

TOP TEN

MSX

COMPUTER MAGAZINE

N.9

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

10
PROGRAMMI
SU CASSETTA



COVER FLIGHT SIMULATOR
COURTESY SUBLOGIC CORP. USA

FANTASTICO!
LA POSTA
ELETTRONICA
GRATIS

SPECIALE
LINGUAGGIO
MACCHINA

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.





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

- ★ LA GESTIONE DELLA MEMORIA
- ★ BBS, O LA POSTA ELETTRONICA
- ★ IL LINGUAGGIO MACCHINA

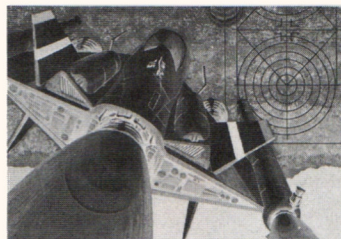
10 PROGRAMMI 10

- PLANET HUNTER
- FANTASY
- PARK TWO
- WARGAME
- GRAFICO FUNZIONI
- RESCUE
- SILVESTRO
- FRECCHE
- ATAKOR
- MSX CALC

MSX TAPE SOFT

I programmi presenti su questo numero di MSX Computer Magazine sono tutti compatibili con qualsiasi sistema MSX. Ecco per voi ben 10 programmi.

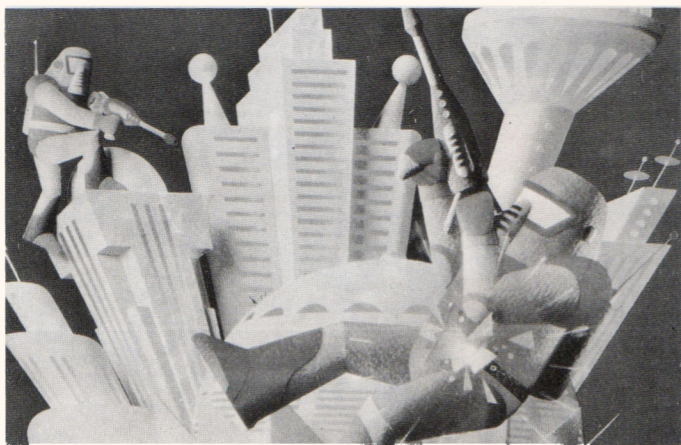
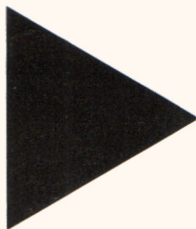
Ricordate di collegare la spina del



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 essere caricato, affinché il nastro scorra per il giusto tempo. Ap-



RESCUE ("RESCUE") di M. Belardi



La base spaziale su un lontano pianeta fuori del sistema solare è stata danneggiata da una pioggia di meteoriti e la sua esplosione è imminente. Il vostro compito è di trasportare sulla grande astronave gli uomini che si trovano sul pianeta, il quale è tuttora circondato da meteoriti che rendono difficile il movimento della navetta di salvataggio. La navetta può trasportare solo un uomo per volta e le manovre di attacco devono essere molto precise, pena la perdita del veicolo.

Il gioco è diviso in 3 fasi di 5 rounds ognuna. Il round si completa trasferendo sull'astronave tutti gli uomini

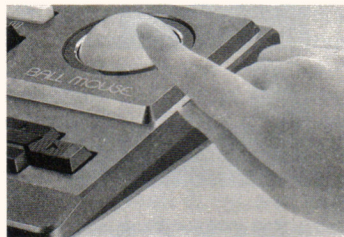
che sono nella base. Nel primo round dovrete trasportare un uomo, nel secondo due e così via fino a cinque per accedere alla fase successiva. Ogni volta che commetterete un errore, perderete la navetta e dovrete ricominciare il round.

Dall'indirizzo 49000 in poi sono contenute le 5 routines in linguaggio macchina per il funzionamento del gioco ed il buffer per memorizzarne lo schermo che, infatti, viene disegnato solo una volta.

Il programma occupa quasi tutta la RAM e deve essere caricato digitando: CLOAD "RESCUE".

pena vedete apparire sul video, dopo un comando di caricamento, la scritta OK, spegnete il registratore.

ATTENZIONE: la presentazione sul lato A va caricata con il comando: BLOAD "CAS:",R



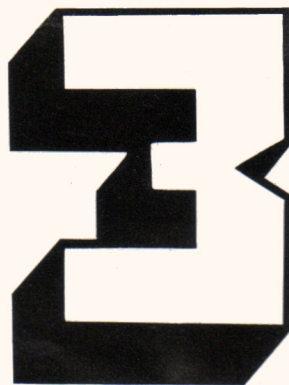
Per gli altri programmi, invece, seguite attentamente le istruzioni riportate nei riquadri.

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

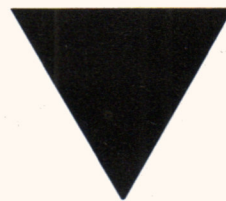
Presentazione, Rescue, Silvestro, Frece, Atakor, MSX calc.

Sul lato B:

Planet hunter, Fantasy, Park 2, Wargame, Grafico di funzioni.



FRECCE
("FRECCE")



di L. Passarella

Si tratta della simulazione del tradizionale "tiro al bersaglio" riportata su computer. Le regole del gioco sono note: bisogna centrare le varie zone del bersaglio il cui punteggio, segnato intorno, viene sottratto da un numero chiesto in precedenza dal calcolatore. Arrivati ad avere due punti, bisogna colpire la zona del bersaglio che vale un punto per poter vincere una delle manches della partita. La vincita sarà aggiudicata a chi avrà vinto più manches.

Il programma è per due giocatori (o per uno che gioca contro se stesso): all'inizio bisogna inserire le maiuscole (CAPS LOCK).

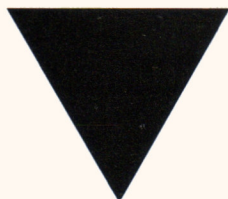
Durante il gioco appare sul bersaglio una crocetta che deve essere posizionata sul punto da colpire per mezzo dei tasti cursore (o del joystick). Il posizionamento non è così semplice perché la crocetta, simulando l'instabilità del braccio del giocatore, tende a spostarsi con maggiore facilità a seconda del livello di difficoltà prescelto. Comunque, una volta approssimato il tiro, premete la barra di spazio per far partire il dardo.

Per caricare il programma digitare CLOAD "FRECCE".



SILVESTRO
("SLVSTR")

di G. Barfi



Silvestro è uno spazzino solerte, giustamente impegnato a rendere più pulito l'ambiente. Per raggiungere il suo scopo deve raccogliere quanti più rifiuti possibile stando attento a pericoli vari fra i quali Giovannone il formicone e le mele velenose. Se malauguratamente viene a contatto con le mele o con Giovannone, il nostro Silvestro deve ingerire una speciale pillola-antidoto che lo preserva da ogni male e gli permette di continuare a lavorare.

All'inizio del gioco Silvestro ha a disposizione tre pillole (visualizzate nella parte superiore dello schermo), ma ne può ricevere altre raggiungendo rispettivamente 12000, 21000 e 30000 punti.

Il gioco è diviso in cinque fasi: nella prima Silvestro

deve ripulire un marciapiede evitando le mele che cadono dagli alberi e "saltando" Giovannone quando passa. Il nostro netturbino può essere mosso a destra e a sinistra con i relativi tasti di controllo del cursore; per farlo saltare bisogna invece premere la barra di spazio. Il suono di una campanella annuncia la fine della fase di gioco.

Nel secondo quadro Silvestro, appeso ad un palloncino, cerca di prendere al volo il maggior numero possibile di rifiuti sempre evitando le mele che cadono. Nella terza fase del suo lavoro il nostro spazzino viene messo in difficoltà dall'attacco di Giovannone e dei suoi fratelli. Il nostro eroe questa volta può difendersi con una bomboletta spray di insetticida, che potete spruzzare premendo la barra di spazio. Attenzione però a non finire troppo presto l'insetticida, perché in questo caso il gioco avrà termine.

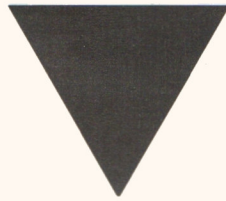
Nella quarta fase Silvestro, vero e proprio operatore ecologico, dovrà ripulire le acque di un fiume dall'alto di un ponte.

Per il suo scopo ha a disposizione un gancio che può essere azionato sempre dalla barra di spazio. A coronamento del suo utile lavoro Silvestro parteciperà infine, alla lotteria degli spazzini e, se sarà fortunato, potrà incrementare il proprio bottino di punti. Le parti più importanti del listato di questo programma sono evidenziate con dei REMarks (commenti). Il joystick (se usato) va inserito nella porta 1.

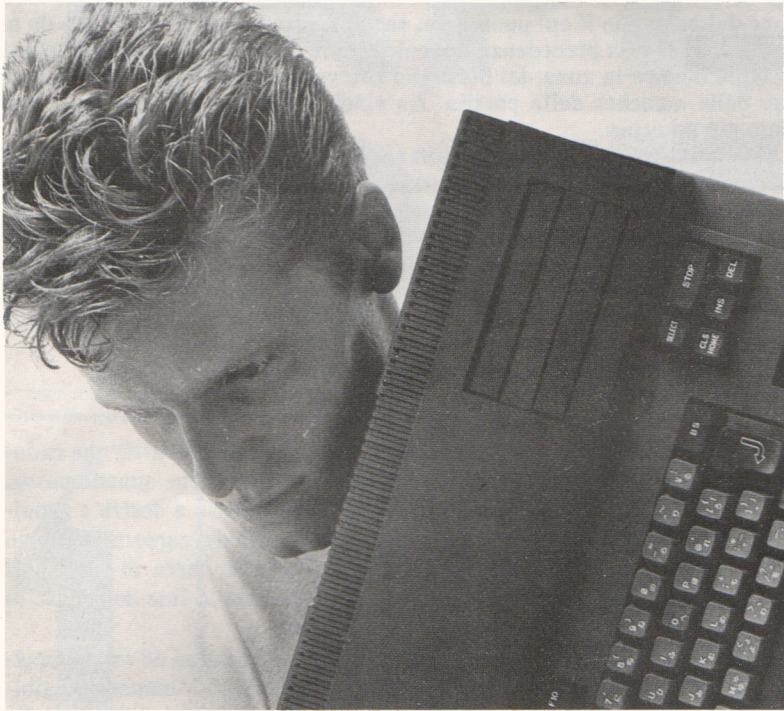
Per caricare il gioco digitare: CLOAD "SLVSTR".

4

ATAKOR
("ATAKOR")



di D. Montresor



Una stupenda avventura che ha come teatro il deserto che, come tutti gli ambienti inusitati, miete le sue vittime tra gli esseri più deboli. L'ultimo disastro ha colpito una spedizione di geologi che avevano l'incarico di scoprire nuovi giacimenti di petrolio.

Per caso, il protagonista di questa avventura è l'unico sopravvissuto, per salvarsi deve innanzitutto cercare dell'acqua ed in seguito, una volta recuperate le proprie forze, tornare alla civiltà.

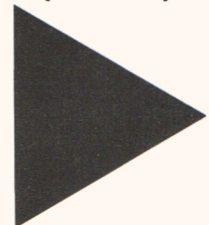
Per rispondere alle domande del programma bisogna sempre usare la prima persona singolare e scrivere le frasi complete di articolo (es. prendo la bottiglia). In alcuni casi è sufficiente scrivere una sola parola, ad esempio nel caso del verbo "guardo". Per spostarvi scrivete n, s, e, oppure o rispettivamente per nord, sud, est od ovest, o ancora su e giù per salire o scendere. Per far l'inventario degli oggetti posseduti basta scrivere: i. Per correggere una parola sbagliata, usate la freccia di spostamento a sinistra. Per sapere dal computer quali verbi possono essere usati digitate: verbi. Per salvare su nastro un'avventura scrivete: salvo; per recuperarla dopo il salvataggio battete: recupero. Non mettete spazi, accenti o apostrofi prima o dopo la frase scritta (eccetto che per "giù" che richiede l'accento).

Il programma occupa tutta la memoria disponibile poiché contiene ben 40 disegni dei luoghi visitati.

Per caricare il gioco digitare: CLOAD "ATAKOR".

5

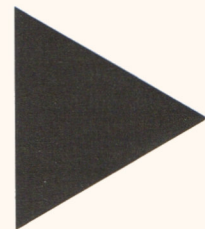
MSX CALC
("CALC")



di P. Gazzarri

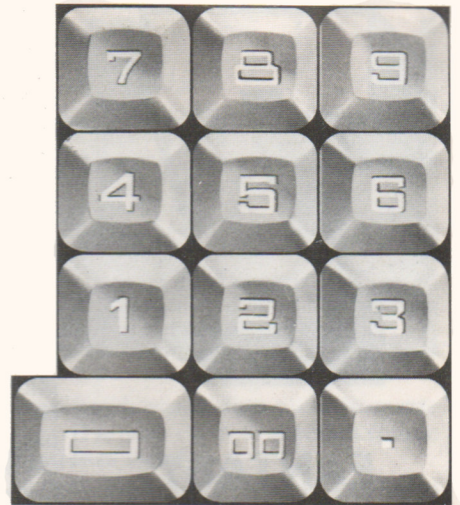
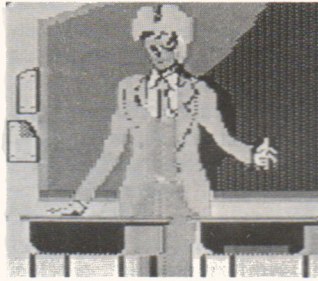
6

PLANET HUNTER
("PH")



di P. Tamburelli

Vi è mai capitato di dover fare qualche calcolo semplice e di avere allo stesso tempo il computer acceso sulla scrivania? L'approccio la prima volta è fiducioso «Caspita! Ho qui nientemeno che un computer!», ma alla fine si risolve sempre con qualche imprecazione «Stupida macchina!». Diventa un'impresa persino provare ad estrarre una semplice radice quadrata. Per trovare la radice di 48 ad esempio, bisogna dare come comando diretto: PRINT SQR (48), ed infine premere RETURN. Sembra semplice, ma quanti di voi, non ricordando che la funzione per estrarre la radice quadrata in basic è SQR, vanno alla disperata ricerca sulla tastiera del simbolo della radice, presente su ogni comune mortale macchinetta calcolatrice? (Senza contare poi quelli che non scrivono nemmeno PRINT o non mettono l'argomento tra parentesi). Beh, grazie a questo semplice programmino i vostri guai saranno ora risolti (anche se non del tutto). Il vostro MSX si trasforma in una gigantesca calcolatrice con tanto di due memorie e con le più comuni funzioni



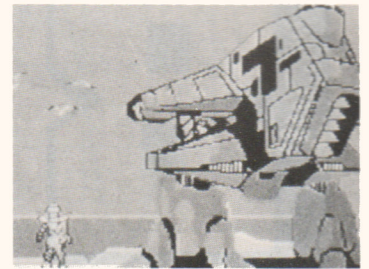
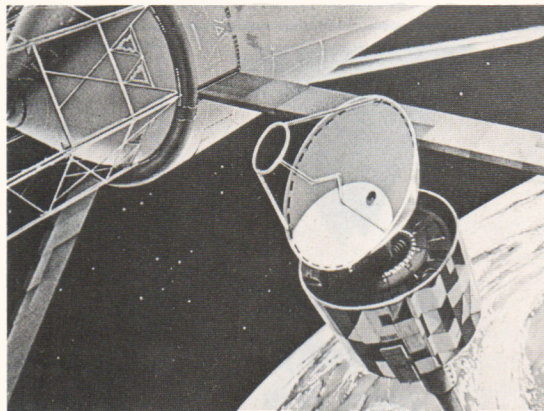
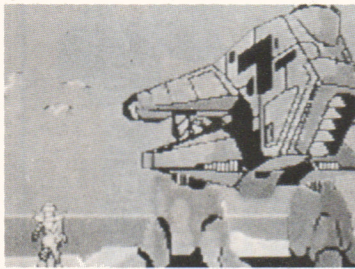
scientifiche. Sul video viene anche stampato un pro-memoria con i tasti da usare per ottenere le diverse funzioni. Per estrarre ad esempio la radice cubica di 8 dovrete premere semplicemente i seguenti tasti in sequenza:

8 G 3 =

Come dicevamo, appunto, i problemi non sono però risolti del tutto: il

computer infatti, per il modo in cui svolge internamente i calcoli, in caso di funzioni particolari (come la nostra estrazione di radice) non darà il risultato esatto (in questo caso 2), ma un'approssimazione di esso (in questo caso 1.999999999998).

Per caricare il programma in memoria digitare: CLOAD "calc".



In questo bellissimo gioco impersonate un povero astronauta naufragato su di un pianeta deserto ed inospitale. Unica vostra speranza di salvezza è costruire un'astronave riutilizzando i pezzi di quella vecchia. Questa impresa risulta piuttosto difficoltosa, causa la scarsa forza di gravità e la presenza di asteroidi vaganti. Un famigerato alieno poi attende alla vostra vita nei modi più subdoli. Per recuperare i rottami dovete semplicemente passarvi sopra a bassa velocità, ma attenzione: vanno recuperati nel giusto ordine: prima la base, poi la coda, quindi la parte centrale etc.

Man mano che recuperate un pezzo dovrete depositarlo alla base dello schermo semplicemente passandoci sopra. Una volta completata la costruzione inizieranno a scendere dall'alto i contenitori di carburante che dovrete raccogliere e depositare nel razzo (lo sgancio automatico avviene sempre passando sopra il razzo). A rifornimento completato il razzo partirà verso un nuovo pianeta dove dovrete ricominciare tutto daccapo. Ogni pezzo di astronave costruito vi frutterà 250 punti, ogni contenitore di carburante 150. A partire dal secondo schermo, se resisterete per 30 secondi

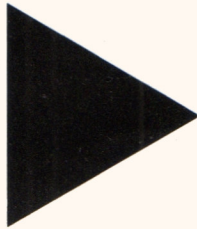
consecutivi senza farvi colpire, scenderà dall'alto un piccolo paracadute che, se recuperato, vi frutterà 1000 punti. Ogni 5000 punti avrete una vita in più. Dal terzo schermo di gioco in avanti, agli ostacoli già presenti si agguincerà un satellite killer.

Per muovervi potete usare il joystick o i tasti cursore: per scendere dovrete attendere che la forza di gravità vi attiri verso il basso.

Chi vuole analizzare il programma per trarne nuove idee trova, dalla linea 620 alla linea 1320, la routine. Per caricare Planet Hunter digitare: CLOAD "PH".



FANTASY ("FANTAS")



di M. Cerelli

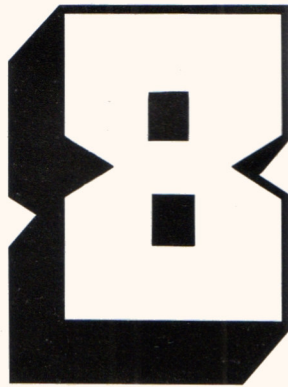
Siete alla guida di una mongolfiera a bordo della quale vi dirigerete, fra mille difficoltà, verso un'isola disabitata nella quale vi attende una giovane fanciulla naufragata che attende di essere portata in salvo da voi. Quale prima prova da affrontare apparirà, sulla sinistra dello schermo, una nave pirata intenzionata a prendervi a cannonate. Evitate di essere colpiti e atterrate con tempismo su di essa per conquistarla, nel momento in cui, terminate le munizioni, cercherà di fuggire. Non crediate che sia semplice agganciare la nave in fuga, perché la sola collisione di sprites non basterà per fermarla: sarà necessario che le coordinate X,Y dello sprite della nave coincidano perfettamente con quelle della mongolfiera. Giunti alla seconda prova, vi accorgete che si sta preparando un grosso temporale dalla presenza di nubi cupe e saette (attenti a non toccarle!). Un uccellaccio maligno, inoltre, tenterà di colpirvi con delle bombe. Utilizzate, senza toccarle, le nuvole per ripararvi dal bombardamento, e raggiungete l'estremità destra dello schermo.

La terza prova è un po' meno movimentata: noterete che vi state avvicinando alla riva di un'isola. Procedete

sempre verso la destra del quadro video. Alla quarta prova l'accoglienza non sarà delle migliori. Sarete presi a cannonate e la prontezza di riflessi è l'unico mezzo che avete per superare questa difficoltà. Se credeste poi di aver scampato il pericolo, vi sbagliate di grosso! Alla quinta prova, quattro cannoni tenteranno di colpirvi e saranno guai.

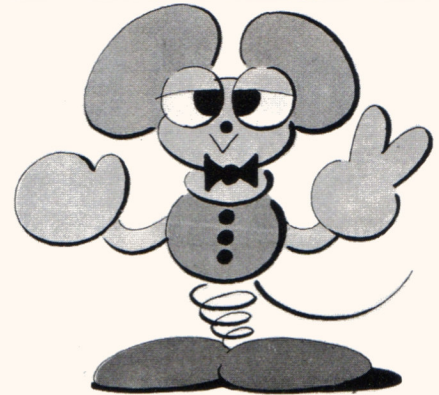
Ora (sesta prova), eccovi sul punto più alto dell'isola: ci sono un insidioso cannone ed una pedana color magenta sulla quale dovrete atterrare per passare alla... settima prova. Scesi dalla pedana, superate dei teschi in movimento e per poter proseguire bisognerà che apriate una porta con la chiave posta in basso, quasi al centro dello schermo. Giungerete così al termine della vostra avventura: vi troverete all'aria aperta, (ormai si sarà fatto buio) ma avrete finalmente raggiunto il vostro scopo perché da una casetta uscirà la vostra fanciulla. Beh ora... abbracciatela!

Per muovervi nell'avventura usate i tasti di controllo del cursore o un joystick; se rimarrete vittime di una delle tante insidie il gioco avrà termine. Per caricare il programma digitare: CLOAD "FANTAS".



PARK 2 ("PARK2")

di S. Loi



Tutti al Luna Park del nostro MSX! Ecco una grande ruota dentro la quale dovrete lanciare un razzo e fargli fare più giri possibile.

Aumentate il più possibile il livello della vostra forza (visualizzato in alto a destra sullo schermo) premendo ripetutamente il tasto "cursore a destra". Quando sarete abbastanza vigorosi e prima che il tempo sia scaduto, premete la barra di spazio per far partire il razzo (e state attenti a non investire le anatre sulla pista!).

Avete a disposizione tre tentativi nel corso dei quali dovrete totalizzare 1000 punti per poter passare allo schermo successivo, nel quale troverete un pallone svolazzante sul parco: aggiustate bene la mira e sparate, sempre con i cursori e la barra spazio o con il joystick. Per ogni colpo ben piazzato riceverete 200 punti e se colpirete il pallone per sette volte passerete al terzo

schermo con un bonus di 600 punti.

Qui potrete divertirvi ad accalappiare i cani che passeggiano sui carretti facendo una gran... cagnara. Li catturerete grazie ad una rete nera che lancerete premendo la barra di spazio o il pulsante del joystick. Se accalappiate il cane azzurro quando ha ancora la rete nera, otterrete 300 punti (la rete in questo caso diventerà gialla). Se riuscirete a fare lo stesso con il cane rosso riceverete 2000 punti (la rete diventerà bianca); con il cane verde guadagnerete 1500 punti (e la rete diventerà magenta). Per ogni rete lanciata e messa sui cani per la prima volta vi aggiudicherete 100 punti.

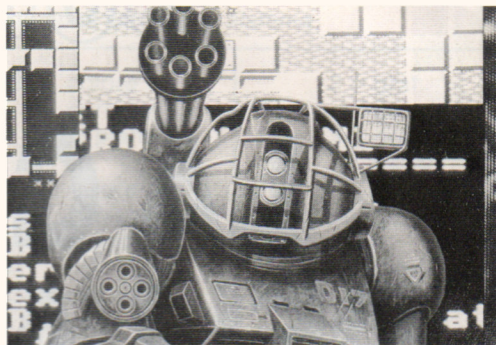
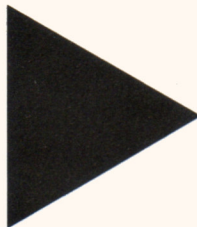
Esaminando il listato scoprirete che, pur senza sfruttare pienamente la potenza offerta dalla macchina, l'autore è riuscito ad ottenere ottimi risultati. Per caricare il programma digitare CLOAD "PARK2".



WARGAME

("WARGAM")

di C. Rocco



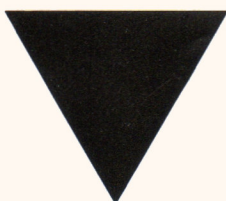
È la volta di un insolito gioco da tavolo con il quale divertirvi a combattere contro il computer: lo scopo è distruggere la maggior parte delle forze nemiche (in nero sullo schermo).

Esse partiranno con una forza complessiva superiore a quella del giocatore (le cui armate sono in rosso sullo schermo), per compensare la presunta inferiorità tattica del computer che, comunque, esegue sempre con precisione ciò che l'uomo lo istruisce a fare. Le forze armate guidate dall'MSX hanno inoltre la tendenza ad avviarsi verso la compagnia nemica più vicina. Per giocare il computer vi chiederà, per ogni unità combattente a vostra disposizione, se avete nuovi ordini da dare. Rispondete puntando la freccia su SI o su NO con i cursori (alto, basso); per confermare la scelta premete la barra di spazio. Se risponderete di voler dare nuovi ordini, apparirà un nuovo menù con i diversi ordini da impartiri-

re. L'ordine di fuoco potrà essere dato esclusivamente all'artiglieria o ai carri armati e avrete successo solo se il nemico sarà abbastanza vicino.

Scegliendo l'opzione STATO, avrete un resoconto dell'unità in questione (forza, armi, morale etc.).

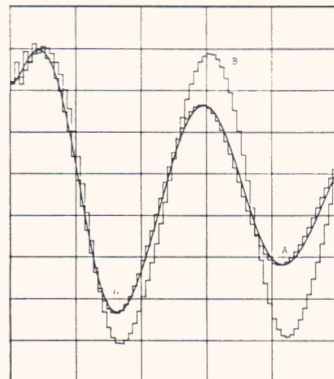
Decisa la sorte di tutte le vostre unità, spetterà al computer muovere le proprie; in seguito toccherà di nuovo a voi. I combattimenti hanno luogo quando due unità nemiche vengono in contatto: durante i combattimenti corpo a corpo ha la meglio l'unità che possiede l'arma più potente, la difesa migliore, la quantità di uomini maggiore. Inoltre, ricordate che c'è un vantaggio se si sta attaccando. Tutti questi fattori messi insieme stabiliscono il numero di perdite che si subiranno. Quando la forza di uno degli schieramenti è troppo superiore all'altra, la partita ha termine. Per caricare il gioco digitare CLOAD "WARGAM".



GRAFICO

("FUNZ")

di B. Sellan



Ecco un utile programma per disegnare il grafico di una funzione ad una variabile. La funzione da disegnare dovrà essere fornita al programma secondo la normale sintassi del basic (a tal fine riferitevi al manuale del vostro MSX). Il programma quindi, a tal scopo, chiederà:

5700 DEF FN f(x) =

A partire dalla posizione del cursore (senza dunque previamente spostarlo) dovrete inserire la funzione da studiare e quindi battere per tre volte il tasto RETURN.

Il programma ora chiederà dei dati per sapere quale parte della funzione stampare e come stamparla:

X1 = ? richiedere l'ascissa del punto più a sinistra che si vuole stampare.

X2 = ? richiedere l'ascissa del punto più a destra che si vuole stampare.

Y1 = ? richiedere l'ordinata del punto più basso che si vuole stampare (dare preferibilmente un numero negativo).

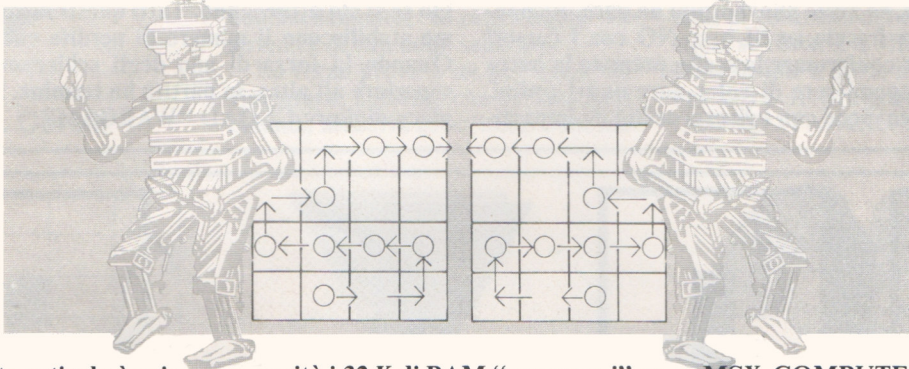
Rispondendo con RETURN a questa domanda l'asse delle ascisse sarà automaticamente posizionato al centro dello schermo. Quando la funzione presenta dei punti che non cadono nel rettangolo definito con le precedenti richieste, il computer emette un BEEP. Il BEEP viene emesso anche nel caso che si verifichi un overflow o un'operazione impossibile (come la divisione per zero). Pertanto, per disegnare una funzione della quale non si sa nulla, è consigliabile una visione allontanata di essa, ottenibile con valori grandi per X1 ed X2. L'ultima richiesta è: passo di tab. ? e vengono suggeriti i limiti.

Per caricare il programma digitare CLOAD "FUNZ".

LA GESTIONE DELLA MEMORIA

ABBIAMO COMPRATO UN COMPUTER 64K RAM MA LA DISPONIBILITÀ AL BASIC APPARE SOLO 28K... CERCHIAMO DI CAPIRCI QUALCOSA!

di G. RICCOBONO



Scopo di questo articolo è spiegare la funzione di quei tre K di memoria RAM riservati all'interprete BASIC e dare così la possibilità di un loro migliore sfruttamento, anche senza nessuna conoscenza del linguaggio macchina.

Come avrete osservato, la maggior parte dei computer MSX oggi esistenti sul mercato, ostenta maestosamente una memoria ad accesso casuale (RAM) di ben 64 K.

Ciò che molti utenti trovano sorprendente, è il fatto che il computer, all'atto dell'accensione, dichiara disponibili al BASIC solo 28 K, cioè esattamente la stessa quantità di memoria, resa disponibile al BASIC da un computer MSX dichiarante solo 32 K di RAM.

I possessori dei computer con 64 K di RAM possono, a questo punto, essere indotti a pensare di essere stati imbrogliati, per aver pagato una parte di memoria che in realtà non esiste. In

verità i 32 K di RAM "scomparsi" sono presenti dentro la macchina, solo che accedervi può rivelarsi un'impresa più difficile del previsto. Il BASIC, infatti, non è capace di riconoscere questi 32 K di RAM in più, presenti nei sistemi a 64K e così, se volete usare questo spazio extra per i vostri programmi, l'unica maniera possibile è di fare un piccolo viaggio nel mondo del linguaggio macchina.

Alcuni possessori di sistemi a 32 K possono essersi resi conto, d'altro canto, che alcuni giochi in linguaggio macchina, esistenti sul mercato, funzionano solo su computer a 64 K e che, avere a disposizione 64 K di RAM, è un obbligo se si vuole una unità a dischi.

Deve esistere allora, evidentemente, un metodo per "liberare" e utilizzare i 32 K "nascosti".

Il meccanismo attraverso cui questo può essere reso possibile è stato già soggetto di un articolo apparso su

MSX COMPUTER MAGAZINE n° 5 del Novembre '85.

Non ci occuperemo pertanto di queste tematiche ma ci rivolgeremo, invece, alla gestione standard della memoria indirizzabile direttamente da BASIC su tutti i computer MSX.

Il microprocessore Zilog Z80 può indirizzare contemporaneamente un massimo di 64 K di memoria.

Di questi 64 K indirizzabili, una metà (32 K), è occupata dalla ROM, la quale contiene il BASIC MSX. L'altra metà (cioè i rimanenti 32 K), sono costituiti dalla parte di RAM "visibile" al calcolatore, nella quale possiamo mettere i nostri programmi BASIC e le relative variabili.

Con un termine che, pur essendo improprio, rende bene l'idea, si potrebbe dire che la ROM nasconde, alla lettura del microprocessore Z80, la parte di RAM superiore ai primi 32 K.

Nasce allora spontanea un'altra domanda: perché quando accendiamo

il nostro calcolatore vengono dichiarati liberi all'utente 28815 bytes, cioè tre K meno dei 32 previsti?

Fortunatamente la risposta a questa domanda è piuttosto semplice ed è proprio di questo che ci occuperemo.

L'interprete MSX, cioè il circuito che traduce le istruzioni BASIC in una serie di codici comprensibili al microprocessore Z80, ha bisogno di un "foglio di brutta copia" dove scrivere parecchie informazioni vitali, di cui egli stesso ha bisogno durante le varie fasi di manipolazione di un programma (esecuzione, listaggio, stampa etc.).

Questa zona di memoria può essere considerata come un'area speciale, dove vengono conservate le cosiddette "variabili di sistema".

Per poter accedere a questa parte di memoria, basta una semplice conoscenza dei comandi BASIC "PEEK" e "POKE" e, al limite, una piccola conoscenza del linguaggio macchina.

Le variabili di sistema contengono informazioni come il colore di primo piano, dello sfondo e del contorno, il rapporto di baud al quale sono salvati i programmi e un numero di altre informazioni utili.

Il semplice programma BASIC che pubblichiamo produce una lista di alcune di queste informazioni. Esso funziona usando l'istruzione PEEK per andare a leggere il contenuto di alcune importanti locazioni di memoria.

La tavola in figura 1, fornisce gli indirizzi di alcune delle più importanti variabili di sistema.

Il programma legge semplicemente i valori correnti delle specifiche variabili di sistema indicate. Usando il comando POKE possiamo, se lo desideriamo, cambiare il contenuto di tali variabili.

Le informazioni stampate durante l'esecuzione, dicono a quale tipo appartengono le variabili (numeri interi, stringhe, numeri in singola o doppia precisione).

Come probabilmente sapete, se non diversamente specificato, tutte le variabili sono considerate numeriche e con una precisione di 14 cifre. Utilizzando i caratteri di dichiarazione di variabile (% per gli interni, \$ per le stringhe etc.) potete cambiare il tipo di una variabile, rispetto al tipo di variabile base sopra esplicitato. Potete anche decidere a priori il tipo di variabile, per un intero gruppo di variabili, usando le istruzioni DEFINT, DEFSTR, DEFNG e DEFDBL.

Se ad esempio viene dato il comando DEFINT A-H, tutte le variabili

FIGURA 1

Tavola degli indirizzi delle principali variabili di sistema

F3DC posizione X cursore	F3E9 colore corrente 1° piano	2 = graf. 256x192 3 = graf. 64x48
F3DD posizione Y cursore	F3EA colore corrente sfondo	F6CA tipo e seg. delle 25 variabili
FCA9 interrutt. visualiz. cursore: O = non vis. 255 = vis.	F3EB colore corrente contorno	non esplicitate: 2 = interi 3 = stringhe 4 = sing. pr. 8 = doppia pr.
F3DB interrutt. visualiz. funzione tasti: O = non vis. 255 = vis.	FCAF modo schermo corrente: O = testo 40x24 1 = testo 32x24	

IL PROGRAMMA

```

10 / *****
20 / *
30 / * programma 1: var. di stato *
40 / *
50 / *****
60 CLEAR 400
70 SCREEN 0:WIDTH 38
80 KEY OFF
90 /
100 /
110 / -- Indirizzi var. di stato --
120 /
130 /
140 FCL=&HF3E9:/ Colore 1° piano
150 BCL=&HF3EA:/ Colore sfondo
160 BDR=&HF3EB:/ Colore bordo
170 K=&HF3DB:/ Comparsa funzioni
180 TYP=&HF6CA:/ Tabella tipo variab.
190 PRINT:PRINT "Stato iniz.:var.num.14
cifre
200 PRINT STRING$(28,195)
210 PRINT
220 PRINT "Colore 1° piano:";
230 PRINT TAB(20);PEEK(FCL)
240 PRINT "Colore sfondo :";
250 PRINT TAB(20);PEEK(BCL)
260 PRINT "Colore bordo :";
270 PRINT TAB(20);PEEK(BDR)
280 PRINT

```

```

300 /
310 / Tipo variabile per ogni
320 / lettera dell'alfabeto
330 /
340 /
350 ITG$="" : /List interi
360 STT$="" : /List stringhe
370 SPR$="" : /Singola precisione
380 DBP$="" : /Doppia precisione
390 FOR I=0 TO 25
400 T=PEEK(TYP+I)
410 IF T=2 THEN ITG$=ITG$+CHR$(65+I)
420 IF T=3 THEN STT$=STT$+CHR$(65+I)
430 IF T=4 THEN SPR$=SPR$+CHR$(65+I)
440 IF T=8 THEN DBP$=DBP$+CHR$(65+I)
450 NEXT
460 IF LEN(ITG$)=0 THEN ITG$="Nessuna"
470 IF LEN(SPR$)=0 THEN SPR$="Nessuna"
480 IF LEN(STT$)=0 THEN STT$="Nessuna"
490 IF LEN(DBP$)=0 THEN DBP$="Nessuna"
500 /
510 / Stampa risultati
520 /
530 PRINT "Tipo variabili senza"
540 PRINT "dichiarazione esplicita"
550 PRINT STRING$(23,195)
560 PRINT "Interi      ";ITG$
570 PRINT "Stringhe     ";STT$
580 PRINT "Sing.prec.   ";SPR$
590 PRINT "Dop. prec.   ";DBP$
600 PRINT
610 PRINT "Appariz. funzioni:";
620 IF PEEK(K!)=0) THEN PRINT "OFF":END
630 PRINT"ON"

```

che cominciano con una qualunque lettera compresa tra A ed H, vengono considerate come intere.

Le 26 locazioni di memoria, a partire dall'indirizzo esadecimale F6CA, definiscono il tipo di variabile per ognuna delle lettere dell'alfabeto, nel caso di assenza di carattere di dichiarazione esplicita (% , \$, ! , etc.). Se ad esempio tutte le variabili il cui nome comincia con A sono definite come intere, la locazione di memoria F6CA conterrà il numero 2 (cfr. tabella fig. 1) e così via.

Il nostro "foglio di brutta copia" contiene, oltre a quelle indicate in tabella, altre informazioni, la maggior parte delle quali concernono la tecnica con la quale l'interprete traduce i nostri programmi BASIC, e non è quindi sfruttabile dall'utente BASIC.

Ad esempio, alcune istruzioni grafiche, tipicamente CIRCLE e PAINT,

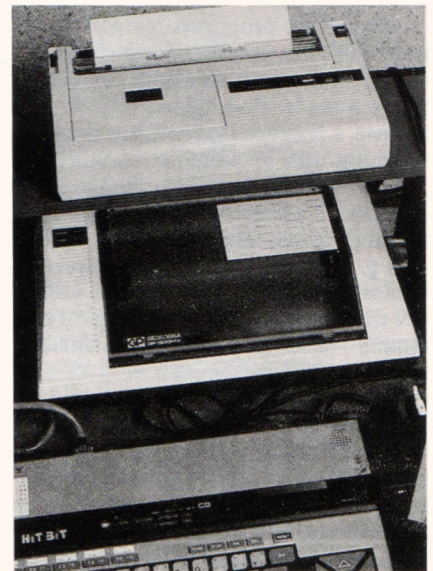
richiedono un elevatissimo numero di valori temporanei, che vengono appunto conservati, durante il tempo di esecuzione del programma, in quest'area di memoria; per non parlare, poi, di tutti i vari parametri necessari alla Music Macro Language, cioè alla generazione del suono tramite le istruzioni BASIC "SOUND" e "PLAY" (tra questi parametri si possono ad esempio menzionare il tempo, la lunghezza della nota, il profilo sonoro etc.). Oltre a tutto ciò, va anche detto, che alcune delle informazioni contenute in questa zona di RAM, possono essere facilmente riallocate, cioè riorganizzate, usando funzioni BASIC esistenti, quali la funzione BASE e la funzione VDP. A tale riguardo potete consultare, per un più dettagliato uso e significato di tali istruzioni, il manuale del BASIC MSX che sicuramente era incluso nel-

la confezione del vostro calcolatore all'atto dell'acquisto.

Oltre alle variabili di sistema, la nostra "area di brutta copia" contiene un certo numero di routines in linguaggio macchina, note con il nome di "agganci".

FAR LAVORARE LA ROM

Queste possono essere usate per gestire semplici salti a varie routines e così permettono di ordinare secondo uno schema preciso il modo di lavorare della ROM. Per esempio, consideriamo il caso in cui vogliamo sviluppare un pezzo di programma dove vogliamo utilizzare entrambe le istruzioni PRINT ed LPRINT. Durante questa fase di sviluppo, dovremo probabilmente provare il programma molte e molte volte, con la conseguenza di stampare (e cioè sprecare) una quantità di carta. Invece di stampare allora valanghe di carta, possiamo fare in modo che, durante le prove, l'istruzione LPRINT si comporti esattamente come una semplice istruzione PRINT, col risultato che i dati destinati alla stampante vengono ora indi-



rizzati al video. L'esempio citato è piuttosto complesso ed esula dagli scopi di questo articolo, il programmatore curioso e con una discreta conoscenza del linguaggio macchina, armato di un manuale decente, potrà comunque riuscire facilmente nell'impresa. Un esempio più semplice può essere quello costituito dal programma seguente, che può essere utile per scoraggiare gli altri dal listare i vostri

segue a pag. 16

IL LINGUAGGIO MACCHINA

COME PROGRAMMARE IN LINGUAGGIO MACCHINA. I CODICI ISTRUZIONE DEL MICROPROCESSORE Z80 A

(4ª PUNTATA)
di EMANUELE DASSI

Con gli argomenti di questa puntata e quelli descritti nelle precedenti, avrete oramai una panoramica quasi completa delle istruzioni dello Z80 e comunque più che sufficiente per poter scrivere programmi anche complessi. Infatti questa volta parleremo di quelle istruzioni che ogni linguaggio, sia esso a basso livello come l'Assembler o ad alto livello come il Basic, ha nel proprio vocabolario e che consentono di trasferire l'esecuzione in una parte qualsiasi del programma: le istruzioni di salto e di accesso alle subroutine.

Mentre nel Basic le istruzioni di trasferimento esecuzione (GOTO e GOSUB) si riferiscono a numeri di linea, nei programmi in Assembler, invece, ogni istruzione ha come label la propria locazione di memoria. In altre parole se in Assembler dobbiamo trasferire l'esecuzione del programma all'istruzione memorizzata nella locazione di memoria m, daremo un comando di salto all'indirizzo m. Le istruzioni di salto e di chiamata a subroutine agiscono su di un registro appartenente al gruppo di quelli secondari: il registro PC (program counter) o contatore di programma. In questo registro a 16 bit è contenuto l'indirizzo della prossima istruzione che la CPU deve interpretare. Modificando il contenuto di PC è possibile saltare da un punto all'altro del pro-

gramma ed è proprio questo quello che fanno le istruzioni di salto e chiamata.

Vi sono due tipi d'istruzioni di salto: JP e JR. La prima, JP, esegue un salto assoluto alla locazione di memoria specificata ponendo nel PC il valore del suo operando. L'istruzione JR, invece, ha come operando un valore

in complemento a due compreso tra -128 e +127 ed aggiunge tale valore a quello contenuto nel PC. In questo modo il salto si dice relativo perché non è riferito direttamente ad una locazione di memoria ma in relazione al valore corrente di PC. Per comprendere meglio quanto detto vediamo lo stesso programma con JP e JR.

PER ESERCITARSI

```

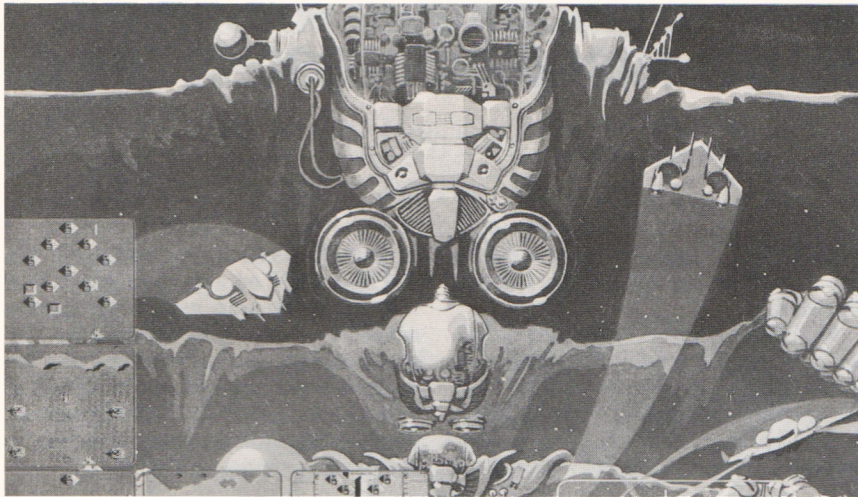
1                                     ROM 0F230H
2                                     B=LUNGHEZZA MESSAGGIO
3                                     ;
4 F230 0615                             LD    B,21
5                                     ;
6                                     DE=START MESSAGGIO IN
7                                     ;      MEMORIA
8                                     ;
9 F232 1141F2                           LD    DE,MES
10                                    ;
11                                    HL=INIZIO STAMPA IN VRAM
12                                    ;
13 F235 210800                           LD    HL,8
14                                    ;
15                                    A=CARATTERE DA STAMPARE
16                                    ;
17 F238 1A                               READ: LD  A,(DE)
18                                    ;
19                                    ; CHIAMATA SUBROUTINE IN ROM
20                                    ;
21 F239 CD4D00                           CALL 004DH
22                                    ;
23                                    ; INCREMENTA CONTATORE

```

```

24 ; POSIZIONE STAMPA E PRELIEVO
25 ; CARATTERE
26 ;
27 F23C 23 INC HL
28 F23D 13 INC DE
29 F23E 10F8 DJNZ READ
30 ;
31 ; RITORNO AL BASIC
32 ;
33 F240 C9 RET
34 ;
35 ; DEFINIZIONE MESSAGGIO
36 ;
37 F241 4D535820 MES: DEFB "MSX COMPUTER MAGA
37 F245 434F4D50
37 F249 55544552
37 F24D 204D4147
37 F251 415A494E
37 F255 45
38 END

```



```

F000 LD A,0
F002 INC A
F003 JP 0F002H

```

```

F000 LD A,0
F002 INC A
F003 JR OFDH

```

I due programmini svolgono la stessa funzione: azzerano l'accumulatore e poi eseguono all'infinito l'incremento di A.

La differenza sta nel tipo di istruzione di salto utilizzata. Nel primo caso si è usata l'istruzione JP all'indirizzo F002H dove è memorizzato il codice di INC A; nel secondo programma, invece, è stata usata l'istruzione JR con spiazzamento FD, che in complemento a due equivale al valore decimale $253-256 = -3$. Dato che il PC punta all'indirizzo d'inizio della prossima istruzione, all'atto dell'esecuzione di JR OFDH il registro PC

contiene il valore F005H che viene modificato da JR in $F005H + (-3) = F002H$ così da trasferire l'esecuzione all'indirizzo F002H, dove vi è appunto l'istruzione INC A.

Un errore comunemente commesso dai programmatori è quello di determinare il byte di spiazzamento dell'istruzione JR prima dell'esecuzione dell'istruzione stessa. Il byte di spiazzamento per JR deve essere calcolato sempre in relazione alla prima locazione di memoria dopo l'istruzione JR completa di byte di spiazzamento. Comunque queste attenzioni possono essere completamente tralasciate visto che oramai quasi tutti gli assembleri prevedono l'uso di label per agevolare la programmazione. Cioè si può scrivere il programma di prima come segue:

```

ORG 0F000H
LD A,0
AGGIUNGI INC A
JR AGGIUNGI

```

In questo caso sarà l'assemblatore a porre al posto dell'operando AGGIUNGI di JR il giusto valore per saltare all'istruzione INC A.

Vediamo ora di trarre qualche considerazione sui due tipi differenti di salto. Innanzitutto dobbiamo dire che l'istruzione JP occupa più memoria di JR, precisamente un byte in più. Infatti JP usa tre byte, uno per il codice operativo e due per l'indirizzo (da 0 a 65535). JR, invece, usa due byte, uno per il codice operativo e l'altro per il byte di spiazzamento. Questo comporta però una limitazione, mentre l'istruzione JP può trasferire il controllo del programma in qualunque locazione di memoria, l'istruzione JR può solo trasferire il controllo nelle locazioni di memoria comprese tra -128 e $+127$ byte a partire dalla locazione di memoria successiva al byte di spiazzamento. Un'ultima considerazione, i programmi contenenti istruzioni JP non sono rilocabili mentre quelli contenenti istruzioni JR generalmente lo sono.

L'istruzione JP prevede due forme sintattiche: JP nn e JP c,nn. La prima, JP nn, esegue un salto incondizionato alla locazione nn. La seconda, JP c,nn, salta alla locazione nn solo se la condizione c si è verificata. In tutte e due le istruzioni nn è un valore a due byte compreso tra 0 e 65535. L'operando c prevede le seguenti condizioni:

c condizione
NZ non zero=flag Z a 0
Z zero=flag Z a 1
NC non carry=flag C a 0
C carry=flag C a 1
PO parità dispari o assenza di overflow=flag P/V a 0
PE parità pari o presenza di overflow=flag P/V a 1
P segno positivo=flag S a 0
M segno negativo=flag S a 1

Supponiamo per esempio di voler eseguire una certa parte di programma solo se il risultato di SUB D è zero, allora scriveremo:

```
SUB D
```

```
JP Z,nn
```

resto del programma

Il programma salta alla locazione nn solo se il flag di zero è settato.

L'istruzione JP prevede anche il trasferimento del controllo del programma tramite un indirizzamento indiretto nelle seguenti tre forme: JP (HL), JP(IX) e JP(IY). Il programma quindi proseguirà dalla locazione di memoria il cui valore è contenuto rispettivamente in HL, IX e IY.

Così come l'istruzione JP anche l'istruzione JR prevede due forme sin-

CODIFICA IN L/M

```

10 CLEAR 200,&HF22F
20 SCREEN 0:WIDTH 40
30 DATA 06,15: LD B,21
40 DATA 11,41,F2: LD DE,F241
50 DATA 21,08,00: LD HL,8
60 DATA 1A: LD A,(DE)
70 DATA CD,4D,00: CALL 004D
80 DATA 23: INC HL
90 DATA 13: INC DE
100 DATA 10,F8: DJNZ -8
110 DATA C9: RET
120 DATA 4D,53,58,20: "MSX "
130 DATA 43,4F,4D,50: "COMP"
140 DATA 55,54,45,52: "UTER"
150 DATA 20,4D,41,47: "MAG"
160 DATA 41,5A,49,4E,45: "AZINE"
170 'CARICAMENTO DATI L/M IN MEMORIA
180 FOR N=&HF230 TO &HF255
190 READ A$:POKE N,VAL("&H"+A$)
200 NEXT N
210 DEFUSR=&HF230 'START L/M
220 CLS:FOR N=1 TO 500:NEXT 'PAUSA
230 A=USR(0)'ESECUZIONE L/M
240 LOCATE 8,10
250 PRINT"VUOI ESEGUIRE NUOVAMENTE"
260 LOCATE 8,11
270 PRINT"IL PROGRAMMA <S/N>";
280 INPUT R$
290 IF R$="S" OR R$="s" THEN 220
300 WIDTH 37:END

```

tattiche: JR b e JR c,b. La prima, JR b, esegue un salto alla locazione indirizzata dal registro PC più il byte di spiazzamento b. La seconda forma, JR c,b, esegue il salto solo se la condizione C è verificata. A differenza di JP, per JR le condizioni sono solo quattro: C, NC, Z e NZ.

Prima di passare alla descrizione delle istruzioni di chiamata e ritorno da subroutine dobbiamo presentare un'altra istruzione di salto relativo condizionato; si tratta dell'istruzione DJNZ. Come vedremo tra poco con un esempio, questa istruzione è assai utile e può servire per simulare in Assembler la struttura FOR-NEXT del Basic. DJNZ ha come operando un byte di spiazzamento come JR b e funziona in unione al registro B; più precisamente quest'ultimo funge da contatore. Quando viene eseguita l'istruzione DJNZ b il registro B viene decrementato di 1, dopodiché viene controllato il suo valore. Se è 0 il pro-

gramma prosegue dall'istruzione successiva a DJNZ, se invece non è 0 avviene un salto relativo. Vediamo subito un esempio: vogliamo incrementare dieci volte il valore dell'accumulatore, ecco il programmino:

```

LD B,10
SOMMA INC A
DJNZ SOMMA

```

Il contatore B viene posto a 10 tante quante sono le volte che vogliamo incrementare A. Successivamente l'istruzione DJNZ decreterà B e salterà alla locazione SOMMA dove vi è il codice operativo di INC A. Questo fino a quando B è diverso da 0, dopodiché il programma continuerà dall'istruzione successiva a DJNZ SOMMA.

Così come nel Basic per chiamare una subroutine si usa il comando GOSUB e per tornare il comando RETURN, anche in Assembler Z80 esistono l'equivalenti istruzioni: CALL e RET.

L'istruzione CALL nn chiama una subroutine all'indirizzo nn dove nn è compreso tra 0 e 65535. CALL c,nn, invece, chiama la subroutine alla locazione nn solo se si è verificata la condizione c, dove c è una delle otto condizioni previste anche dall'istruzione JP c,nn.

Il ritorno da una subroutine può essere incondizionato con l'istruzione RET o condizionato RET c, dove c è una delle otto condizioni. Vediamo con un esempio il funzionamento di CALL e RET.

```

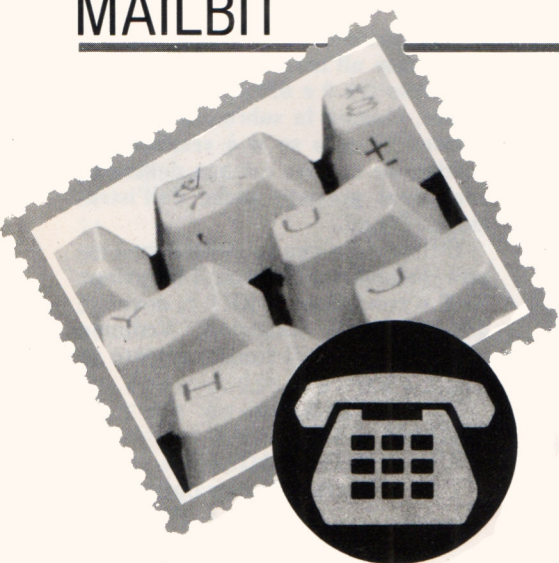
LD B,100
CHIAMA CALL SUB
DJNZ CHIAMA
      resto del programma
.....
SUB LD A,B
ADD A,B
CP 100
RET Z
ADD A,100
RET

```

L'istruzione CALL SUB chiama 100 volte la subroutine alla locazione SUB ponendo l'indirizzo dell'istruzione DJNZ CHIAMA nello stack. All'indirizzo SUB viene caricato in A il valore di B e moltiplicato per due (ADD A,B). Dopodiché l'istruzione CP confronta il contenuto di A con 100. Se sono uguali, flag Z a 1, allora ritorna (RET Z) altrimenti somma ad A il valore 100 e ritorna (RET). L'istruzione RET preleva il primo valore contenuto in cima allo stack e ritorna a quella locazione.

Esiste un'altra istruzione di chiamata: RST n. È una call ad un byte dove n è uno dei seguenti otto indirizzi: 00H, 08H, 10H, 18H, 20H, 28H, 30H, 38H. Normalmente questa particolare call viene usata per richiamare routine in ROM.

Come sempre, prima di concludere presentiamo il programmino in Assembler del listato 1 e la sua codifica in codice macchina nel programma Basic del listato 2. Il programma utilizza una subroutine in ROM per la scrittura nella VRAM in screen 0 del messaggio "MSX COMPUTER MAGAZINE". Questa subroutine ha inizio all'indirizzo 004DH e ha bisogno come valori di input il registro HL contenente l'indirizzo della VRAM dove vogliamo scrivere ed il registro A contenente il dato da scrivere. Questa subroutine modifica il valore di A. Studiate attentamente i due listati perché sono ben commentati e riassumono i concetti visti in questa puntata.



BBS 2000 ovvero la posta elettronica

Da subito, come promesso in copertina, un nuovo bellissimo servizio per voi tutti che leggete: la banca dati a disposizione diventa un vero e proprio servizio di posta elettronica. E molto, molto di più! Un nuovo computer in linea, un grosso hard disk di 20 Mbytes e un software costituito da un programma che lavora in multitasking sotto il sistema operativo DOS 3.00.

L'accesso al nuovo BBS è sempre libero ma richiede un maggior impegno da parte dell'utente che si mette in contatto. La prima volta è indispensabile inserire il proprio nominativo, indirizzo e password e in seguito utilizzare sempre lo stesso riferimento per un ottimo motivo: la posta indi-

rizzata a voi potrà essere letta solo da voi stessi se entrate nel BBS con la vostra password.

Il tempo di accesso al BBS è stato fissato per ogni utente in 15 minuti ma durante la stessa giornata si può accedere più volte. Nel BBS è disponibile un servizio di posta elettronica e un archivio di informazioni in continuo aggiornamento.

Sono disponibili alcuni comandi per gestire la posta elettronica. Prima di tutto, ricordarsi di precedere ogni comando da un punto. I comandi possono essere dati sia in minuscolo che in maiuscolo seguiti dal tasto <RETURN>.

I comandi permessi sono i seguenti:

.MSG - per visionare la lista dei messaggi lasciati da altri utenti e la posta privata (riconoscibile da un asterisco).

.ENT - permette di lasciare uno o più messaggi pubblici o privati durante il collegamento. Per inserire un messaggio bisogna conoscere il nome e cognome del destinatario (si può reperire con il comando .UTEN) e specificare se la comunicazione è personale con un asterisco. A questo punto il BBS mette a disposizione un semplice editor di linea che può contenere fino a 20 righe di 79 colonne. Se tutto è andato bene il messaggio viene inserito nel disco fisso.

Esiste un'altra categoria di comandi, utile per leggere i FI-

```
RICCARDO
ARIENTI
VIA MARIO DONATI 12
20146
MILANO
02
4233024
SONO ALLA DISPERATA RICERCA DI PROGRAMMI DI SIMULAZIONE AEREA PER PC IBM. CA MBID CON ALTRI PROGRAMMI DI OTTIMO LIVELLO. RISPONDERE SOLO SE IN ZONA MILANO E DINTORNI.
```

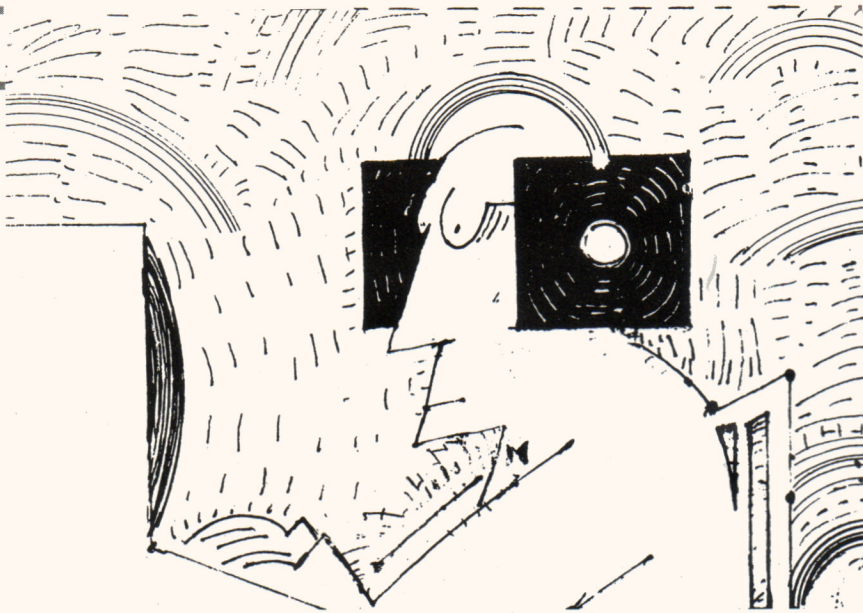
**SOS, amico cerca strumenti adatti per decollare.
Piloti da tastiera, telefonate subito!**

```
FRANCO
MISSOLI
VIA S. RITA DA CASCIA 13/A
20143
MILANO
02
816877
CERCO PERSONA DISPOSTA PROGRAMMARE IN BASIC INDUSTRIALE. OTTIMA RETRIBUZIONE LAVORO CONTINUATIVO.
```

Occasioni di lavoro! Sì, è vero, tra gli annunci ci sono anche delle buone opportunità per chi cerca lavoro. Eccone una: chi ha inviato il messaggio è un noto tecnico del settore telematico!

```
GIANLUIGI
GIAZZI
VIA MONTELEONE n62
21013
GALLARATE
0331
798942
CERCO ISTRUZIONI PER VIP TERMINAL IN ITALIANO < SONO DISPONIBILE PER PROVE CIAO A TUTTI >
```

Gianluigi ha bisogno di aiuto: contattatelo subito e (un consiglio per quanti usano Vip Terminal) leggete con attenzione il manuale d'uso, ci siamo accorti che molti non sanno usarlo.



COME SI USA

Hai un computer e un modem?
Puoi oggi stesso chiamare

02706857

ti risponderà il nostro
supercomputer sempre pronto
ventiquattrore su ventiquattro.

CHIEDI PURE

con una semplice cartolina
postale che ti venga assegnata
una password. Scrivi a
MSX Computer, C.so Vitt.
Emanuele 15, Milano 20122.

LES della banca dati. Tutti i files sono contenuti in una directory del disco fisso. Per leggere la directory utilizzare il comando .FILES; nel caso in cui scopriamo nella directory un file che interessa, per leggerlo bisogna battere il nome del testo preceduto da un punto. Per esempio: se esiste un file chiamato RETE per vederne il contenuto battere .RETE

L'ultima categoria di comandi disponibili viene utilizzata per scambiare informazioni con la banca dati stessa. Sono permessi i seguenti comandi:

.MOD - permette all'utente di cambiare la propria password e le informazioni caratteristiche (utilizzare sempre lo stesso nominativo per ricevere messaggi).

.UTEN - elenca tutti gli utenti abilitati in quel momento ad utilizzare la banca dati e ricevere messaggi.

.CIAO - fine del collegamento (scollandosi in questo modo si evita di perdere le modifiche alla password).

.HELP - fornisce la lista dei comandi disponibili.

Un ulteriore servizio è in sperimentazione e sarà attivato il più presto possibile. Si tratta di un servizio offerto agli utenti che desiderano ricevere del software via modem. Il comando .COMM abilita una procedura che permette di ricevere direttamente a casa un programma scelto dall'utente utilizzando il protocollo di comunicazione XMODEM. L'ac-

cesso è regolato da password che potranno essere richieste alla redazione.

Per concludere alcune notizie di carattere tecnico. La banca dati o BBS è accessibile telefonando al solito numero 706857 e il formato di trasmissione cambia in:

	oppure
8 bit di dati	7
1 bit di stop	2
parità NONE	NONE
300 baud/s	300 b/s

Durante il collegamento per interrompere momentaneamente la trasmissione di dati premere CONTROL <S>, per riprendere CONTROL <Q>.

L'operatore è disponibile *in diretta* giovedì dalle 15 alle 18.

FLAVIO
BERNARDOTTI
VIA TRENTO 10
15040
ALES. MONTECASTELLO
0131
355506
NATO NUOVO CBBS DEDICATO A
L'INFORMATICA.
TEL. 0131-355506
SUPPORTA XMODEM
GRATUITO.

Anche questo mese segnaliamo la nascita di una nuova BBS; se ne occupa Flavio. A lui chiediamo di tenerci aggiornati sulle sue prossime iniziative per poterle diffondere tramite la nostra Banca Dati.

LUCA
BALLERIO
VIA VINCENZO MONTI 32
20123
MILANO
02
4695630
DESIDERO INFORMAZIONI SULL'
USO DI XTALK PER PC IBM
SONO INTERESSATO ANCHE AD ALTRO
SOFTWARE PC IBM --STOP

Xtalk user's cercasi.
Luca è un buon
corrispondente
e il suo PC IBM
attende con ansia istruzioni...

ENEA
MANSUTTI
VIA AMEDEI 4
20123
MILANO
02
862057
CERCO UTENTI MACINTOSH
CON MODEM PER SCAMBIO
IDEE E PROGRAMMI.
TELEFONARE ORE PASTI

Anche gli «Applisti» con Macintosh si incontrano sulla 300 Baud della Banca Dati; invitiamo quanti usano «la mela al 68000» a contattare Enea.

È IN EDICOLA

by Computer Magazine
N. 1 - GIU/LUG 1986

MSX

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

HIT PARADE

RIVISTA DI PROGRAMMI SU CASSETTA PER COMPUTER STANDARD MSX



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

segue da pag. 10

programmi e dal manipolare indesideratamente il vostro calcolatore.

Una routine in ROM, chiamata CHPUT, chiama un programma aggancio sito alla locazione FDA4.

Normalmente questo aggancio non fa niente. Scrivendo però un brevissimo programma in linguaggio macchina, possiamo causare il "reset" automatico del computer, nel caso che un qualunque comando venga scritto sullo schermo.

Il programma in linguaggio macchina richiesto ha il seguente listato assembler:

FDA4 CRASH : JP0000

Se un tasto di funzione viene schiacciato, il computer si reinizializzerà da solo (come se lo avessimo spento e riacceso).

Il seguente programma BASIC mette inoltre in guardia un potenziale intruso dal non toccare il nostro calcolatore. Se il curioso non rispetta il nostro ammonimento riceverà una bella sorpresa!

Ricordatevi di salvare PRIMA su cassetta qualunque cosa ci sia in memoria!

```
10 REM --- antintrusi ---  
20 PRINT "NON TOCCARE !"  
30 REM --- Ling. Macc. ---  
40 FOR I=0 TO 2  
50 READ X  
60 POKE &HFDA4+I,X  
70 NEXT I  
80 DATA 195,0,0  
90 END
```

IN CONCLUSIONE

Nel caso che al vostro ritorno volete tornare a uno stato normale dei comandi, potete utilizzare il seguente programmino (sempre che nessuno abbia usato comandi del tipo LIST, INPUT, PRINT etc.):

```
500 FOR I=0 TO 2  
510 POKE &HFDA4+I,201  
520 NEXT
```

Ci sarebbero molti altri "programmini di aggancio" da descrivere, ma ciò va oltre le possibilità di questo articolo.

Per esplorare maggiormente la zona di memoria dedicata all'interprete, dovrete cercare di procurarvi dei buoni manuali, anche se, purtroppo, ben poco esiste attualmente in commercio e praticamente nulla in italiano. Qualcosa è in catalogo alla Hoepli (Milano, via Hoepli 5).

**Non lasciare solo
il tuo computer**

r. marchetti

microcomputer[®]

microcomputer[®]

la più autorevole rivista del settore

microcomputer[®]

Technimedia
00157 Roma, via Carlo Perrier 9 - tel. (06) 4513931-4515524

RIVISTA PROGRAMMI PER MSX

MSX Computer Magazine

9



Per caricare digitare CLOAD

SOMMARIO

LATO A:

- RESCUE
- SILVESTRO
- FRECCE
- ATAKOR
- MSX CALC

LATO B:

- PLANET HUNTER
- FANTASY
- PARK TWO
- WARGAME
- GRAFICO FUNZIONI

QUESTA CASSETTA
È DI

NOME _____

COGNOME _____

VIA _____

N. _____

CITTA _____

MSX COMPUTER MAGAZINE



SPLENDIDO

UN LOOK
COLORATO
PER LA TUA
CASSETTA



RITAGLIA
LUNGO
IL BORDO
SEGNATO
IN NERO
E
PIEGA
SEGUENDO
IL
TRATTEGGIO
INDICATO

PERSONALIZZA
LA
CASSETTA
CON IL
TUO NOME

MSX COMPUTER
MAGAZINE
PER LA TUA
SOFT-TECA