FEBBRAIO 1990
ANNO III

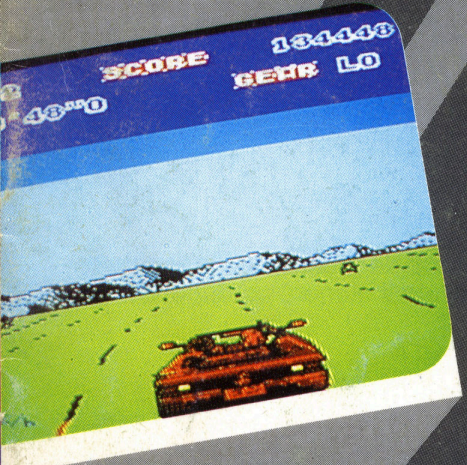
MSX DISK

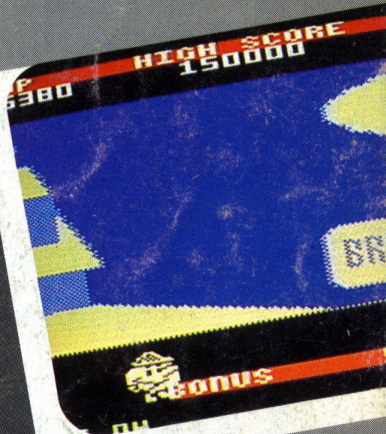
N 14

- Landpac
- Moonstear
- Go Alone
- Sexy Girl
- Mr T.
- Megasprites
- Carbon Copy
- Type Text

giochi e utilità

£ 13.000





MSX DISK - REGISTRAZIONE N. 775 DEL 20/11/87
PRESSO IL TRIBUNALE DI MILANO - GRUPPO
EDITORIALE INTERNATIONAL EDUCATION
S.R.L., VIALE FAMAGOSTA 75, 20142 MILANO -
DIRETTORE RESPONSABILE: GRAHAM
JOHNSON - DISTRIBUZIONE: MESSAGGERIE
PERIODICI S.p.A. ADERENTE A.D.N., V.LE
FAMAGOSTA, 75 - MILANO - TEL. 84.67.545

SOMMARIO

2 Sommario
Sul disco
Caricamento
Avvertenze

3 Landpac
Abbonamenti

4 Moonstear
Sexy Girl

5 Go Alone
Mr T.

7 Megasprites
Type Text

8 Carbon Copy

9 Impariamo l'Assembler (7a lezione)

SUL DISCO

1 Landpac

2 Moonstear

3 Go Alone

4 Sexy Girl

5 Mr T.

6 Megasprites

7 Carbon Copy

8 Type Text

CARICAMENTO

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

AVVERTENZE

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

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

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

LANDPAC

Benvenuti nel fantastico mondo di Landpac dove tutto quello che fino ad ora avete sempre ritenuto impossibile, accadrà davanti ai vostri occhi, come d'incanto.
Premete il tasto d'invio per aumentare i crediti di gioco (un credito ottenuto = una partita guadagnata).
Strofinatevi le mani in modo da scaldarle opportunamente. Ringraziate MSX-DISK per avervi fornito la versione di questo gioco stupendo e infine, se sceglierete l'opzione per un solo giocatore, premerete il tasto "1" mentre, per due giocatori, sceglierete il tasto "2".
Aiutate il vostro amico a compiere il tragitto di gioco senza farvi uccidere dai Fantavirus che, malefici, vogliono farvi fuori. Muovete il vostro piccolo e brutto eroe (n.d.r.) con i tasti "T" e "Y" per i movimenti sinistra e destra e fatelo saltare con la barra spaziatrice.
Consci che la miglior difesa è... la fuga, non esitate a mettere in pratica questa massima che racchiude in sé secoli di saggezza popolare.
Potrete tuttavia vendicarvi dei soprusi subiti e di tutte le vostre sofferenze ingurgitando le pillole proteiche agli spinaci gialli (ultimo ritrovato nel campo degli integratori idroproteici) che vi forniranno di una impressionante forza distruttrice che vi permetterà di fagocitare i pericolosi Fantavirus.
Good Luck boys!

COMANDI
Joystick in porta 1
Tastiera

ABBONAMENTI

Comunicato importante
da oggi potrete abbonarvi alla rivista MSX DISK e riceverla comodamente a casa semplicemente sottoscrivendo uno speciale abbonamento per 10 numeri allo specialissimo prezzo di Lit. 108.000 invece di Lit. 120.000. Potrete così assicurarvi la vostra copia e risparmiare ben Lit. 12.000.

Desidero abbonarmi alla rivista MSX DISK allo speciale prezzo di Lit. 108.000 anziché Lit. 120.000 per 10 numeri.

COGNOME _____ NOME _____

VIA _____

CAP _____ CITTA' _____ PROV. _____

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

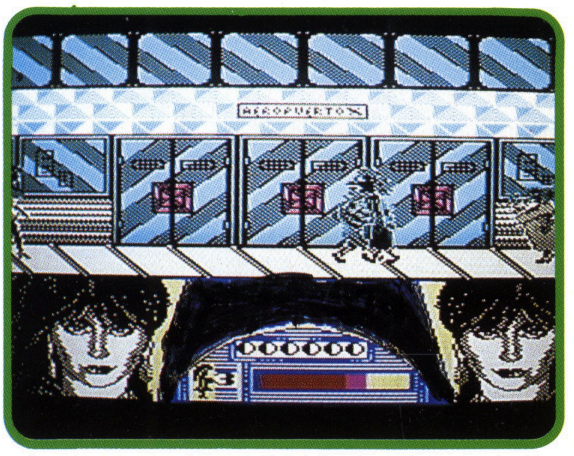
Ritagliare e spedire a Gruppo Editoriale International Education – Viale Famagosta 75 – 20142 Milano.

MOONSTEAR

Se l'horror è la vostra passione, se il brivido vi eccita, se il terrore vi entusiasma, se il panico è il vostro pane, se amate le tenebre, se siete dei pazzi omicidi (forse stiamo esagerando un po'...!), questo è proprio il gioco che fa per voi.
Guidate Mortisia all'interno del castello incantato, dove le forze del male, comandate dal malefico Bif Facciadigomma, tengono prigioniero Enry il Balordo, accusato e condannato per eccesso di sincerità e voglia di lavorare.
La dinastia Crinasti non può finire nelle mani di questa manica di assassini!
Voi lo sapete e, per questo, è vostro dovere aiutarli.
Buona morte carissimi...

COMANDI
Joystick in porta 1
Tasti:
Z = sinistra
X = destra
O = su
K = giù
SPAZIO = FUOCO

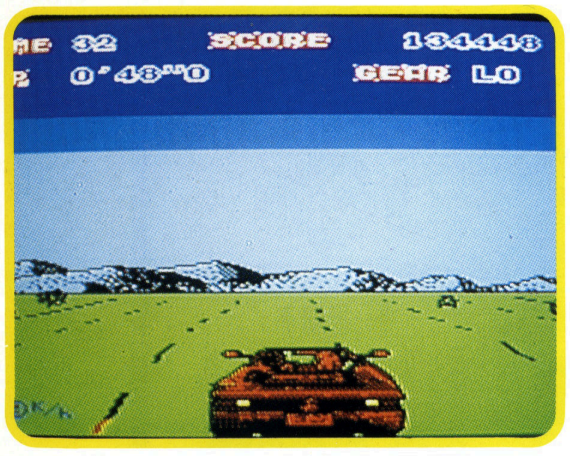
SEXY GIRL



Finalmente anche per il nostro beneamato MSX il gioco della Sexy Girl più amata d'Europa.
In giro per una tournée, la cantante dovrà cercare di difendersi dai fans scatenati che tenteranno di strapparle i vestiti di dosso.
Agguerrita e, soprattutto, intenzionata a sopravvivere, la nostra eroica star dovrà difendersi con le uniche armi a sua disposizione: cazzotti, scarpate e incredibili colpi di paraPETTO che vi lasceranno a bocca aperta.
Controllate la diva con il joystick in porta 1 o con i cursori (movimenti) e i tasti "1", "2" e "3" (colpi mortali).

COMANDI
Joystick in porta 1

MR T.

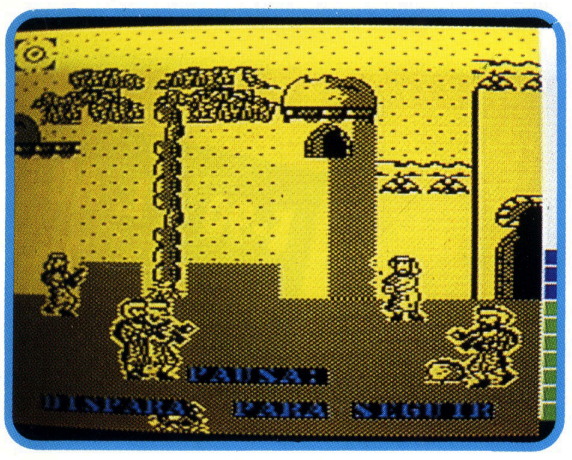


Sali sulla tua Ferrari Testarossa cabriolet e schiaccia l'acceleratore per vincere la gara che ti renderà, finalmente!, miliardario.
Devi destreggiarti in um "coast to coast" da una parte all'altra degli Stati Uniti d'America in una corsa al limite delle possibilità umane.

La competizione ammette qualsiasi tipo di vettura ma, soprattutto, qualsiasi tipo di scorrettezza pur di vincere la gara.
Armati di coraggio e di tanta forza di volontà... e che l'Agip sia con te visto che, di benzina, ne avrai parecchio bisogno.

COMANDI
Joystick in porta 1
TASTI:
CRS destro = destra
CRS sinistro = sinistra
CRS su = accelera
CRS giù = rallenta
Spazio = cambio marcia.

GO ALONE



Nel 1972 quattro uomini dell'esercito americano che facevano parte di un commando speciale furono accusati di un crimine che non avevano commesso.

Fuggirono dal carcere di massima sicurezza dove erano stati confinati e, a tuttora, sono ricercati dalla polizia militare.
Adesso lavorano come mercenari, e se avete qualche problema non dovrete fare altro che spargere la voce in giro: saranno loro a trovare voi!
Avete sicuramente capito di chi stiamo parlando.
Ma no, non della banda Bassotti.
Va bene, allora caricate il gioco e lo capirete presto.

COMANDI
Joystick in porta 1

TASTI:
Cursori = movimento mirino
Spazio = fuoco

Entra alla grande nel mondo della videoregistrazione

Approfitta anche tu della nostra grande offerta

10 **Videocassette vergini nel loro elegantissimo box rigido in plastica trasparente al prezzo speciale di**

E-90 L. 60.000
E-120 L. 75.000
E-180 L. 89.000

Compilate il coupon allegando ricevuta (o fotocopia) del versamento sul C/C n. 11319209 intestato a: Gruppo Editoriale International Education s.r.l. oppure assegno non trasferibile e spedire a:

GRUPPO EDITORIALE INTERNATIONAL EDUCATION S.R.L.
Viale Famagosta, 75
20142 MILANO

Desidero ricevere dieci cassette tipo
E-90 a Lit. 60.000 ☐
E-120 a Lit. 75.000 ☐
E-180 a Lit. 89.000 ☐
(+ contributo spese postali L. 2.500)
Allego ☐ assegno
☐ ricevuta versamento

NOME .....
COGNOME .....
VIA ..... N .....
CAP ..... CITTA' .....
..... TEL. .....

MEGA SPRITES

Questo programma vi sarà utilissimo per disegnare, correggere e verificare velocemente gli sprites 16*16 nei vari colori e formati.
La codifica in decimale verrà fornita su disco sotto forma di file basic contenente i data in un'unica linea di programma: la 100.
La barra spaziatrice colora quando il cursore risulta puntato su uno spazio vuoto e cancella se quest'ultimo appare pieno.

Il tasto "N" fornisce il negativo dello sprite.

"S" forma gli sprites.

"D" salva su disco il lavoro effettuato.

"C" cambia il colore.

"F" modifica il formato.

"R" resetta lo sprite.

"M" rimanda al menù.

TYPE TEXT

Questo semplice ma utilissimo programma vi permetterà di stampare su carta qualsiasi testo precedentemente digitato su video.

CARBON COPY

Ecco accontentati tutti coloro che, in un modo o nell'altro, hanno espresso il desiderio di veder pubblicato su MSX-DISK un buon copiatore multiuso.

Semplicissimo da usare, Carbon Copy riunisce nove funzioni in un unico menù:

1.
Consente di leggere l'header del programma su cassetta;

2.
Permette di leggere l'header del programma su disco;

3.
Copia i programmi da disco a disco o da disco a cassetta;

4.
Copia traccia per traccia l'intero disco;

5.
Editor del disco che consente di vedere il contenuto delle tracce del dischetto, di modificarle e di copiarle modificate;

6.
Permette di vedere, ed eventualmente di modificare, un programma in linguaggio macchina;

7.
Consente di copiare un programma anche protetto da cassetta a cassetta;

8.
Permette di copiare il contenuto di alcune cartridges su disco.

IMPARIAMO L'ASSEMBLER

L'ARITMETICA A 16 BIT

Nelle due puntate precedenti ci siamo occupati degli operatori logici e aritmetici a otto bit. Stavolta tratteremo invece l'aritmetica a 16 bit. Infatti, benché il nostro Z80 sia un microprocessore a 8 bit, dispone di una serie di istruzioni che ci permettono di manipolare numeri fino a 16 bit, corrispondenti a una coppia di registri. Non dimenticate però che la CPU non "vede" più di otto bit per volta, quindi le istruzioni che ci consentono di manipolare dati di dimensioni maggiori sono necessariamente più lente perché richiedono alla CPU un maggior numero di operazioni. A parte questo, le istruzioni aritmetiche a 16 bit sono simili a quelle a 8 bit viste in precedenza, tranne per l'effetto che hanno sui flag, come vedremo in seguito.

INCREMENTO E DECREMENTO

Come potete immaginare queste istruzioni servono a incrementare o decrementare di una unità il contenuto di un registro a 16 bit e hanno il seguente formato:

```
INC rr
INC IX
INC SP
DEC rr
DEC IX
DEC SP
```

Queste operazioni si comportano in modo analogo a quelle operanti su dati di otto bit, ma in questo caso, come potete notare, l'operando è sempre un registro a 16 bit. Oltre ai consueti BD, DE e HL, possiamo quindi eseguire incrementi e decrementi anche sui registri indice IX e IY nonché sul puntatore di stack SP, benché quest'ultima operazione, se usata male, rischi di falsare l'indirizzo dello stack con conseguenze disastrose per il programma in esecuzione. Vediamo comunque qualche esempio.

```
LD DE,1500
INC DE
LD IX,20000
DEC IX
```

Dopo questa serie di istruzioni possiamo constatare che il registro DE conterrà 1501, mentre IX, che è stato decrementato, conterrà il valore 19999.
A questo punto occorre fare una precisazione molto importante in quanto, mentre le istruzioni di incremento e decremento operanti sui registri a OTTO bit modificavano i flag in funzione del risultato dell'operazione o del tipo di operazione svolta, queste istruzioni operanti sui registri a sedici bit non influenzano i flag. Vedremo comunque nelle puntate successive come è possibile verificare il risultato di tale operazione.

ADDIZIONE E SOTTRAZIONE

Anche in questo caso, come per le addizioni e sottrazioni a otto bit, la CPU dispone di due set di istruzioni, uno dei quali permette di tenere conto anche del riporto dovuto al flag di carry. Vediamo intanto le addizioni semplici.

```
ADD HL,rr
ADD HL,SP
ADD IX,BC/DE
ADD IX,IX
ADD IX,SP
```

Come potete vedere le operazioni possono essere effettuate usando il registro privilegiato HL oppure uno dei registri indice IX e IY. Usando come primo operando il registro HL, possiamo utilizzare come secondo operando qualsiasi altro registro comune a 16 bit (BC,DE,HL), oppure il puntatore di stack. Utilizzando invece come primo operando un registro indice, il valore da sommare potrà essere costituito o dal registro BC, o dal registro DE, opppure dallo stack pointer, o dallo stesso registro indice usato come primo operando. Il motivo di una tale scelta da parte dei progettisti della CPU potrebbe sembrare apparentemente illogico, ma in realtà è stato determinato da effettive esigenze pratiche, come potremo contstatare quando ci addentreremo maggiormente nei meandri della programmazione assembler. Per quanto riguarda la sottrazione semplice a 16 bit c'è poco da dire in quanto non è disponibile. Esaminiamo quindi le istruzioni che ci consentono di effettuare addizioni e sottrazioni con riporto.

```
ADC HL,rr        SBC HL,rr
ADC HL,SP        SBC HL,SP
```

Anche stavolta possiamo notare come sia privilegiato il registro HL rispetto agli altri registri a 16 bit; infatti le operazioni di sottrazione devono necessariamente utilizzare come primo operando tale registro, mentre il valore da sottrarre può essere dato o da una qualsiasi coppia di registri o da SP. Per quanto riguarda invece l'addizione, notiamo che è possibile addizionare a XL un qualsiasi registro a 16 bit (operazione inversa della sottrazione) oppure sommare ad HL il contenuto dello stack pointer. Vediamo qualche semplice esempio:

```
LD HL,25000
LD BC,1000
LD DE,10000
XOR A
SBC HL,DE
ADC HL,HL
ADC HL,BC
```

Dopo aver caricato opportuni valori nei registri HL,BC e DE, eseguo l'operazione logica XOR A che, come abbiamo visto la scorsa puntata, mi permette di resettare il flag di carry, condizione indispensabile prima di eseguire una sottrazione a 16 bit con riporto, poi sottraggo DE da HL che ora conterrà 15000, sommo quindi HL a se stesso, il che equivale a moltiplicare per due, ottenendo 30000, infine sommo a HL il contenuto di BC ottenendo in HL il valore decimale 31000. Vediamo quindi come fare se vogliamo sommare un dato a 8 bit ad un altro dato a 16 bit (lo stesso vale anche per la sottrazione). Innanzitutto sappiamo che un dato a 16 bit è dato dalla composizione di due dati di 8 bit, quindi dobbiamo far coinci-

dere la parte bassa (ultimi otto bit) del dato a 16 bit con gli otto bit del dato da addizionare. Nell'esempio che segue eseguiremo la somma tra un dato contenuto nell'accumulatore e un altro dato contenuto nel registro doppio BC, restituendo il risultato in BC.

```
LD A,30
LD BC,500
LD E,A
LD D,O
PUSH BC
POP HL
ADC HL,DE
PUSH HL
POP BC
```

Dopo aver caricato i due operand in questione con dei valori a piacere è necessario che il dato di otto bit contenuto in A venga fatto coincidere con gli otto bit meno significativi di un registro a 16 bit (in questo caso DE), mentre gli otto bit più significativi vanno ovviamente posti a zero. Trasferisco allora il contenuto di BC in HL mediante stack quindi, dopo aver eseguito l'addizione, trasferisco il risultato da HL a BC come richiesto dal problema. Semplice vero?
Per finire una precisazione. Mentre le addizioni a 16 bit senza riporto alterano solo il flag di carry, le addizioni e sottrazioni con riporto influenzano anche il flag di zero, parità e segno.

SALTI E CICLI

I salti, le subroutine e i cicli sono componenti fondamentali dei programmi assembler e la loro importanza viene ulteriormente messa in evidenza dalla possibilità di poter condizionare tali istruzioni, facendo in modo che esse vengano eseguite solo al verificarsi di particolari condizioni da noi imposte. Ciò equivale a dotare i nostri programmi dell'equivalente del GOTO e IF THEN dei programmi BASIC.

SALTI ASSOLUTI

Le istruzioni di salto assoluto ci permettono di trasferire l'esecuzione del programma in un altro punto della memoria, che può essere indirizzato come segue.

```
JP addr
JP (HL)
JP (IX)
```

Sostanzialmente queste istruzioni non fanno altro che caricare nel program counter l'indirizzo specificato nell'istruzione e quindi consentono di saltare in qualunque punto della memoria. La prima di queste istruzioni salta a un indirizzo fornito esplicitamente che solitamente in un programma viene sostituito dalla label realtiva a tale indirizzo (vedi n. 4).
La seconda istruzione permette invece di saltare a un indirizzo contenuto nel registro HL, analogamente a quanto avviene nel terzo caso ove però al posto di HL troviamo uno dei due registri indice. Ma vediamo un breve esempio.

```
LD A,100
XOR A
JP PIPPO
...
...
...
PIPPO: INC BC
       LD E,2
```

In questo esempio le due istruzioni che precedono il salto non sono rilevanti, in quanto sono state poste a titolo dimostrativo; giusto per simulare una qualsiasi sequenza di istruzioni; ciò che conta è che una volta incontrata l'istruzione JP il controllo del programma sia trasferito all'indirizzo rappresentato dalla label, eludendo tutte le eventuali istruzioni esistenti tra il punto di partenza e quello di arrivo. Vediamo ora un semplice esempio, simile al precedente, ma nel quale utilizzeremo il registro HL come parametro di salto. Fate riferimento alla puntata n. 4 per quel che riguarda le pseudo operazioni.

```
IND    :EQU 8000H
        LD HL, IND
        JP (HL)
        ...
8000H   LD A,10
        AND A
```

In questo esempio l'indirizzo cui il programma dovrà saltare è già stato determinato e tale indirizzo è stato posto uguale alla label IND, quindi tale indirizzo viene caricato in HL e viene eseguito il salto, facendo riprendere l'esecuzione del programma dall'indirizzo 8000H. Naturalmente invece di usare il registro HL avremmo potuto scrivere, più semplicemente, JP IND e avremmo ottenuto lo stesso effetto. Oltre a questi tre tipi di salto, detti incondizionati, ne esiste un quarto, chiamato salto condizionato, il quale permette di porre delle condizioni allo svolgimento dell'operazione. Tale istruzione si presenta nella forma seguente:

```
JP cond.,addr
```

dove "cond" indica una delle possibili condizioni viste nella puntata relativa ai flag. Esaminiamo quindi un semplice esempio.

```
LD A,50
CP D
JP C,MAGG
...
...
```

Come avrete certamente capito il programma non fa altro che confrontare il contenuto del registro D con l'accumulatore che è stato precedentemente posto uguale a 50, quindi se D è maggiore di A (carry settato) il programma salta alla label MAGG, altrimenti, se D è minore o uguale a A il programma prosegue dall'istruzione successiva.

SALTI RELATIVI

Le istruzioni di salto relativo permettono anch'esse di dirottare l'esecuzione del programma, ma diversamente dalle precedenti non forniscono un indirizzo assoluto cui saltare, bensì un "parametro di scostamento", simile a quello usato negli indirizzamenti mediante registri indice, il quale verrà poi sommato al valore attuale del program counter. Questo scostamento, analogamente a quanto visto per i registri indice, può andare da −128 a +127 bytes.

La sintassi dell'istruzione è la seguente.

```
JR d
```

Dove d rappresenta appunto il parametro di scostamento che comunque si usa sostituire anche in questo caso con le opportune label, in quanto poi provvederà l'assemblatore stesso a calcolarne il valore in funzione della lunghezza del salto. I vantaggi del salto relativo rispetto al salto assoluto sono essenzialmente due:
1) Quando si salta a indirizzi relativamente vicini si può risparmiare un byte di memoria utilizzando l'istruzione di salto relativo al posto di quella di salto assoluto, la quale occupa 2 byte per l'indirizzo invece di un solo byte richiesto dalla prima.
2) Utilizzando le istruzioni di salto relativo si svincola il programma dall'uso di indirizzamenti assoluti e pertanto si rende lo stesso rilocabile.
Anche nei salti relativi è possibile porre delle condizioni, ma in questo caso gli unici flag che possiamo utilizzare sono quello di zero e quello di carry, per cui le condizioni pertinenti sono le seguenti: C, NC, Z e NZ. Le istruzioni di salto relativo sono inoltre molto comode per eseguire dei cicli, simili al familiare FOR NEXT del BASIC; vediamo come.

```
        LD HL,AOOOH
        LD B,100
LOOP:   LD (HL),B
        INC HL
        DEC B
        JR NZ,LOOP
        ...
```

Questo programma riempie le locazioni di memoria da AOOOH a AOOOH+100 con valori decrescenti da 100 a 1, così la locazione AOOOH conterrà 100, la locazione AOOIH conterrà 99, ecc. Questo è stato possibile mediante un ciclo che sfrutta il regitro B come indice, e partendo da 100 viene progressivamente decrementato, finché non arriva a zero, nel qual caso il programma prosegue dall'istruzione successiva, mentre finché tale registro è maggiore di zero l'istruzione di salto condizionato fa riprendere l'esecuzione a partire da LOOP. Notate che è assolutamente indispensabile provvedere a decrementare a ogni passaggio il registro B, operazione che invece viene svolta automaticamente dall'istruzione NEXT dei cicli BASIC. Tuttavia lo Z 80 dispone anche di una istruzione molto simile al NEXT del BASIC; si tratta di DJNZ d, la quale provvede a decrementare il registro B e a controllare se esso è uguale a zero. Utilizzando questa comodissima istruzione possiamo riscrivere il programma precedente come segue.

```
        LD, HL,AOOOH
        LD B,100
LOOP:   LD (HL),B
        INC HL
        DJNZ LOOP
        ...
```

Fate attenzione però, ricordate che l'istruzione DJNZ utilizza solo il registro B, quindi se vi troverete nelle condizioni di dover realizzare cicli utilizzando altri registri dovrete operare come nel primo esempio.

SUBROUTINE

Anche in assembler, come in BASIC, è possibile utilizzare delle subroutine che possono essere utilizzate in qualunque momento e in qualsiasi punto del programma. Nella programmazione assembler l'uso delle subroutine è molto più importante di quanto lo sia nel BASIC, vista la lunghezza delle procedure relative a ottenere un determinato scopo. Per esempio se in un programma sappiamo che dovremo eseguire diverse moltiplicazioni e, visto che l'assembler non dispone di simili istruzioni, possiamo costruire una subroutine che svolga questo compito e quindi richiamarla tutte le volte che ne avremo bisogno. Vediamo quindi la sintassi delle istruzioni che ci consentono di richiamare una subroutine.

```
CALL addr
CALL cond,addr
```

Questa istruzione, che può essere condizionata, non fa altro che salvare nello stack l'attuale indirizzo contenuto nel program counter, quindi saltare all'indirizzo specificato.
Esistono ovviamente anche delle istruzioni che provvedono a ritornare al programma principale al termine della routine; esse sono le seguenti.

```
RET
RET cond
```

Questa istruzione preleva dallo stack l'indirizzo di ritorno, precedentemente salvatovi e salta a tale indirizzo.
Poiché gli indirizzi di ritorno dalla subroutine sono salvati nello stack, il programmatore dovrà porre molta attenzione nel manipolare lo stack e in particolare bisogna sempre fare in modo che vi siano nella routine tante operazioni di PUSH quante di POP, in caso contrario verrebbe irrimediabilmente falsato l'indirizzo di ritorno. Vediamo ora un banalissimo esempio di uso della subroutine.

```
        LD A,20
        CALL DAF
        LD A,45
        CALL DAF
        ...
        ...
DAF     :ADD A,10
        OR 20H
        RET
```

In questo caso la subroutine aumenta di 10 il contenuto dell'accumulatore, quindi ne esegue un OR con il valore 20H, dopodiché ritorna al programma principale. Tale routine può essere eseguita innumerevoli volte per qualsiasi valore di A.
Concludiamo con un'osservazione che ci tornerà utile più avanti; nel BASIC MSX è compresa la funzione USR() che ci permette di utilizzare delle subroutine l/m, tale istruzione equivale a una CALL, quindi la nostra routine dovrà terminare con un RET per restituire il controllo al BASIC.
Si conclude qui anche questa decima puntata, densa come sempre di importanti e utilissime notizie sulla programmazione assembler.

(7 – continua)

SPLENDIDI INEDITI DEI MOSTRI SACRI DEL JAZZ

- ○ CHARLIE PARKER CDJJ 610
- ○ BENNY GOODMAN CDJJ 609
- ○ COUNT BASIE CDJJ 604
- ○ SIDNEY BECHET CDJJ 603
- ○ DIZZY GILLESPIE CDJJ 606
- ○ DUKE ELLINGTON CDJJ 602
- ○ LIONEL HAMPTON CDJJ 605

7 COMPACT DISC AL PREZZO DI L. 84.000

Desidero ricevere l'offerta "JAZZ" **codice CD7**

Allego assegno ☐ ricevuta versamento ☐

+ L. 2.500 quale contributo spese postali

NOME COGNOME

VIA N.

C.A.P. CITTÀ

Firma

Compilare il coupon allegando ricevuta (o fotocopia) del versamento effettuato sul C/C n. 11319209 intestato a **Gruppo Editoriale International Education srl** oppure assegno non trasferibile e spedire a:

Gruppo Editoriale International Education srl
viale Famagosta 75
20142 Milano

JAZZ