

RIVISTA  
SU CASSETTA

# MSX

# COMPUTER MAGAZINE

N.12

Sped in abb. post. Gr. III L. 9.000



SLOT &  
CARTRIDGE

DATA BASE  
HASHING

L'ISTRUZIONE  
DEF FN

TOEI COURTESY

# 10

PROGRAMMI  
SU CASSETTA

# UNA MAGLIETTA IN REGALO!

a chi si abbona a

**MSX  
COMPUTER  
MAGAZINE**



sei magnifiche cassette di programmi di gioco e di utilità, sempre più belle e ricche!



il prezzo dell'abbonamento (Lire 50 mila) è bloccato per sei numeri e non ti verranno quindi richiesti aumenti (già subito intanto risparmi 4 mila lire)!



avrà subito, direttamente a casa, un'elegante maglietta (realizzata con le riviste consorelle Elettronica 2000 e Load'n'Run) assolutamente gratis!

**ABBONATI  
OGGI  
STESSO**

Basta inviare un vaglia ordinario (quello rosa, da richiedere in un qualunque ufficio postale) di lire 50 mila. Indica esattamente da quale fascicolo desideri l'abbonamento ed i tuoi dati chiari e precisi. Indirizza a MSX Computer Magazine, C.so Vitt. Emanuele, 15 - 20122 Milano.



Nell'immagine di copertina, i soldati meccanici di Zerore, dal bellissimo film Spielvan della Toei Television.



MSX Computer Magazine è edita da Arcadia srl,  
C.so Vitt. Emanuele 15, Milano.  
Tel. 02/706329 (solo giovedì h. 15-18).  
Una copia L. 9.000.  
Fotocomposizione: Composit.  
Stampa: Garzanti,  
Milano. Distribuzione: SO.DI.P. Angelo  
Patuzzi srl, Via Zuretti 25, Milano.  
Registrato Trib. Milano N. 52 del 2/2/85.  
Resp. Sira Rocchi.  
Sped. in abb. post. Gr. III/70.  
MSX is a trademark of MicroSoft Co.  
Manoscritti, disegni, fotografie  
e programmi inviati non si  
restituiscono anche se non pubblicati.

# IN QUESTO NUMERO

★ PAGINE, SLOT, CARTRIDGE

★ DATA BASE: L'HASHING

★ 10 FUNZIONI UTILI

★ IN DIRETTA DAI LETTORI

★ DIECI PROGRAMMI DIECI

- FUGA
- MORRA CINESE
- BATTAGLIA NAVALE
- CORSO D'INGLESE
- GOLF

- MINIERA
- GIOCO DELL'OCA
- EDIPLAN
- SIMULATORE VOLO
- GRAF 64

# MSX TAPE SOFT

I programmi contenuti in questo fascicolo di MSX COMPUTER MAGAZINE sono tutti compatibili con qualsiasi sistema MSX. Ecco per voi ancora 10 bei programmi! Ricordate di collegare la spina del controllo motore alla presa REM del vostro registratore se quest'ultimo la possiede. Assicuratevi che la spina nera sia collegata alla presa EAR del registratore e che la spina rossa sia inserita nella presa MIC. Se il vostro mangiacassette non possiede la presa REM, fate particolare attenzione a quando un programma è stato caricato o deve esserlo, affinché il nastro scorra per il giusto tempo. Appena vedete apparire sul video, dopo un comando di caricamento, la scritta OK, spegnete il regi-



stratore. Nelle istruzioni che seguono troverete, accanto al titolo ed al codice di caricamento di ogni programma, la notazione della memoria minima necessaria per vedere il programma.

Nella cassetta allegata a questo fascicolo troverete, sul lato A:

Fuga, Morra cinese, Battaglia Navale, Corso d'Inglese, Golf.

Sul lato B:

Miniera, Gioco dell'Oca, Ediplan, Simulatore di volo, Graf 64.

# 1

## FUGA

(CLOAD "FUGA" - 32 K)

di M. Zanchi



Caricate la prima parte del gioco con CLOAD, premete F5 oppure scrivete RUN e premete RETURN. Attendete quindi qualche secondo per caricare la parte del gioco in codice macchina. Fuga è un simpaticissimo game composto da 15 schermi diversi attraverso i quali dovrete guidare un ometto alla ricerca di una chiave magica che gli consentirà di cambiare schermo fino alla porta che lo conduce alla libertà. Il gioco comincia sempre, in ogni schermo, con il nostro ometto nell'angolo in alto a sinistra del quadro. In ogni schermo troverete quattro

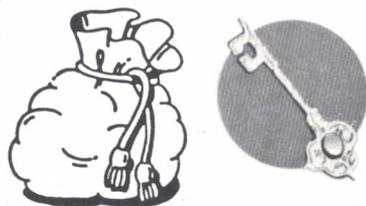


mostriattoli che cercheranno di ostacolarvi: se volete "rallentarli", salite le scale con il tasto di salto. Avete a disposizione 5 vite e 10 mila unità di tempo per schermo: ogni volta che superate un livello, il tempo rimasto vi viene regalato come bonus.

Quando avrete raccolto la chiave nella parte bassa a sinistra dello schermo, sotto la scritta 'PUNTI', apparirà la parola 'CHIAVE': è il segnale che potrete raggiungere la porta ed aprirla! Sul vostro cammino troverete diamanti e tesori di ogni tipo: ogni diamante vale 5 punti, ogni tesoro 10. Prima di giocare potete scegliere se



usare la tastiera o il joystick (porta 1): con T selezionerete la tastiera, con J il joystick. Per iniziare a giocare premete S, oppure la barra dello spazio o se usate la tastiera, o il tasto di fuoco se avete scelto il joystick. Per modificare il gioco interrompetelo prima con i tasti CRT + STOP, battete poi 'SCREEN 1' e premete RETURN. È consigliabile cancellare la prima riga del programma dopo un'interruzione. Se volete aumentare il numero delle vite, modificate il valore della variabile LIV posta nella linea 103. Modificando il valore della variabile VS si varia la velocità di gioco: diminuendo il valore della variabile fino ad un minimo di 0 il gioco sarà più veloce mentre aumentando il valore di VS il gioco sarà più lento. Nella variabile SCR è contenuto il numero dello schermo, che varia da -1 a 13; per modificarla vedete la linea 103. Il numero di unità di tempo è contenuto nella variabile TI posta alla linea 104. Per fare una copia del gioco interrompetelo, premete REC e PLAY sul registratore, digitate 'GOTO 163', quindi premete RETURN.



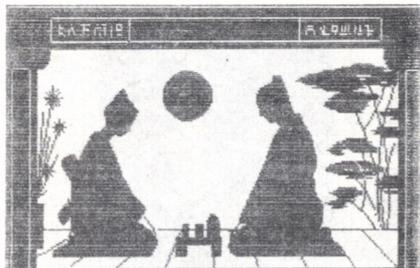
## MORRA CINESE (CLOAD "MORRA" - 16 K)

di G. Barfi



**E**cco per il nostro computer il famosissimo passatempo orientale. Le regole le conosciamo tutti, ma riassumiamole. Ciascun giocatore può scegliere uno fra questi tre oggetti: Carta, Forbici, Sasso.

Fatta la scelta, si pongono a con-



fronto i due oggetti e da questo momento in poi si può vincere, pareggiare o perdere. Vincono: Carta su Sasso; Forbici su Carta; Sasso su Forbici. Si pareggia solo quando i due segni scelti sono uguali. La Carta è rappresentata da una mano aperta, le Forbici da due dita (l'indice ed il medio); il Sasso è rappresentato da un pugno. Per fare la prima scelta, il giocatore 1 dovrà usare i tasti 1, 2 e 3; il giocatore 2 userà invece i tasti 7, 8 e 9. Ecco lo schema dei tasti:

Carta: giocatore 1=tasto 1  
giocatore 2=tasto 7

Forbici: giocatore 1=tasto 2  
giocatore 2=tasto 8  
Sasso: giocatore 1=tasto 3  
giocatore 2=tasto 9

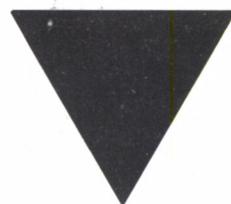
Al centro del video troverete due cinesini su una piattaforma galleggiante sull'acqua. Il cinesino di sinistra rappresenta il giocatore 1, quello di destra il giocatore 2. Chi vince fa un passo avanti, chi perde ne fa uno indietro: vince la sfida quello dei due che riesce a far cadere in acqua l'avversario. Avversario che può essere, a vostra scelta, anche lo stesso computer.

Il programma è commentato da remarks e si carica con CLOAD. Un consiglio: quando si sceglie il segno, meglio nascondere la mano!



## BATTAGLIA NAVALE (CLOAD "NAVALE" - 32 K)

di F. Bonati



**U**n classico gioco da tavolo che farà sempre piacere ritrovare. E quanto è più bello sullo schermo del nostro MSX!

Il primo quadro utile per il gioco chiede le coordinate di posizionamento delle navi, indicate nella parte destra del quadro stesso. Dopo ogni dato che avrete fornito ricordate di premere ENTER. Attenzione: la coordinata che sceglierete corrisponderà sempre alla prua della nave, sia in orizzontale che in verticale. Dopo ogni posizionamento giusto apparirà sempre la nave stilizzata nei riquadri scelti.



Terminati tutti i posizionamenti resteranno solo i due campi di battaglia nei quali sia voi che il calcolatore, a turno, potrete cominciare a sparare. E quando diciamo sparare beh... intendiamo proprio sparare! Il resto lo lasciamo scoprire a voi. Ricordate solo

che le navi possono essere eventualmente affiancate solo sugli spigoli.

Se credete che giocare a questo gioco sia facile, avrete da ricredervi perché il nostro avversario MSX ha una sua logica tutta da invidiare! Buona fortuna dunque.

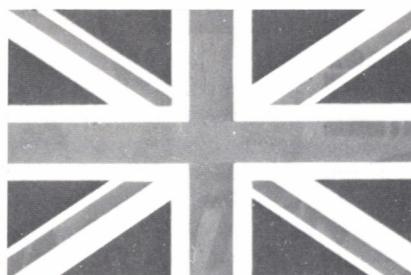


## CORSO DI INGLESE (2<sup>a</sup> parte)

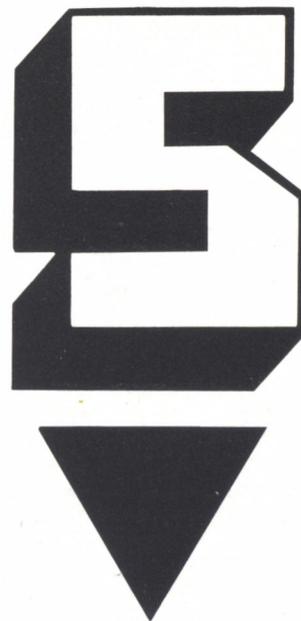
(CLOAD "PARTE2" - 32 K)

di G. Bellomusto

Continua il breve, semplice corso di inglese iniziato sul fascicolo scorso. In questa puntata troverete:



gli aggettivi qualificativi epiteti, il progressivo interrogativo ed affermativo etc. Il programma è sviluppato come la prima parte pubblicata sul fascicolo scorso: esempi, traduzioni ed esercizi. Un'ottima strategia, anche per chi non è proprio portato per le lingue, per imparare l'inglese. Auguri e... good work!



## GOLF 18 HOLE (CLOAD "GOLF18" - 32 K)

di N. Paggin

Un'ottima simulazione del famosissimo gioco chic: questa volta si è data meno importanza alla grafica e più rilevanza alla completezza di gioco e al realismo della partita. Il numero massimo di buche è 18; quello dei giocatori è 8. Alla comparsa della prima buca vedrete una zona di color verde chiaro (il campo vero e proprio) nella quale la pallina può correre liberamente; una zona di color verde scuro (alberi) nella quale la pallina NON può andare. Se, in seguito ad un tiro, la pallina dovesse urtare questa zona, paghereste un colpo di penalità. Vedrete quindi un cerchietto rosso (TEE) sul quale c'è la vostra pallina; un ovale verde con bandierina (GREEN) che indica la buca; una zona azzurra (lago) sulla quale la pallina può transitare ma NON fermarsi, pena un punto tolto; una zona gialla (sabbia) che sarà bene evitare: se la pallina vi si fermasse sopra, consigliamo l'uso della mazza S-W.

Lo scopo di questo gioco è, ovviamente, quello di far arrivare la pallina sul green con il minor numero di colpi. Sotto il disegno della buca vedrete, da sinistra a destra:

- 1 - indicatore della distanza pallina-buca (es: m. 186)
- 2 - numero della buca
- 3 - par della buca (numero di colpi in



cui si dovrebbe "imbucare")

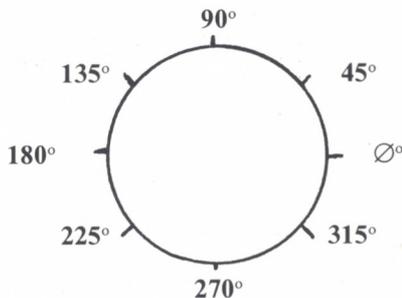
4 - punteggio totale

5 - punteggio parziale della buca

6 - WIND (vento) indica la direzione e la forza (0-8) del vento es: ▲ 7 indica un vento forza 7 che spinge la pallina verso nord/est. Es: ◀ 2 indica un vento di forza 2 che spinge la pallina verso ovest. In base al vento bisogna regolare l'angolo di tiro.

7 - GRADE (pendenza) indica la pendenza all'interno del green (vedi più avanti).

Come effettuare un tiro. 1) scelta della mazza. Vi sono 9 mazze diverse che possono essere utilizzate e ad ognuna corrisponde una certa distanza. In ordine di potenza esse sono: W1 (WOOD1), W3 I1 (Iron 1), I3, I5, I7, I9, S-W (Sand-Wedge) P-W (Putter-Wedge). Per scegliere la mazza, agire sui tasti di sinistra e di destra del cursore per selezionare e premere quindi la barra per confermare la scelta. 2) scelta dell'angolo in cui indirizzare la pallina. Gli angoli disposti come in figura.

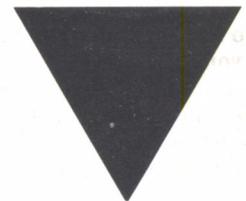


Ad esempio, per tirare tutto a sinistra, selezionate 180° gradi. Per selezionare l'angolo agite sui tasti di sinistra e di destra del cursore per avanzamenti di un grado per volta, e sui tasti superiore ed inferiore del cursore per avanzamenti di 90° gradi per volta. Per confermare l'angolo scelto premete la barra. Ricordate che è fondamentale, nella scelta dell'angolo, tenere conto del vento! 3) forza del tiro. Ogni mazza ha una sua potenza massima ed una sua potenza minima.

Ad esempio, tirando un 17 al massimo della forza, si otterrà circa la forza data da un 15 tirato al minimo. Per far partire il colpo premete la barra: compariranno la scritta POWER ed un rettangolo. Con un altro colpo sulla barra l'indicatore si metterà in moto. Per far partire il tiro premete ancora la barra. Se non darete quest'ultimo colpo alla barra, il tiro partirà al massimo della potenza della mazza scelta. Quando riuscirete ad entrare nel green, la scena cambierà e dovrete imbucare la pallina nella buca nera. Le modalità di tiro sono le stesse; solo la mazza è scelta automaticamente. La pendenza (GRADE) gioca qui lo stesso effetto che il vento giocava nella buca complessiva.

Sulla destra compariranno il nume-

ro di BOGEY, PAR, EAGLE etc. da voi realizzati. Terminologia: EAGLE: buca effettuata 2 colpi sotto il PAR  
BIRDY: buca effettuata 1 colpo sotto il PAR  
PAR: buca effettuata in PAR  
BOGEY: buca effettuata 1 colpo sopra il PAR  
DOUBLE BOGEY: buca effettuata 2 colpi sopra il PAR  
TRIPLE BOGEY: buca effettuata 3 colpi sopra il PAR  
BLEAH: commento personale del programmatore alla vostra eventuale pessima prestazione! Infine: se commettete un errore d'impostazione nel tiro, premete F1 per ripetere l'impostazione. Vinca il migliore!



**MINIERA**  
(CLOAD "M" - 16 K)

di D. Montesor

Alla guida di un modulo comincia la nostra avventura: dobbiamo recuperare dei cristalli di Iridio da un satellite. Il gioco inizia con la navicella ferma su una piattaforma: per decollare potete usare il joystick oppure la tastiera. Decollati che sarete, dovrete scavare la roccia con il laser in modo da poter raggiungere il cristallo (premendo il primo tasto di fire del joystick). Lo stato di carica del laser è indicato in basso a sinistra dello schermo; per ricaricarlo è necessario tornare alla piattaforma di partenza.

Per prendere il cristallo bisogna po-

sizionarlo esattamente fra i ganci dei quali è dotata la vostra navicella poi, raggiunta la verticale sul nastro trasportatore visibile sulla destra, dovrete usare il secondo tasto di fire del joystick, oppure ENTER, per passare alla zona di scavo seguente.

Ogni sei livelli il modulo diviene più veloce ma anche più difficile da manovrare. In basso sullo schermo sono indicati, oltre che l'indicatore di energia, anche i moduli di riserva ed il punteggio. Quest'ultimo dipende sia dal livello raggiunto che dal tempo impiegato per recuperare i cristalli.

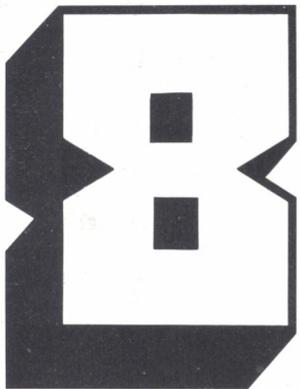




## GIOCO DELL'OCA (CLOAD "OCA" - 32 K)

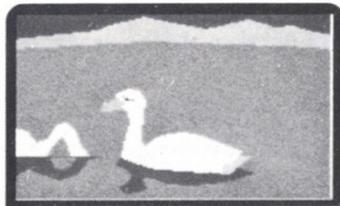
di A. Stringhi

Un altro classico rivisitato per computer, uno dei giochi più famosi nel mondo. Come tutti sanno, si tratta di lanciare dadi ed avanzare a seconda della fortuna in un percorso:



Ci sono in giro tanti programmi per ridefinire i caratteri ma nessuno di essi ha risolto il problema nella sua totalità perché, una volta ridisegnati i caratteri, bisogna in genere andarli a cercare sulla tastiera e poi scrivere linee e linee di PRINT per posizionarli sullo schermo. Ediplan evita tutto questo lavoro: oltre che ridefinire i caratteri con i loro colori, questo programma permette di comporre fino ad 8 schermi con estrema facilità e

vince il primo che arriva alla casella centrale con il numero esatto (altrimenti retrocede secondo il numero eccedente). Possono partecipare al gioco da uno a tre sfidanti, i quali lanceranno il dado (sia da tastiera che con il joystick) e la loro pedina si sposterà di tante caselle quante corrispondenti al numero uscito. Durante il percorso



si incontrano però diverse caselle speciali che provocano spostamenti diversi secondo queste regole:

OCA: lo spostamento viene ripetuto  
AEREO: si avanza direttamente fino al n. 39

DADO: si avanza di dieci caselle

JOLLY: si tira un'altra volta il dado

DADO VUOTO: si retrocede di cinque caselle

GRANCHIO: si retrocede direttamente al n. 13

TESCHIO: si comincia da capo

Il tutto è condito da musicchette appropriate. Al termine della partita viene proclamato il vincitore. Attenzione: il numero 6 non è proprio sempre fortunato...



## EDIPLAN (CLOAD "EDIPLA" - 32 K)

di M. Belardi

di registrarli poi insieme a delle routine in assembler per il loro utilizzo in altri programmi. Gli otto schermi, il set dei caratteri ed i loro colori vengono sistemati sotto la ROM del basic; in RAM restano solo le due routine che servono una a trasferire in VRAM uno degli otto schermi e l'altra il set dei caratteri con i loro colori. In questo modo all'utente restano liberi circa 24 Kbyte. Per poter gestire facilmente gli schermi e le routine, Edi-

plan genera anche un programma in basic che esplica diverse funzioni.

Ediplan lavora sia su nastro che su disco; può copiare una pagina sull'altra; permette tutte le opzioni in tempo reale.

Data la complessità delle parti in assembler il programma, dopo esser stato caricato, chiederà se si vuole fare una copia su nastro o su disco; in pratica si autocopia. Una volta superata questa fase, non sarà possibile ritornarvi e, anche se si esce dal programma, la copiatura di Ediplan non verrà più proposta perché la gestione dell'assembler è dinamica, quindi l'eventuale copia non funzionerebbe. Se infatti si preme il tasto '7' nel menù principale oppure CONTROL-STOP in una fase che non sia quella di input o di output, Ediplan cancella le righe di programma che consentono di realizzarne una copia, proprio per evitare la realizzazione di copie che non funzionerebbero. Per questo motivo quando si esce da Ediplan, si dà il RUN di nuovo e si esce ancora, verrà segnalato un errore: non tenetene conto, è tutto regolare!

Il programma si sviluppa da un menù principale che si dirama su vari sotto menù e dalla parte di edit vera e propria che funziona su due schermi: SCHERMO A indirizzo in VRAM ' 4192

SCHERMO B indirizzo in VRAM ' 5120

Ediplan usa 7 routine in assembler:

usr 0 - lettura carattere sotto il cursore; disegno del pattern del carattere ingrandito e sua lavorazione.

usr 1 - pulizia della memoria-buffer in RAM da C100 a D8FF.

usr 2 - trasferimento di una pagina dalla VRAM alla RAM.

usr 3 - trasferimento di una pagina dalla RAM alla VRAM.

usr 4 - trasferimento profili e caratteri dalla VRAM alla RAM.

usr 5 - trasferimento profili e caratteri dalla RAM alla VRAM.

usr 6 - trasferimento disegno SCHERMO A dalla locazione VRAM 6144 alla locazione 4192.

### SCHERMO A

Presenta l'elenco dei 10 tasti funzione, il loro effetto e su quale schermo operano; il pattern ingrandito del carattere sotto il cursore; un blocco di 32 caratteri.

Usando i tasti di controllo cursore sui caratteri, sul pattern e sulla scelta blocco. Spostandosi sui caratteri si vede il loro profilo ingrandito simul-

taneamente.

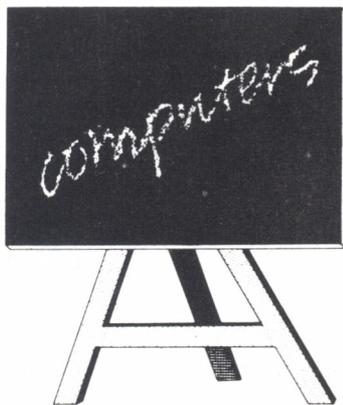
Premendo la barra spaziatrice stando su uno dei 32 caratteri, il cursore salta sul pattern.

Premendo la barra su uno degli 8 numeri si provoca la visualizzazione del gruppo di 32 caratteri corrispondente al numero. Si possono lavorare 248 caratteri, dal codice 8 al codice 255.

Se ci si trova sul pattern si potrà modificare il profilo del carattere premendo la barra spaziatrice. Grazie all'assembler si potrà veder cambiare il profilo del carattere in tempo reale.

## SCHERMO B

Contiene la pagina in lavorazione; si sposta il cursore con il carattere dentro e si posiziona con la barra; si può vuotare il cursore con F5 e poi prendere un carattere già posizionato



in precedenza, sempre premendo la barra.

Premendo F4 compare, sovrapposto alla pagina, il pattern del carattere contenuto nel cursore. In questo modo si può modificare un carattere anche in schermo B, colori compresi. Per far scomparire il pattern bisogna spostare il cursore fuori di esso; si rivedrà così completamente la pagina che si sta lavorando. Se il pattern impedisce la visione della parte di schermo che interessa, si potrà spostare a sinistra e a destra premendo F6 e F7. Praticamente è possibile modificare qualunque carattere anche già posizionato semplicemente portandoci sopra il cursore e premendo F4.

In fase di registrazione verrà chiesto quante pagine si vogliono registrare. Si possono anche caricare pagine e profili lavorati in precedenza.

Quando si passa dallo schermo A allo schermo B il cursore contiene sempre l'ultimo carattere su cui si è

passati prima di cambiare schermo.

## TASTI FUNZIONE ATTIVI SIA IN SCHERMO A CHE IN SCHERMO B

F1 - commuta lo schermo da A a B e viceversa.

F2 - cambia il colore del carattere.

F3 - cambia il colore del fondo del carattere.

## TASTI FUNZIONE ATTIVI IN SCHERMO A

F9 - incrementa il numero della pagina in lavorazione sullo schermo B.

F10 - si torna al menù principale.

## TASTI FUNZIONE ATTIVI IN SCHERMO B

F4 - visualizza, sovrapposto alla pagina, il pattern del carattere contenuto nel cursore per modificarlo, nell'ultima posizione specificata.

F5 - cancella il carattere nel cursore.

F6 - sposta, se presente, il pattern a sinistra di un passo.

F7 - sposta, se presente, il pattern a destra di un passo.

F8 - visualizza sovrapposto un grigliato. Premere ancora per toglierlo.



## MACH 2 (CLOAD "MACH2" - 16 K) di G. Riccobono

Uno splendido simulatore di volo per quelli che non hanno nessuna paura ad impegnarsi. Per giocare inserite le maiuscole.

Diciamo subito quali tasti usare ed a cosa corrispondono.

I tasti ESDX corrispondono alla cloche:

E = per le picchiate

X = per cabrare

S = per virare a sinistra

D = per virare a destra

O = riduce la potenza del motore

P = aumenta la potenza del motore

W = mette il timone a sinistra

R = mette il timone a destra

F = fa uscire i flaps

A = fa rientrare i flaps (senza flaps lo stallo è ad 80 nodi, con i flaps lo stallo

è a 60 nodi)

G = cambia la posizione del carrello (fuori/dentro e dentro/fuori)

PER GIOCARE: nella parte inferiore dello schermo si vedono gli strumenti di bordo.

HDG: è lo "Heading", cioè l'angolo fra il naso dell'aereo ed il nord; HDG=90° l'aereo va ad est; HDG=180° l'aereo va a sud etc.

RDR: posizione del timone, negativo a sinistra e positivo a destra.

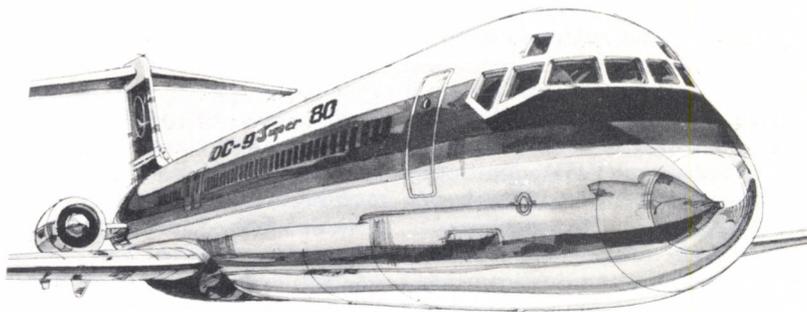
GAS: quantità di benzina rimasta.

SPD: velocità.

PWR: potenza attualmente erogata dal motore.

ALT: quota alla quale ci si trova (altimetro).

ROC/ROD: rapporto di salita (climb)



oppure rapporto di discesa (descent).  
FLP: posizione dei flaps ( $0^\circ$ =flaps completamente ritirati;  $30^\circ$  = flaps completamente estesi).

CAR: su/giù posizione del carrello.

Nella parte superiore dello schermo si vede il terreno che state sorvolando. Verde: zone fino a 100 piedi di quota; marrone: zone fino a 1000 piedi di quota; nero: zone fino a 4000 piedi di quota. Il vostro aereo è rappresentato sulla cartina da un aereo rosso lampeggiante. Le piste degli aeroporti sono rappresentate da due strisce gialle. Quando si arriva in prossimità dell'aeroporto la cartina scompare ed appare la pista di atterraggio.

La velocità di stallo è la minima velocità di sostentamento dell'aereo, al di sotto della quale l'aereo è ingovernabile, entra in vite e precipita. I flaps sono gli alettoni mobili posizionati in genere sul bordo d'uscita dell'aereo e sono concepiti in maniera da abbassare la velocità di stallo. Essi producono però un notevole aumento di resistenza aerodinamica cosicché, alte velocità (nel nostro caso a velocità superiore agli 80 nodi, che è la velocità di stallo senza flaps) è meglio ritirarli, altrimenti il consumo di carburante aumenta di molto e, peggio ancora, la struttura dell'ala è caricata da forze così grandi da rischiare la rottura. Non mettete quindi MAI i flaps quando siete a velocità superiori ai 100 nodi!

Ricapitolando: la velocità minima di sostentamento senza flaps (flaps a  $0^\circ$ ) è di 80 nodi, mentre con i flaps tutti fuori (flaps a  $30^\circ$ ) è di 60 nodi. Con i flaps fuori non si possono superare i 100 nodi. Anche con il carrello fuori non si possono superare velocità di 100 nodi, altrimenti le forze di resistenza aerodinamica lo spaccano. Ricordate quindi sempre di ritirare il carrello dopo il decollo e non estraetelo mai prima dell'atterraggio. Non superate mai comunque i 200 nodi di velocità. PER DECOLLARE: aumentate la potenza con il tasto P finché la velocità supera gli 80 nodi, dopo di che "tirate la cloche" schiacciando il tasto X. A questo punto l'aereo si stacca da terra e comincia a salire. In fase di salita tenete delle potenze elevate (200 cv). PER ATTERRARE: cercate di arrivare ad inizio pista con rapporti di discesa (ROD) piuttosto bassi, non più di 300. Cercate anche di arrivare lentamente (70 nodi) e con i flaps tutti fuori ( $30^\circ$ ). Appena l'altimetro segna 0 riducete drasticamente la potenza del motore schiacciando ripetutamente il tasto O.



## GRAF 64 (CLOAD "GRAF64" - 16 K) di D. Montesor

Questa utility permette di visualizzare 64 colonne sullo schermo grafico degli MSX, con caratteri formati da una matrice di 4x8 pixel. Si possono usare caratteri di codice compreso tra 32 e 127, altrimenti vengono convertiti nel carattere spazio. Scrivete, alla locazione 61557, la coordinata X (da 0 a 63) ed alla loca-

zione 61558 la coordinata Y (da 0 a 23); definite la routine in L/M a partire dalla locazione 61249; scegliete eventualmente il colore del carattere e dello sfondo con la normale istruzione color del basic; infine eseguite la seguente istruzione: A\$=USR(A\$) dove AS è la stringa da scrivere sullo schermo. Nel caso durante la stampa, le coordinate escano dallo schermo durante la visualizzazione, esse vengono azzerate. Appena visualizzata una stringa, i valori di X ed Y sono quelli successivi all'ultimo carattere visualizzato. Le linee da 196 a 236 sono un semplice demo.

# MSX DATABASE

## PER UN BUON ALGORITMO DI RICERCA DATI. LA TECNICA SEQUENZIALE, L'HASHING.

a cura della Redazione

L'archivio elettronico è una delle applicazioni più diffuse per il computer e tutti noi, per lavoro o per hobby, vorremmo realizzare sul nostro MSX un archivio. Il problema principale da affrontare, a prescindere dalla memoria utente e di massa disponibile, è l'algoritmo di ricerca dei dati. Se l'inserzione o la cancellazione di un dato in un archivio non crea infatti problemi, ne possono creare invece il fare la ricerca e quindi la consultazione delle informazioni. Può ad esempio capitare di dover trovare un nominativo in un archivio di 1000 persone e, già in questo caso, la velocità di elaborazione ha il suo bel peso. Per risolvere il problema, cioè per evitare noiose attese di ricerca, bisogna gestire i dati in modo intelligente e quindi applicare un buon algoritmo.

Vedremo qui ora tre differenti metodi di ricerca: dal più banale, quello sequenziale, a quello raffinato, l'hashing.

Partiamo dunque dal presupposto che il nostro archivio sia su vettore, quindi in memoria utente.

### LA RICERCA SEQUENZIALE

La logica più spontanea per trovare un nome in una lista è quella di far scorrere tutti gli elementi del vettore dall'inizio fino a trovare il nome cercato oppure, in caso di esito negativo, fino alla fine della lista. Questa è la ricerca sequenziale. La sua realizzazione è molto semplice ma il suo rendimento è assai scarso perché, per trovare una informazione, bisogna far

scorrere tutte quelle precedenti a quest'ultima. Supponiamo di avere il vettore V\$( ), dimensionato a 100 locazioni, contenente l'archivio dei nomi e di voler trovare il nome contenuto nella variabile S\$. L'algoritmo di ricerca sequenziale è molto semplice, come si può notare dal listato n. 1. Un semplice ciclo iterativo ed una istruzione di confronto consentono di realizzare il tutto anche se, come già detto prima, la soluzione non è delle migliori.

Se come speriamo tutto è chiaro, possiamo passare ad esaminare un'altra tecnica di ricerca, quella binaria.

"hot-dog" e, dato che quest'ultimo si trova oltre quello che noi cerchiamo, riapriamo il libro a metà, sempre tra la prima pagina e quella in cui siamo posizionati. La pagina selezionata inizia con la parola "gabbia", quindi sfogliamo le pagine in avanti, precisamente fino ad arrivare a metà tra la pagina attuale e quella contenente la parola "hot-dog". Ed ecco che abbiamo trovato infine la pagina contenente il vocabolo "galleggiare".

Non è che un vocabolario si consulti sempre nel modo appena descritto... ma tutto questo serve per spiegare

```
100 REM ----- RICERCA SEQUENZIALE -----
110 N=1
120 IF V$(N)=S$ THEN PRINT S$;" TROVATO ALLA POSIZIONE "; N:END
130 N=N+1
140 IF V$(N)="" OR N>100 THEN PRINT S$;" NON E' NELL'ARCHIVIO": END
150 GOTO 120
```

LISTATO N.1 - ALGORITMO DI RICERCA SEQUENZIALE

### LA RICERCA BINARIA

Questa tecnica di ricerca prende il nome di "binaria" perché, per trovare il nome, divide per due la lista. Per capire come funziona facciamo un esempio.

Quando dobbiamo consultare un termine in un vocabolario apriamo innanzitutto il libro e poi cerchiamo quel termine che ci interessa sfogliando in avanti o indietro il volume a seconda di dove siamo posizionati. Se vogliamo cercare il termine "galleggiare" e ci troviamo con il vocabolario aperto sulla parola "palla", riapriamo il libro a metà circa tra la prima pagina e quella attuale. Ci troviamo ora posizionati sul vocabolo

come avviene la ricerca binaria. La ricerca inizia sempre confrontando il dato centrale della tabella e, se non si tratta dell'informazione desiderata, allora l'archivio sul quale eseguire la ricerca si riduce a metà tabella, precisamente la parte "superiore" o "inferiore", a seconda che il nome confrontato non sia inferiore o superiore all'informazione desiderata. Successivamente si ripetono le operazioni di prima, ora con una tabella di lunghezza pari alla metà di quella del precedente confronto. Il dato da confrontare è sempre quello che si trova a metà tabella. Tutto questo viene ripetuto finché non si trova l'informazione, oppure fino a quando la tabella si riduce ad essere composta da un solo

```

100 REM ----- ARCHIVIO BINARIO -----
110 PRINT "PREMI <R> PER RICERCA"
120 PRINT "PREMI <I> PER INSERIMENTO"
130 PRINT "PREMI <C> PER CANCELLARE"
140 INPUT SCELTA
150 ON SCELTA GOSUB 200,300,400
160 GOTO 110
200 REM ----- RICERCA BINARIA -----
210 INPUT "NOME DA CERCARE ";S$
220 REM ----- ALGORITMO DI RICERCA -----
230 I=1:S=100:C=50
240 IF S$=V$(C) THEN PRINT S$;" TROVATO ALLA POSIZIONE ";C:RETURN
250 IF S$>V$(C) THEN I=C+1 ELSE S=C-1
260 IF S=I AND S$<>V$(S) THEN PRINT S$;" NON E' STATO TROVATO":RETURN
270 C=INT((S+I)/2)
280 GOTO 240
300 REM ----- INSERIMENTO NOME -----
310 INPUT "NOME DA INSERIRE ";S$
320 GOSUB 230
330 IF S$<I THEN PRINT "NOME GIA' PRESENTE":RETURN
340 FOR N=99 TO S STEP-1
350 V$(N+1)=V$(N)
360 NEXT N
370 V$(S)=S$
380 RETURN
400 REM ----- CANCELLAZIONE NOME -----
410 INPUT "NOME DA CANCELLARE ";S$
420 GOSUB 230
430 IF S=I THEN PRINT "NOME GIA' CANCELLATO":RETURN
440 FOR N=C TO 99
450 V$(N)=V$(N+1)
460 NEXT N
470 RETURN

```

LISTATO N.2 - GESTIONE ARCHIVIO BINARIO

elemento; in tal caso, se l'unica informazione rimasta non è quella desiderata, la ricerca ha avuto esito negativo.

Perché tutto il meccanismo di ri-

cerca binaria possa funzionare, la tabella dei nomi deve essere ordinata alfabeticamente e quindi l'aggiunta di nomi nuovi deve essere fatta in modo opportuno e non semplicemente ag-

```

100 REM ----- ARCHIVIO HASH -----
110 PRINT "PREMI <R> PER RICERCARE"
120 PRINT "PREMI <I> PER INSERIRE"
130 PRINT "PREMI <C> PER CANCELLARE"
140 INPUT SC
150 ON SC GOSUB 200,300,400
160 GOTO 110
200 REM ----- RICERCA HASH -----
210 INPUT "INSERISCI IL NOME DA CERCARE ";S$
220 GOSUB 500 'CALCOLO DEL VALORE HASH DI S$
230 IF V$(H)="" THEN PRINT S$;" NON E' STATO TROVATO":RETURN
240 IF V$(H)=S$ THEN PRINT S$;" E' STATO TROVATO ALLA POSIZIONE ";
H:RETURN
250 H=H-1
260 IF H=0 THEN H=100
270 GOTO 230
300 REM ----- INSERIMENTO NOME -----
310 INPUT "INSERISCI IL NUOVO NOME ";S$
320 IF MN=99 THEN PRINT "LISTA PIENA":RETURN 'MN CONTIENE IL NUMERO
DEI NOMI PRESENTI IN ARCHIVIO
330 MN=MN+1 'INCREMENTA IL NUMERO DEI NOMI PRESENTI IN ARCHIVIO
340 GOSUB 500
350 IF V$(H)="" OR V$(H)="*" THEN V$(H)=S$:RETURN 'NOME INSERITO
360 H=H-1
370 IF H=0 THEN H=100
380 GOTO 350
400 REM ----- CANCELLAZIONE NOME -----
410 GOSUB 200
420 IF V$(H)<>S$ THEN RETURN
430 V$(H)="*":RETURN 'IL SIMBOLO "*" STA AD INDICARE LOCAZIONE
CANCELLATA
500 REM ----- CALCOLO DEL VALORE HASH -----
510 H=0
520 FOR N=1 TO LEN(S$)
530 H=H+ASC(MID$(S$,N,1))
540 NEXT N
550 RETURN

```

LISTATO N.3 - GESTIONE ARCHIVIO AD ACCESSO DIRETTO (METODO HASHING)

giungendo in coda il nuovo dato così come poteva essere fatto nel caso di una tabella a ricerca sequenziale. Per aggiungere un nuovo nome alla lista bisogna utilizzare la ricerca binaria la quale, dopo il mancato ritrovamento del nome, si posizionerà nel punto giusto nel quale inserire la nuova informazione. A partire da quella posizione tutti i dati verranno spostati di un posto in avanti in modo da lasciare lo spazio per il nuovo nome.

La ricerca binaria è decisamente migliore di quella sequenziale; i confronti che servono a trovare l'informazione sono molto meno rispetto a quelli necessari in una ricerca sequenziale e, per la precisione, sono al massimo pari alla parte intera del logaritmo in base due di n più uno, dove n è il numero di elementi presenti nella lista.

Nel listato 2 è riportato il programma di ricerca e di inserzione di un dato in una tabella "binaria".

## RICERCA AD ACCESSO DIRETTO: IL METODO HASHING

Fino ad ora abbiamo visto che per ottimizzare un algoritmo di ricerca bisogna che quest'ultimo riesca, con il minimo possibile dei confronti, a trovare l'informazione desiderata.

L'algoritmo ideale sarebbe quindi quello che, date l'informazione da trovare ed una formula appropriata, estrae direttamente il record contenente il nome da trovare. La ricerca ad accesso diretto (il nome stesso lo dice) accede direttamente all'informazione dell'archivio che noi desideriamo. La formula che consente di ricavare la posizione nell'archivio del nome da cercare si basa sul metodo hashing che associa ad ogni nome (in generale ad ogni stringa) un valore numerico dato dalla somma del valore ASCII di ogni carattere. Per esempio per il nome "DINO" il valore hash è:  $DINO = ASC("D") + ASC("I") + ASC("N") + ASC("O") = 68 + 73 + 78 + 79 = 298$

Avendo però a che fare con un archivio da 100 elementi, la locazione esatta all'interno dell'archivio è data dal resto del valore hash diviso 100, cioè:

$$POS = (298/100 - INT(298/100)) * 100 = 98$$

Quindi il nome "DINO" si troverà nell'archivio alla posizione n. 98. In altre parole, se l'archivio è nel vettore V\$( ) il nome "DINO" si troverà in V\$(98).

Nasce un problema quando, a nome diverso, corrisponde la posizione di un altro; in tal caso si parlerà di collisione.

Per esempio il nome ADA ha un valore hash identico a quello di DINO:  
ADA=ASC("A")+ASC("D")+ASC("A")=  
65 + 68 + 65 = 198

Per evitare tutto ciò il nome ADA sarà inserito nella posizione 97, se libera, altrimenti nella 96 e così via. Nel caso si raggiungesse la posizione 1 dell'archivio senza trovare uno spazio vuoto nel quale inserire il nome, la scansione della tabella: riprenderebbe dalla posizione 100. Per evitare un inconveniente del genere è consigliabile non riempire più dell'80% di una tabella destinata ad una organizzazione hashing.

La ricerca esegue lo stesso processo dell'inserzione: si calcola il valore hash del nome da trovare e successivamente si accede alla locazione N calcolata. Se non si trova l'informazione desiderata, si controlla la locazione N-1 e così via fino a trovare l'informazione; altrimenti, se l'indice punta ad una casella vuota, vuol dire che quel nome non esiste nella tabella. Una lista lunga N locazioni deve contenere al massimo N-1 nomi altrimenti, in caso di ricerca negativa, l'algoritmo si perde in un ciclo senza fine.

L'eliminazione di un nome non deve soltanto lasciare vuota la casella occupata da quel nome ma deve anche marcarla per evitare inconvenienti dovuti alla collisione di dati.

Se per esempio cancellassimo il nome DINO semplicemente lasciando vuota la casella 98, ricercando in seguito il nome ADA troveremmo V\$(98) vuota concludendo erroneamente che ADA non esiste nella lista.

#### IN CONCLUSIONE

Il metodo di ricerca sequenziale rimane senz'altro il più intuitivo.

La ricerca binaria invece apporta già notevoli vantaggi rispetto alla sequenziale ed è tra le più usate in applicazioni per le quali si richiede l'uso di un archivio. Il metodo hashing infine è quello che assicura la massima velocità di ricerca: anche su tabelle di notevoli dimensioni (2000 e più nomi) infatti la ricerca di un'informazione si ottiene statisticamente in soli 3 confronti. Un risultato del tutto eccezionale, anche se per applicare il metodo hashing bisogna prevedere alcune condizioni spiacevoli quali la collisione e la saturazione dell'archivio.

**PUOI RICHIEDERE  
con vaglia  
IN REDAZIONE**

by Computer Magazine  
N. 1 - GIU/LUG 1986

L. 12.000  
Sped. in abb. post. Gr. III/70

# MSX

# HIT PARADE

RIVISTA DI PROGRAMMI SU CASSETTA PER COMPUTER STANDARD MSX



COLLEZIONE  
DI PROGRAMMI  
GIÀ PUBBLICATI

2  
CASSETTE  
GIOCHI, UTILITY

**UNA BUONA OCCASIONE  
PER IL SOFTWARE  
CHE ABBIAMO GIÀ PUBBLICATO  
DUE CASSETTE INSIEME  
A PREZZO SPECIALE!**

# PAGINE, SLOT E CARTRIDGE

## CHI HA DETTO CHE LA MEMORIA DEL NOSTRO COMPUTER È SCARSA?! ECCO COME RAGGIUNGERE UN MEGA!

di E. DASSI

Se il componente più importante di un computer è il microprocessore, la memoria è, in ordine di rilevanza, l'elemento indispensabile per poter memorizzare le informazioni, memoria spesso insufficiente per le nostre esigenze.

Vogliamo qui descrivere allora l'intelligente struttura della memoria degli MSX, che permette di raggiungere la quantità più che rispettabile di ben un mega byte!

La CPU degli MSX, il microprocessore Z80, ha un bus indirizzi da 16 bit il quale può indirizzare fino a 64K byte di memoria in linea. Per aumen-

tare tale quantità l'MSX "vede" i 64 Byte di memoria divisi in 4 pagine da 16K byte ciascuna. In tal modo l'MSX può espandere la sua memoria selezionando differenti slot per ogni pagina. Uno slot è formato da uno spazio di memoria pari a 64K, ed è un banco di memoria separata.

Fisicamente, uno slot può essere un cartridge o una espansione di memoria come la ROM del BASIC MSX e la sua RAM di sistema.

Per selezionare lo slot che deve essere usato per una certa pagina, bisogna modificare il registro di selezione slot (ne parleremo più avanti). Soli-

tamente un sistema base MSX ha la memoria configurata come nella tabella n. 1.

Tutti i computer MSX hanno almeno una porta per cartridge alla quale collegare una serie di dispositivi. Ecco una lista dei possibili dispositivi collegabili tramite lo slot-cartridge:

- espansione RAM. Solitamente usata per espandere i sistemi 16K a 32K.
- ROM su cartridge contenente giochi.
- ROM di estensione linguaggio BASIC. Le estensioni sono accessibili tramite l'istruzione di chiamata in BASIC CALL.

PAGINA	SLOT 0	SLOT 1,2 o 3
3	16K RAM per i sistemi MSX di 16K RAM	
2	16K RAM per i sistemi MSX di 32K RAM	cartridge ROM giochi 16K RAM espansione
1	ROM BASIC MSX	espansioni ROM BASIC altri linguaggi
0	ROM BASIC MSX (BIOS)	

Tabella n. 1. Configurazione base di un sistema MSX.

	SLOT 0	SLOT 1	SLOT 2	SLOT 3
&HFFFF	+-----+ ! PAGINA 3 !			
&HC000	+-----+	+-----+	+-----+	+-----+
&HBFFF	+-----+ ! PAGINA 2 !			
&HB000	+-----+	+-----+	+-----+	+-----+
&H7FFF	+-----+ ! PAGINA 1 !			
&H4000	+-----+	+-----+	+-----+	+-----+
&H3FFF	+-----+ ! PAGINA 0 !			
&H0000	+-----+	+-----+	+-----+	+-----+

Figura n. 1. Configurazione degli slot di base.

	SLOT 0 ESPANSO				SLOT 1 ESPANSO				SLOT 2 ESPANSO				SLOT 3 ESPANSO			
	←-----→				←-----→				←-----→				←-----→			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
PAGINA 3	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++
	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!
	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++
PAGINA 2	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++
	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!
	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++
PAGINA 1	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++
	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!
	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++
PAGINA 0	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++
	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!	!!!
	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++	+++

Figura n. 2. Configurazione del sistema MSX con tutti gli slot.

- Dispositivi di I/O quali floppy disk, light pen ecc.

Ogni dispositivo sopra elencato può essere inserito in qualsiasi slot. Infatti l'MSX ha un meccanismo di selezione slot tale da poter riconoscere quale cartridge deve accedere al sistema.

L'MSX ha 4 slot base; lo slot numero 0 è quello nel quale si trova la ROM del BASIC MSX ed è chiamato slot di sistema. Ognuno in questi 4 slot può essere espanso a sua volta con altri 4 slot, perciò il numero totale di slot installabili è 16. Questi 16 slot possono essere costituiti tutti da 64K di RAM, dando così ben un mega byte di memoria. Questa è la massima quantità di RAM supportata dal sistema MSX, anche se non è possibile accedere a tutto un mega byte di me-

moria da BASIC. Per poter disporre contemporaneamente di più di 64K di memoria bisogna utilizzare l'MSX-DOS, oppure un programma in linguaggio macchina. La figura n. 1 mostra la configurazione base degli slot; la figura n. 2 la versione espansa degli slot.

#### COME SELEZIONARE ED ABILITARE UNO SLOT

L'MSX può avere diversi slot ed espansioni di memoria che, se non sono controllati, possono interferire tra di loro perchè la CPU, come detto all'inizio, gestisce 64K byte alla volta. Per determinare quale slot deve essere usato e per quale pagina, l'MSX ha uno speciale meccanismo di selezione. La CPU può accedere direttamente a

64K o a 4 pagine da 16K. Queste 4 pagine possono essere indistintamente in differenti slot. Le pagine assegnate agli slot sono selezionate usando la porta A dell'8255. Per far ciò via software bisogna utilizzare l'indirizzo di I/O &HA8 il quale è l'uscita della porta A dell'8255. Il valore è ad 8 bit ed ha il seguente significato:

bit 0-1 numero slot per la pagina 0  
bit 2-3 numero slot per la pagina 1  
bit 4-5 numero slot per la pagina 2  
bit 6-7 numero slot per la pagina 3

Volendo quindi usare lo slot 0 per la pagina 0, 1 e 3 e lo slot 1 per la pagina 2, la configurazione del valore della porta A sarà (in binario) 00010000 cioè 16 in decimale.

Per impostare tale valore si può usare la funzione Basic OUT (&HA8), 16, ma è meglio utilizzare la

routine in linguaggio macchina del BIOS chiamata ENASLT (indirizzo di start &H0024).

L'MSX ha la possibilità di avere fino a 4 slot base, da 0 a 3. Non tutti questi sono presenti in ogni macchina, ma ogni slot base può essere espanso usando un box d'espansione. Perciò il numero massimo di slot è 16. Ogni slot può contenere 64K di RAM, così la memoria massima per un MSX è di 1M bytes.

Per selezionare uno slot bisogna per prima cosa selezionare lo slot base, nel quale lo slot espanso è alloggiato, usando la porta A dell'8255.

A questo punto il registro di selezione degli slot espansi è alla locazione &HFFFF del box d'espansione. Questo determina se la pagina selezionata in quello slot è usata o no. Per verificare se uno slot ha collegati altri slot bisogna scrivere alla locazione &HFFFF con l'istruzione POKE &HFFFF, &HOF. Se, facendo successivamente una lettura alla locazione &HFFFF con PEEK (&HFFFF), si ottiene quanto scritto precedentemente, vuol dire che c'è collegato un box d'espansione con relativi slot aggiuntivi.

Quando il computer viene acceso, il sistema operativo ricerca tutti gli slot per selezionare la memoria RAM di sistema. La procedura di ricerca si sviluppa secondo i seguenti punti:

1. Il sistema ricerca prima la memoria RAM disponibile in pagina 2 dall'indirizzo &H8000 all'indirizzo &HBFFF, poi abilita lo slot con la maggiore quantità di RAM in pagina 2. Se c'è più di uno slot con la stessa quantità in memoria RAM, allora viene selezionato lo slot avente il numero di slot più piccolo.

2. La stessa operazione viene fatta per la pagina 3, dall'indirizzo &HC000 all'indirizzo &HFFFF. Anche in questo caso i criteri di selezione slot sono identici a quelli del punto 1.

3. Per ultimo viene fatto un test sulla continuità della RAM dall'indirizzo &H8000 all'indirizzo &HFFFF e, successivamente, viene settata la variabile BOTTOM (indirizzo &HFC48) all'indirizzo d'inizio della RAM disponibile.

Dopo aver selezionato la RAM di sistema, l'MSX provvede alla ricerca dei cartridge della ROM dall'indirizzo &H4000 all'indirizzo &HBFFF per la pagina 1 e 2. Viene quindi creata una tabella all'inizio di ogni pagina, dallo slot 0 allo slot 3, ed anche agli slot espansi quando vi siano. La tabella creata è mostrata in figura 3.

## INIZIO PAGINA

+&H00	+-----+	! ID 'AB' !	+-----+
+&H02	+-----+	! INIT !	+-----+
+&H04	+-----+	! STATEMENT!	+-----+
+&H06	+-----+	! DEVICE !	+-----+
+&H08	+-----+	! TEXT !	+-----+
+&H0A	+-----+	! RISERVATA!	+-----+
+&H10	+-----+		+-----+

Figura n. 3. Indirizzi relativi alle routine del cartridge assegnato alla pagina di memoria.

Come si può osservare i primi due byte, "ID", contengono il codice "AB" indicante che c'è il cartridge della ROM. Il terzo ed il quarto byte, indicati con "INIT", contengono l'indirizzo della routine di inizializzazione di quel cartridge. I seguenti due byte, "STATEMENT", contengono l'indirizzo nel quale vengono gestiti eventuali statement aggiuntivi. "DEVICE" è l'indirizzo di start della routine che gestisce i dispositivi aggiunti; "TEXT" e, infine, l'indirizzo di start del programma Basic contenuto in quel cartridge. Tutte queste locazioni di memoria possono contenere anche l'indirizzo &H0000, in tal caso vuol dire che non esiste la relativa routine.

Il BASIC MSX esegue i seguenti processi come procedura di ricerca cartridge:

- Testa l'area ID ed estrae quale tipo di routine ha.
- Esegue, se c'è, la routine INIT.
- Esegue, se c'è, il programma Basic presente in quel cartridge.

Le routine STATEMENT e DEVICE non vengono eseguite direttamente dal sistema ma solo quando l'utente ne vuole fare uso; per esempio, volendo usare un nuovo statement, lo si chiamerà con l'istruzione CALL — nome statement —.

In conclusione, riportiamo le locazioni di alcune variabili di sistema riguardanti gli slot.

Variabili di stato di ogni slot: EXPTBL (indirizzo &HFCC1) indica quale slot è espanso. La variabile è lunga 4 byte:

- &HFCC1 per lo slot 0
- &HFCC2 per lo slot 1
- &HFCC3 per lo slot 2
- &HFCC4 per lo slot 3

Se il contenuto di queste locazioni è &H80 vuol dire che lo slot è espanso, se invece è &H00 vuol dire che non lo è. SLTTBL (indirizzo &HFCC5) indica quale valore è in uscita nel registro di selezione degli slot espansi. Tale valore ha significato solo se la corrispondente variabile EXPTBL segnala che quello slot è espanso. SLTTBL è lunga 4 byte:

- &HFCC5 per lo slot 0
- &HFCC6 per lo slot 1
- &HFCC7 per lo slot 2
- &HFCC8 per lo slot 3

Variabili di stato di ogni pagina: SLTATR (indirizzo &HFCC9) dice che cosa c'è in ogni possibile pagina. La variabile è lunga 64 byte, tante quante sono le pagine collegabili in un MSX:

- &HFCC9 per lo slot base 0 slot espanso 0 e pagina 0
- &HFCCA per lo slot base 0 slot espanso 0 e pagina 1
- &HFCCB per lo slot base 0 slot espanso 0 e pagina 2
- ...
- ...
- ...
- &HFDD6 per lo slot base 3 slot espanso 3 e pagina 1
- &HFDD7 per lo slot base 3 slot espanso 3 e pagina 2
- &HFDD8 per lo slot base 3 slot espanso 3 e pagina 3

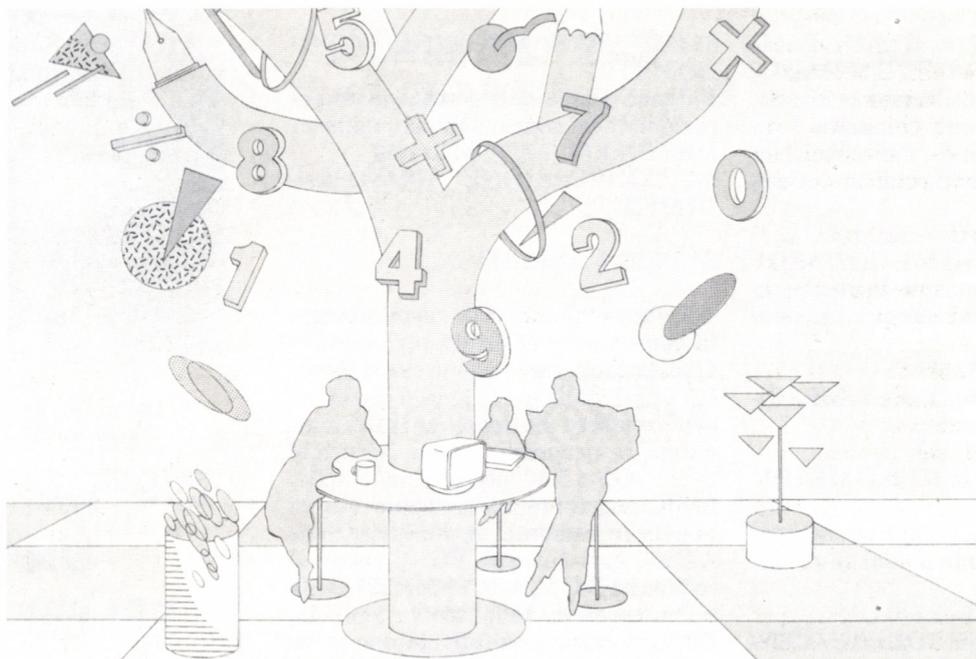
Il valore contenuto in ognuna di queste locazioni va interpretato così: il bit 5 indica (se a 1) che vi è in quella pagina un'espansione di statement; il bit 6 (se a 1) che vi è una routine di gestione di un dispositivo ed il bit 7 indica che c'è nella pagina un programma Basic.

SLTWRT (indirizzo &HFDD9) è una variabile lunga 128 byte e riserva 2 byte per ogni pagina come area di lavoro. Inizia dall'indirizzo &HFDD9-&HFDDA per lo slot di base 0 con lo slot espanso 0 di pagina 0 (prima pagina) e termina all'indirizzo &HFD87-&HFD88 per lo slot di base 3 con lo slot espanso 3 di pagina 3 (ultima pagina)

# 10 FUNCTION 10

**L'UTILIZZAZIONE PRATICA DEL NOSTRO COMPUTER. DESCRIZIONE DI DIVERSE FUNCTION (ISTRUZIONE DEF FN) UTILI PER CHI STUDIA O CHI LAVORA.**

di G. RICCOBONO



**R**iprendiamo su questo numero, come preannunciato, l'articolo dedicato alle function. Tutto quello che riguarda l'istruzione DEF FN (il formato, il tipo di utilizzo, la sua utilità etc.) è già stato trattato sul numero dieci, quindi questa volta presenteremo soltanto una serie di altre dieci function utili.

Come completamento alle dieci function "matematiche" trattate la volta scorsa, vedremo quest'oggi dieci conversioni divenute ormai, a causa della sempre maggiore "inglesizzazione" a cui siamo soggetti, di uso comune nella vita quotidiana.

L'idea dell'importanza delle conversioni tra diversi sistemi di misure,

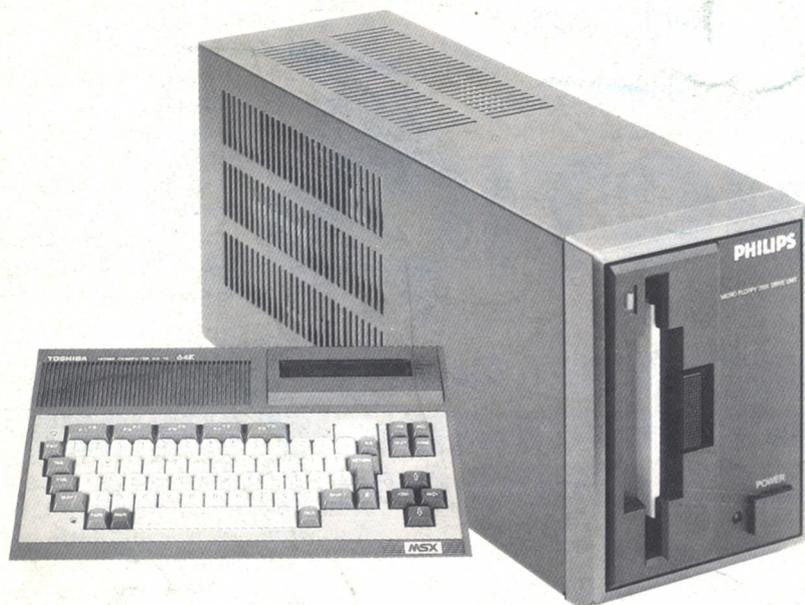
mi è sorta pressappoco un anno fa quando, scrivendo un programma di ingegneria sul calcolo strutturale degli sforzi in un'ala di aeroplano, mi è stata data come specifica fondamentale, da colui che mi aveva commissionato il programma, la possibilità di eseguire tutti i calcoli sia in chilogrammi e metri che in libbre e pollici. Chi di voi si trova a dover usare il calcolatore per scopi tecnici e scientifici, si troverà prima o poi probabilmente di fronte a questa stessa esigenza, cioè quella di poter usare uno stesso programma utilizzando ogni tanto una unità di misura e ogni tanto un'altra, soprattutto in tempi come quelli odierni, in cui la maggior parte

della bibliografia è in lingua inglese.

Per fare un paragone un po' azzardato, dare la possibilità, all'utente di un programma, di ottenere i risultati in due sistemi di misura differenti, è come insegnare al nostro calcolatore a parlare in due lingue, il nostro solito italiano e l'inglese.

Vediamo dunque come ottenere queste famose conversioni utilizzando l'istruzione DEF FN.

1) CONVERSIONE GRADI CENTIGRADI/GRADI FAHRENHEIT  
Chiamata C la temperatura in gradi centigradi, la temperatura in gradi Fahrenheit è ottenibile attraverso la function così definita:  
DEF FN GF (C)=1.8\*C+32



## 2) CONVERSIONE GRADI FAHRENHEIT/GRADI CENTIGRADI

È ovviamente la conversione inversa di quella precedente; chiamata F la temperatura in gradi Fahrenheit, la temperatura in centigradi si ottiene con:

$$\text{DEF FN GC (F)} = (F - 32) / 1.8$$

## 3) CONVERSIONE METRI/PIEDI

Chiamata M la misura in metri, la corrispettiva in piedi si può ottenere con:

$$\text{DEF FN P(M)} = .32 * M$$

## 4) CONVERSIONE PIEDI/METRI

Si ottiene ovviamente con:

$$\text{DEF FN M(P)} = M / .32$$

## 5) CONVERSIONE CENTIMETRI/POLLICI

Chiamando CM la misura in centimetri la corrispettiva in pollici si ottiene con:

$$\text{DEF FN PL (CM)} = 2.54 * CM$$

## 6) CONVERSIONE POLLICI/CENTIMETRI

Anche questa è immediatamente ricavabile dalla precedente come:

$$\text{DEF FN CM(PL)} = PL / 2.54$$

## 7) CONVERSIONE CHILOMETRI/MIGLIA

### PER ESSERE PRECISI

Si stia bene attenti al fatto che la conversione qui riportata è relativa alle miglia terrestri e non a quelle nautiche, se volete ottenere quelle nautiche basta che sostituiate il fattore 1610 con il fattore 1852. Indicando con KM la misura in chilometri, la corrispondente misura in miglia terrestri è ottenibile con:

$$\text{DEF FN MT(KM)} = 1.61 * KM$$

## 8) CONVERSIONE MIGLIA/CHILOMETRI

Valgono anche qui le stesse considerazioni di cui sopra, abbiamo dunque:

$$\text{DEF FN KM (MT)} = MT / 1.61$$

## 9) CONVERSIONE GRADI/RADIANTI

### SUPERMATEMATICA

Concludiamo questa carrellata con la conversione di utilità più comune. Questa è una conversione che si discosta un po' dalle precedenti a causa del suo carattere più matematico. Ricordiamo in proposito che le istruzioni SIN, COS e TAN presenti nel BASIC MSX, accettano come argomento un angolo in radianti, anche l'istruzione ATAN, così come le FUNCTION di ARCOSENO e ARCOSENO da noi presentate nello scorso articolo, fornisce come risultato un angolo in radianti e sarà quindi spesso necessaria una conversione radianti/gradi e viceversa. È dunque decisamente conveniente, a tal fine, inserire nei nostri programmi due FUNCTION che ci permettano una volta per tutte di ottenere le conversazioni citate. La prima può essere così ottenuta:

$$\text{DEF FN RD(GR)} = GR / .0174533$$

## 10) CONVERSIONE RADIANTI/GRADI

È ovviamente la conversione inversa di quella precedente e valgono dunque per essa le stesse considerazioni del punto precedente. Questa conversione può essere comunque ottenuta con:

$$\text{DEF FN GR(RD)} = RD * .0174533$$

# RAGAZZI ATTENZIONE

Se hai un computer  
e un modem  
puoi chiamare  
**02/706857**  
ed entrare  
in un fantastico club!!!



## A VOSTRA DISPOSIZIONE UNA SPLENDIDA BANCA DATI

Per informazioni  
più complete chiama  
o scrivi a Arcadia srl,  
C.so Vitt. Emanuele 15,  
20122 Milano

in collaborazione  
con Elettronica 2000