero schermo.

Nella modalità multicolore si può usare la forma:

```
PAINT(X,Y), colore di riempimento, colore del bordo.
```

(continua)

CARATTERIZZAZIONE ASTRATTA E CONCRETA DEI DATI

Molte persone, anche del settore informatico, interrogate sul significato del termine astrazione, sono portate a rispondere:

“cosa solo pensata”

o anche

“argomento poco concreto”.

In realtà noi tutti usiamo quotidianamente delle astrazioni senza rendercene conto, come ad esempio i numeri, e l'uomo avrebbe fatto ben poca strada dall'età della pietra se non avesse esercitato la pratica dell'astrazione.

L'informatica, più di qualsiasi altra scienza, e in particolare la programmazione, ha il grosso merito di aver messo in luce la grande potenza di questa capacità mentale dell'uomo che trova, proprio nel calcolatore, un punto di riferimento concettualmente banale, ma concreto per dare peso a molte delle proprie astrazioni.

Una volta dato peso, mediante un lavoro di programmazione, a una di queste astrazioni, è possibile utilizzarla per concretizzare altre astrazioni di livello più elevato e, generalizzando questo procedimento, disporre di astrazioni sempre più espressive, il cui uso ci consente e ci consentirà di risolvere problemi sempre più complessi.

Non nascondiamo che si tratta di un concetto che richiede un'accurata riflessione anche perchè non siamo abituati a pensare a cose astratte in questi termini.

Ad esempio, nell'insegnamento della matematica, che è la scienza delle astrazioni per eccellenza, anche quando si dà loro un corpo, per poterne parlare o per manipolarle, si dimentica di sottolineare la differenza fra le astrazioni e la loro rappresentazione, che finiscono spesso per identificarsi.

Ora si svilupperà il concetto di astrazione principalmente in tre direzioni che, pur presentando forti legami, si cercherà di affrontare separatamente:

Astrazioni sui dati: cui è dedicata la lezione.

Astrazioni sulle procedure (prossime lezioni).

Astrazioni sugli esecutori (prossime lezioni).

Un concetto comune a tutte queste astrazioni è quello di implementazione, un termine tipicamente informatico coniato per indicare la soluzione al problema di dare un corpo alle astrazioni.

L'uso integrato, ordinato e sistematico di queste astrazioni è la grande scoperta dell'informatica.

E' nostra opinione che la rapidità con cui si registreranno i successi dell'informatica nei prossimi anni dipenderà, in gran parte, dalla misura in cui i nuovi linguaggi di programmazione sapranno stimolare le capacità di astrazione dell'uomo.

Ancora, è nostra convinzione che è proprio su questo terreno che potremo sempre distinguere le possibilità dell'uomo da quelle delle macchine.

I CONCETTI DI ASTRAZIONE E DI IMPLEMENTAZIONE

Ormai è chiaro che l'uso, molto frequente in informatica, del termine

“rappresentazione”

è legato al problema di disporre concretamente di oggetti astratti appartenenti a un ipotetico mondo concettuale per poterli elaborare secondo determinate regole.

Anche se tutta la lezione sarà dedicata ai dati o ai cosiddetti dati astratti, le considerazioni che seguono servono a comprendere il più generale concetto di astrazione.

Si badi bene che l'aggettivo “astratto” riferito a determinate entità, come ad esempio i dati, sottintende chiaramente la possibilità, o la presenza, di due diversi livelli di visione delle entità: appunto, uno astratto e un altro concreto.

Tuttavia, quando tale visione è unica, l'aggettivo “astratto” scompare in modo abbastanza naturale.

Tanto per intenderci, nessuno di noi avverte la necessità di sapere in che modo vengono rappresentati i numeri “all'interno” delle comuni calcolatrici tascabili proprio perchè, come utenti, abbiamo un'unica visione che è quella “decimale”.

Senza rendercene conto, facciamo dell'astrazione nei riguardi della rappresentazione interna dei numeri tanto che nessuno si sognerebbe di dire che lavoriamo con dati astratti.

Chi ha progettato la calcolatrice ha però certamente avuto due visioni distinte dei dati: una esterna, decimale, destinata all'utente, e una “concreta”, diciamo binaria, che era l'unica che poteva utilizzare per effettuare le varie operazioni sui numeri.

Questo significa che il progettista ha dovuto risolvere il problema di rappresentare i numeri “decimali” nel sistema binario, e viceversa, utilizzando espliciti algoritmi di conversione per consentire la comunicazione da un livello all'altro.

Analogamente, il manuale di riparazione della calcolatrice parlerà un linguaggio profondamente diverso da quello destinato “all'uso”; in esso compariranno termini come:

“circuiti integrati”
“memorie”
“cavicorda”, ecc.

In altre parole, vogliamo rilevare il fatto che il termine “astratto” è sempre relativo a un livello sottostante “concreto” che viene introdotto dal fatto che tale astrazione non è disponibile, e che “scompare” quando si sarà stabilita e realizzata la corrispondenza con gli oggetti concreti, cioè quelli disponibili. Vogliamo insistere su questo fatto perché il programmatore svolge praticamente sempre il ruolo svolto dal progettista della calcolatrice tascabile.

Egli è chiamato a permettere una visione astratta da parte degli utenti sia nei riguardi dei dati, sia dei dettagli della soluzione dei problemi, secondo il detto “premi un bottone ed otterrai tutti i risultati”.

E' lo stesso principio in base al quale non abbiamo bisogno di sapere come è fatta internamente un'automobile per poterla usare.

In generale, la visione utente, cioè quella implicitamente “astratta”, è quella interessata a conoscere il comportamento esterno delle cose e non il modo con cui questo si può raggiungere.

L'utente deve quindi essere messo in grado di comprendere il comportamento di un dispositivo fisico o concettuale senza dover ricorrere ai dettagli di una sua realizzazione concreta che, nel gergo informatico, assume il nome di

“implementazione”.

In questo caso la visione dell'utilizzatore viene detta

a “scatola nera”

proprio per indicare che non interessa sapere come sia fatta dentro.

Nessun manuale d'uso di un televisore scriverebbe: “per regolare il volume ruotare il cursore del potenziometro connesso all'ingresso dell'integrato XXX” per non compromettere la commerciabilità del prodotto.

Anzi, per garantire la separazione fra visione astratta e visione concreta, questi dispositivi domestici vengono rinchiusi in involucri, spesso sigillati, che servono a proteggere l'utente dalle “misteriosità” della macchina e, la macchina da manomissioni inopportune o accidentali da parte dell'utente.

L'utente può e deve servirsi di un dispositivo solo attraverso i comandi previsti per il suo uso.

Potrà sembrare strano, ma questi principi di dominio comune non sono stati immediatamente patrimonio della programmazione; anzi, ancor oggi, vi sono molte persone del settore che ignorano questi aspetti.

La causa di ciò va probabilmente ricercata nel seguente modo di pensare da parte dell'implementa-

tore (in pratica il risolutore del problema):

“Siccome non sono io l'utente e sono io a stabilire come far funzionare le cose, non posso avere delle visioni astratte delle cose anzi, il fatto di non aver barriere a livello concreto, mi concede ampie libertà sul modo migliore per farlo”.

In questo modo però egli si esclude anche la possibilità di usare delle astrazioni per dominare meglio la complessità del problema, preoccupandosi subito di dettagli irrilevanti, almeno nelle prime fasi del progetto, che gli possono impedire una visione globale del problema.

Già nella stessa elettronica, l'avvento dei circuiti integrati ha introdotto automaticamente un livello di astrazione nei riguardi del progettista di dispositivi elettrici; egli li può utilizzare tenendo semplicemente conto del loro comportamento esterno.

Ciò ha contribuito in modo sostanziale a determinare quel crescente sviluppo delle applicazioni elettroniche cui assistiamo ogni giorno.

Ma, mentre nell'elettronica, e in altre branche tecnologiche, le “scatole nere” che separano la visione esterna astratta da quella interna concreta assumono il carattere di un involucro fisico realmente inaccessibile, non si può dire altrettanto per la programmazione.

In questo caso le scatole nere sono solo immaginarie o espresse da certi vincoli linguistici.

Le difficoltà non stanno nell'usare le astrazioni messe a disposizione quanto piuttosto nell'intuire e nel comprendere che introducendone di nuove si può crescere meglio più e più in fretta, prendendo così coscienza delle differenze fra astratto e concreto nei termini appena esposti.

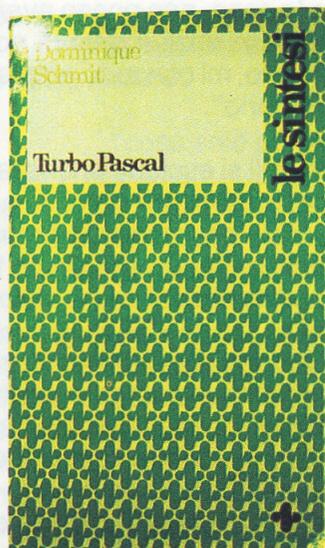
Le astrazioni costituiscono un importante punto di contatto fra informatica, sia teorica che pratica, e matematica; l'informatica se ne serve continuamente ma ha la necessità pratica di realizzarle concretamente attraverso il processo creativo di implementazione.

Astrarre in matematica, come in informatica, significa pensare alle cose senza la necessità di fare riferimento ad oggetti concreti che già “possediamo” nella nostra mente.

Si tratta forse una delle capacità più potenti dell'intelletto umano ed è l'indice migliore per saggiare l'attitudine di una persona a “crescere” nell'informatica.

(continua)

IL LIBRO DEL MESE



TURBO PASCAL - SINTESI

di Dominique SCHMIT

Edito da: Tecniche Nuove

Pagine: 212

Prezzo: 18.000 lire

Questo mese vogliamo proporvi una sintesi che ci è sembrata molto chiara e ben fatta. Com'è chiaro, questo testo parla di Turbo Pascal e, anche se non aggiornato all'ultimissima versione 6 che sta per uscire, riesce a dare un'ottima visione globale illustrando in maniera chiara e decisamente sintetica tutte le funzioni e i comandi disponibili con il compilatore della Borland. Questo libro è rivolto ai programmatori in Turbo Pascal con l'obiettivo di offrire risposte rapide ed esaurienti a qualsiasi genere di domande su questo linguaggio.

La padronanza completa di un linguaggio, comunque, si ottiene soltanto con l'uso: nessuna lettura può sostituire l'esperienza pratica. Restano sempre dei punti oscuri: perchè si è dimenticata la sintassi di un certo comando, il valore massimo di un certo parametro oppure perchè non si è mai provato a usare una certa caratteristica del linguaggio.

Inoltre, capita molto spesso che il programmatore di qualsiasi livello, cioè studente, hobbista o professionista, utilizzi

diversi linguaggi di alto livello (Pascal o C), di basso livello (Assembler) o specializzati (Prolog o dBASE).

Nascono allora nuovi problemi: si può chiamare il sottoprogramma scritto in un linguaggio da un programma scritto in un altro linguaggio? Quello che è fattibile in questo linguaggio è possibile anche nell'altro? E così via.

In definitiva, il programmatore ha di fronte due tipi di domande: su un determinato elemento del linguaggio (sintassi di un comando, effetto di una parola chiave, causa di un certo errore di esecuzione) e sul modo in cui ottenere il risultato desiderato. Questo testo è così stato concepito con l'intento di trovare facilmente una risposta alle domande di entrambi i tipi. Per questo, il lettore può trovare la pagina che gli interessa in tre modi diversi.

Innanzitutto, può consultare l'indice degli argomenti all'inizio del libro, che permette di passare in rassegna gli argomenti trattati nei vari capitoli, nell'ordine in cui sono esposti.

In secondo luogo, può consultare l'indice delle parole chiave di Turbo Pascal alla fine del libro. Questo indice è in ordine alfabetico, quindi consente di individuare rapidamente un elemento specifico del linguaggio.

Infine, si può sfogliare il libro. L'intestazione in cima ad ogni pagina indica l'argomento trattato, quindi permette di localizzare facilmente l'argomento desiderato. All'interno di ogni capitolo le procedure e le funzioni sono raggruppate per argomento e ordinate alfabeticamente.

Di ogni procedura e funzione è data la sintassi, la definizione di ogni parametro con il tipo e i valori massimo e minimo, la descrizione, gli eventuali commenti e rimandi ad altri capitoli.

In conclusione, è un buon testo seppure non aggiornato alle ultime versioni, lacuna che probabilmente verrà colpata con le nuove edizioni. Per il resto è un libro molto utile e di facilissima consultazione. Il costo poi non è decisamente proibitivo ma ragionevolmente contenuto.



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	TIPO	Kb	MSX	CODICE	NOME PROGRAMMA	AUTORE	TIPO	Kb	MSX
10175	BLITZ!	GE. INT. ED.	ARCADE	32	1	10231	BLOCADE RUNNER	TOSHIBA	SFAZIALE	64	1
10315	BLOCKBUSTER	KUMA	ARCADE	64	1	104104	BLOW-UP	PANTERSOFT	ARCADE	32	1
10541	BMX REKENCROSS	LONGMAN	ARCADE/EDU	32	1	10641	BMX SIMULATOR	CODEMASTER	ARCADE	64	1
10719	BOGGY '84	COLPAX	ARCADE	32	1	10840	BOING	IDEALOGIC	ARCADE/EDU	32	1
10958	BOKOSUKA WARS	ASCII	ARCADE/SIM	32	1	1100	BOMBER	HUDSON	ARCADE	32	1
11182	BOMULUS	TEKNOPISTE	AVV/ARCADE	32	1	11274	BOOGA BOO	QUICKSILVA	ARCADE	32	1
11322	BOOGIE WOOGIE	AMFLE	ARCADE	32	1	11474	BOOMERANG	ASCII	ARCADE	32	1
115104	BOOTY	EUROSOFT	ARCADE	32	1	11627	BOP!	JONES	ARCADE	64	1
1179	BOSCONIAN	NAMCO	SFAZIALE	64	1	1187	BOULDERDASH	ORPHEUS	ARCADE	32	1
11923	BOULDERDASH 2	ORPHEUS	ARCADE	32	1	12061	BOUNCE	DE-SOFT	ARCADE	64	1
12120	BOUNCING BLOCK	INDESCOMP	PENGO	32	1	12250	BOUNDER	GREMLIN	ARCADE	64	1
12313	BOXING	KONAMI	SPORT	64	1	124167	BREAK IN	METHODIC	ARCADE	64	1
12575	BREAKER	RADARSOFT	ARKANOID	64	2	12652	BREAKER	INDESCOMP	ARCADE	64	1
12776	BREAKOUT	ASCII	CLASSICO	32	1	12841	BREAKOUT	TOSHIBA	ARCADE/AVV	64	1
12932	BRIAN JACK 1	MARTECH	SPORT	0	1	13032	BRIAN JACK 2	MARTECH	SPORT	0	1
13143	BRIDGE	NICE IDEAS	CARTE	32	1	13257	BRUCE LEE	COMPTIQ	ARCADE/AVV	64	1
13334	BUBBLER	ACG	ARCADE/AVV	64	1	134103	BUBBLES	BUBBLE-GEE	ARCADE	32	1
1359	BUCK ROGERS	SEGA	ARCADE	64	1	1362	BUTAN PANTS	HAL	ARCADE	64	1
13750	BUFFZOFF	ELECTRIC	ARCADE	64	1	138100	CABBAGE PATCH KIDS	KONAMI	ARCADE	64	1
13916	CAMELOT WARRIOR	DINAMIC	ARCADE/AVV	64	1	14039	CAN OF WORMS	LIVEWIRE	SCACCHI	32	1
14113	CANDOO NINJA	ASCII	ARCADE/AVV	64	1	1427	CANNON BALL	HUDSON	ARCADE	32	1
1437	CANNON FIGHTER	MOORWOOD	ARCADE	32	1	14476	CAPITAN CHEF	COPAX	ARCADE	32	1
14584	CAPITAN SEVILLA 1	DINAMIC	ARCADE/AVV	64	1	14684	CAPITAN SEVILLA 2	DINAMIC	ARCADE/AVV	64	1
14782	CAR FIGHTER 2	CASIO	ARCADE	64	1	1483	CAR JAMBOREE	OEC	ARCADE	64	1
14922	CAR RACE	AMFLE	ARCADE	32	1	150103	CARRY	TM SOFT	ARCADE	32	1
1510	CASTLE	ASCII	ARCADE/AVV	64	1	15273	CASTLE EXCELLENT	ASCII	ARCADE/AVV	64	1
15379	CATBOY	CASIO	ARCADE	64	1	154101	CETUS	ARCADE	32	1	
15813	CHACK'N'POP	TAITO	ARCADE	32	1	1560	CHAMP	ASSEMBLER	32	1	
15799	CHAMPION HOCKEY	SEGA	SPORT	64	1	15857	CHAMPIONS	PONY	SPORT	64	1
159188	CHESS	KUMA	SCACCHI	32	1	16047	CHESS	B.U.G.	SCACCHI	64	1
16167	CHESSMASTER 3D	RACKOSOFT	SCACCHI	64	2	16240	CHEXDER	GAME ARTS	ARCADE/AVV	64	1
163115	CHICAGO'S 30	ACE	ARCADE	64	1	16440	CHICK FIGHTER	HUDSON	ARCADE	32	1
16516	CHICKEN CHASE	BUG BYTE	ARCADE	64	1	166170	CHILLER	MASTERTRON	ARCADE	32	1
16740	CHIMACHIMA	ROTHTEC	ARCADE	64	1	168143	CHING	PROFEZIE	64	2	
16940	CHOPFLIFTER	BRODERBUND	ARCADE	64	1	17039	CHOPPER	ARCADE	64	1	
17165	CHOPPER 2	BYTEBUSTER	ARCADE/SIM	64	2	17213	CHOROQ	TAITO	ARCADE	32	1
17399	CHUBBY GRISTEL	TOFO	ARCADE	64	1	17437	CHUCKIE EGG	A+F	ARCADE	64	1
1751	CIRCUS CHARLIE	KONAMI	ARCADE	64	1	17631	CITY CONNECTION	JALECO	ARCADE	64	1
1772	CLAPTON II	T&E	SFAZIALE	32	1	178181	CLUEDO	LEISURE GE	GC.SOCIETA	64	1
17913	COASTER RACE	SONY	ARCADE	64	1	180174	COROL 80	MICROSOFT	LINGUAGGIO	64	1
18160	COBRA'S ARC	DINAMIC	ADV/ARCADE	64	1	18296	COCKPIT	SFAZIALE	64	2	
183161	COCKPIT	NIDECOM	SIM. VOLO	256	2	184103	COLISEUM	TOPO SOFT	AVV/ARCADE	64	1
18550	COLOR BALL	HUDSON	ARCADE	32	1	186115	COLOSSUS CHESS 4.0	CDS	SCACCHI	64	1
18728	COLT 36	TOFO	ARCADE	64	1	18817	COMBLOT	INFOGRAMES	CARTE	32	1
18924	COME ON! PICOT	PONYCA	ARCADE	64	1	19041	COMECOCOS	IDEALOGIC	PAC MAN	32	1
19115	COMET TRAIL	ASCII	ARCADE	32	1	1923	COMIC BAKERY	KONAMI	ARCADE	64	1
19357	COMOHOLO	JULIET	ARCADE/AVV	64	1	1940	COMPUTER BILLIARD	UPL/KONAMI	BILIARDO	64	1
195170	CONDORI'		ARCADE	32	1	19650	CONFUSED	BYTEBUSTER	PUZZLE	64	1
19798	CONGO	LIVEWIRE	ARCADE	32	1	1989	CONGO BONGO	SEGA	ARCADE	64	1
199188	CONTO BANCARIO	COMP.HOUSE	CONTO CORR	32	1	200167	CONTRACT BRIDGE	BURKINSHAW	CARTE	32	1

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: