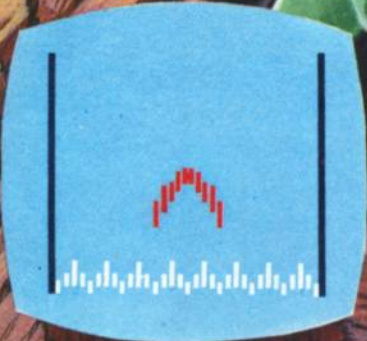


Speedy Computer

# Giochi Spaziali



GRUPPO EDITORIALE JACKSON

per ZX81, ZX Spectrum,  
BBC, TRS-80,  
Apple, Vic  
e C 64



# Giochi Spaziali

**Daniel Isaaman  
e Jenny Tyler**

## Indice

- |                                    |                                       |
|------------------------------------|---------------------------------------|
| 2 A proposito di questo libro      | 30 Allunaggio                         |
| 4 Il decollo dell'astronave        | 30 Allunaggio versione TRS80          |
| 6 Giochi intergalattici            | 31 Allunaggio versione VIC20          |
| 8 L'alieno malvagio                | 32 Allunaggio versione ZX81           |
| 10 Spara agli occhi del<br>rettile | 33 Allunaggio versione<br>ZX Spectrum |
| 12 Modulo lunare                   | 34 Allunaggio versione BBC            |
| 14 I mostri di Galacticon          | 35 Allunaggio versione<br>Apple       |
| 16 Pattuglie Aliene                | 36 Aggiunte ai programmi              |
| 18 La fascia degli Asteroidi       | 38 Scrivete i vostri<br>programmi     |
| 20 Viaggio nel futuro              | 40 Sommario di Basic                  |
| 22 La Valle della Morte            | 47 Risposte                           |
| 24 Le miniere spaziali             |                                       |
| 26 Soccorso spaziale               |                                       |

## Illustrazioni:

**Martin Newton, Tony Baskeyfield, Graham Round,  
Jim Bamber, Mark Duffin, John Bolton**

## Progetto:

**Graham Round e Roger Priddy**

**Traduzione: Stefano Dorigo**

**Programmi di Bob Merry e Richard Nash**

## A proposito di questo libro

Questo libro contiene semplici programmi di giochi per personal computer. Sono stati scritti per essere usati sui microcomputer ZX81, ZX Spectrum, BBC, VIC 20, TRS-80, C64 ed Apple e molti di essi sono abbastanza brevi da poter essere contenuti nella memoria da 1K dello ZX81. La maggior parte dei micro utilizza il linguaggio BASIC, ma ciascuno presenta le proprie variazioni o la propria versione di tale linguaggio. In questo libro, il listato presentato per ciascun programma funziona com'è sul VIC e le linee che vanno modificate per gli altri computer sono contrassegnate da simboli particolari, e riscritte sotto il listato. Il fatto che i programmi siano scritti per diversi microcomputer, implica che essi non sfruttino al massimo le potenzialità di ciascun modello. Potreste provare a rendere, in qualche modo più brevi e più ordinati questi programmi per il vostro microcomputer. Con ciascun gioco vengono fornite idee per apportare delle modifiche o delle aggiunte ai programmi, e verso la fine del libro potrete trovare spunti e suggerimenti per scrivere da soli i vostri giochi. Nel libro è anche riportata una tabella di conversione, per aiutarvi ad adattare al vostro calcolatore i programmi che trovate su riviste o su altri libri, ed un sommario dei termini BASIC usati in questo libro.

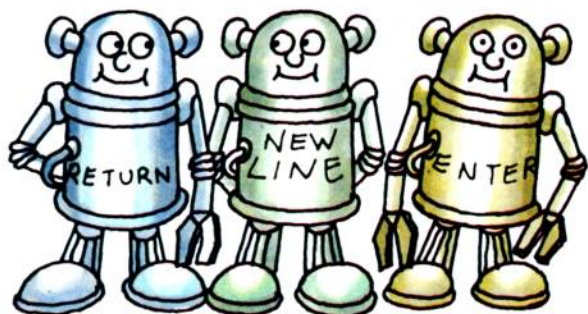
### Inserimento dei programmi da tastiera

Le linee che necessitano di modifiche, per i computers diversi dallo ZX81 sono contrassegnate da questi simboli:

- ▲ per ZX81
- \* per BBC e Acorn Electron
- per TRS-80
- per Apple
- S per ZX Spectrum

Ogni volta che vedrete il simbolo corrispondente al microcomputer che state usando, cercate sotto il listato il numero di linea accanto allo stesso simbolo, ed inserite quella linea al posto di quella originale.

*Le versioni per VIC 20 di tutti i programmi, funzionano sul C64.  
Escluso quello grafico, dovrebbero funzionare anche sui modelli PET*



### Punti da ricordare

- 1 Inserite le linee esattamente come sono stampate, includendo le punteggiature e gli spazi.
- 2 Premete il tasto RETURN, NEWLINE o ENTER alla fine di ogni linea di programma.
- 3 Controllate di volta in volta ciascuna linea.
- 4 Cercate di non dimenticarvi una linea, o di non confondere una linea con un'altra. È utile usare un foglio o un righello per tenere il segno di dove siete arrivati nel listato.
- 5 Fate attenzione ai simboli ed assicuratevi di usare le linee giuste per il vostro microcomputer.
- 6 Se state usando uno ZX81 o ZX Spectrum, ricordatevi di non inserire le istruzioni del programma lettera dopo lettera, ma di usare invece lo speciale tasto relativo a ciascuna istruzione. Potreste trovare più comodo di farvi dettare il programma da qualcuno mentre voi lo inserite. In questo caso, non dimenticate di spiegare la necessità di leggere tutte le virgole, i punti, le parentesi e gli spazi, e di distinguere tra la lettera 'O' e lo zero.

### Il debugging dei programmi

Quando avete inserito il programma, controllate sul vostro manuale come si fa a mostrarlo sul video. (Di solito si inserisce LIST, seguito dai numeri di linea della sezione che volete vedere). Controllate se avete inserito tutto correttamente. È facile

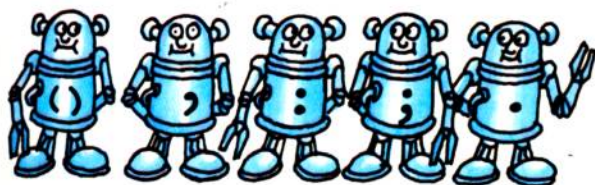




commettere errori, per cui aspettatevi pure di trovarne qualcuno. Usate il vostro manuale per vedere come effettuare modifiche al programma già inserito. Se avete dei dubbi, potete sempre reinserire la linea che vi interessa. Infatti tutti i computer sostituiscono una linea esistente con una nuova linea inserita con lo stesso numero.

Ecco una lista di controllo dei piú comuni errori da evitare:

- 1 Linea non inserita
- 2 Numero di linea errato
- 3 L'inizio di una linea attaccato alla fine della precedente.



4 Parentesi, virgole, due punti, punti e virgola, punti o spazi non inseriti, soprattutto nelle linee lunghe e complicate. In particolare, fate attenzione alle doppie parentesi.

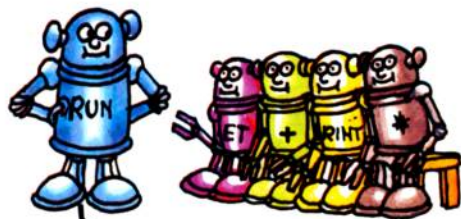
5 Uso di linee non corrette per il vostro computer.

6 La lettera "O" confusa con lo zero.

7 Uso di numeri sbagliati, ad esempio, con degli zeri di troppo.

## Iniziare a giocare

Per cominciare il gioco, dovete dare il comando RUN. In certi giochi tutto l'azione è molto veloce, per cui accertatevi di aver letto le istruzioni e di sapere esattamente quel che dovrete fare. È facile che il programma contenga



ancora qualche errore, e non giri affatto, oppure che il gioco non si svolga come dovrebbe. Certe volte il vostro computer vi fornirà un messaggio d'errore, che potrete interpretare con il manuale. Questo può anche se non sempre, aiutarvi a trovare l'errore. Fate listare di nuovo il programma sul video, e confrontatelo attentamente con il listato sul libro. Alla fine del gioco, il computer trasmetterà di solito un messaggio del tipo BREAK IN LINE 200. Per giocare di nuovo, dovete ridare il RUN.

## Esperimenti con i giochi

In tutto il libro vi sono suggerimenti per apportare modifiche o aggiunte ai programmi, ma non dovete aver paura di sperimentare delle modifiche ideate da voi. Non danneggerete il computer e potrete sempre tornare alla versione originale, se le modifiche non funzionano. Probabilmente vi verrà voglia di ritoccare la velocità dell'azione in certi giochi, specialmente dopo avervi giocato diverse volte. Nella pagina di ciascun programma troverete quale linea dover modificare.

Ogni volta che trovate PRINT, potrete modificare il messaggio che segue tra virgolette, in qualsiasi modo vi piaccia. Inoltre, a meno che non abbiate lo ZX81 con solo 1 K di memoria, potrete aggiungere altri messaggi.



Battete un numero di linea (diciamo 105, se volete inserire un messaggio tra le linee 100 e 110), poi battete PRINT, quindi il vostro messaggio tra virgolette. Se il vostro computer può generare suoni e colori, potrete usare il vostro manuale per scoprire come funzionano, e provare ad aggiungerli ai programmi di questo libro.



## Il decollo dell'astronave

**Siete il capitano di un'astronave. Siete precipitato con la vostra astronave su di uno strano pianeta, e dovete ridecollare rapidamente rapidamente con l'astronave aliena che avete catturato. Il computer di bordo vi informa della forza di gravità sul pianeta. Dovrete indovinare la forza di spinta richiesta per un decollo ben riuscito. Se ne inserite una troppo bassa, l'astronave non si staccherà dal suolo. Se ne inserite una troppo alta, il sistema di sicurezza dell'astronave viene innescato per evitare che prenda fuoco. Se siete ancora sul pianeta dopo dieci tentativi, verrete catturato dagli alieni.**





## Come funziona il programma

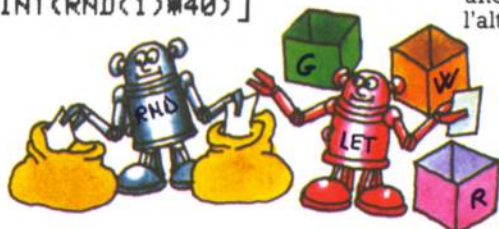
```

*■s▲●10 PRINT CHR$(147)
20 PRINT "IL DECOLLO DELL'ASTRONAVE"
■▲s30 LET G=INT(RND(1)*20)
40 LET W=INT(RND(1)*40)

```

Cancella il video.

Il computer sceglie due numeri uno tra 1 e 20, da attribuire a G, l'altro tra 1 e 40, da attribuire a W.



```

■▲s50 LET R=G*W
60 PRINT "FORZA DI GRAVITA'=" ;G
70 PRINT "INSERISCI UN NUMERO"
80 FOR C=1 TO 10

```

Moltiplica il numero in G per il numero in W. Mette il risultato in R.

Stampa FORZA DI GRAVITÀ, e il numero contenuto in G.

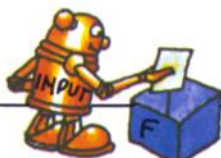
Vi chiede di inserire un numero.

Inizia un ciclo che fa ripetere al computer la sezione che segue, per 10 volte, per darvi dieci possibilità.

```

90 INPUT F

```



Immagazzina il vostro numero in F.

```

100 IF F>R THEN PRINT "TROPPO ALTO";
110 IF F<R THEN PRINT "TROPPO BASSO";
120 IF F=R THEN GOTO 190

```

Controlla il numero in F con il numero in R, e stampa il messaggio opportuno, oppure salta all'istruzione 190.

```

130 IF C>10 THEN PRINT ",
PROVA ANCORA!"

```

Stampa questo, se avete effettuato meno di dieci tentativi senza aver indovinato la risposta esatta.

```

140 NEXT C

```

Fine del ciclo. Torna all'istruzione 80 per il turno successivo.

```

150 PRINT
160 PRINT "HAI SBAGLIATO -"
170 PRINT "GLI ALIENI TI
HANNO CATTURATO!"
180 STOP
190 PRINT "DECOLLO BEN RIUSCITO!"

```

Stampa questo dopo 10 tentativi falliti.

Il listato visto funziona sul VIC e sul C 64. Per altri computers, effettuate le modifiche riportate sotto.

```

●10 HOME
s■▲10 CLS
■30 LET G=INT(RND(0)*20)
▲s30 LET G=INT(RND*20+1)
■40 LET W=INT(RND(0)*40)
▲s40 LET W=INT(RND*40+1)

```

Su tutti i modelli, con l'eccezione di ZX81 e Spectrum, potete sostituire STOP con END. (Guardate in tal caso cosa succede).



## Come rendere il gioco più difficile

Potete modificare il programma cosicché vi permetta meno di 10 tentativi. Questo si fa cambiando l'ultimo numero alla linea 80 e il numero alla linea 130 (che devono essere uguali).

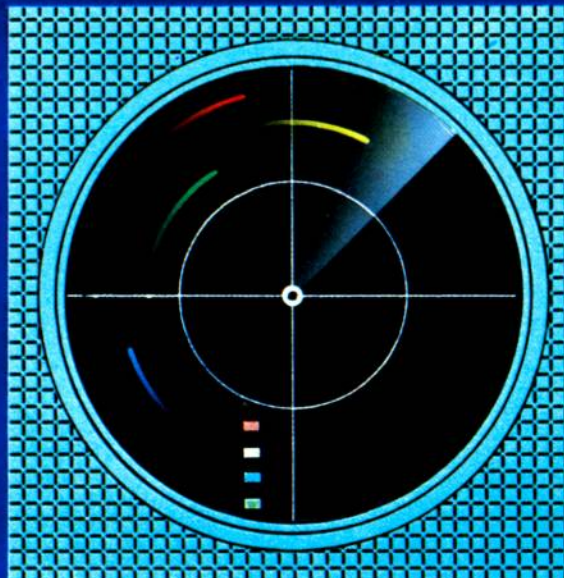
## Angolo del quesito



Potreste cambiare l'insieme di valori ammessi per la forza di spinta. Provate a vedere se ci riuscite.

# Giochi intergalattici

Le compagnie televisive mondiali si battono per ottenere l'esclusiva della trasmissione dei primi Giochi Intergalattici. Quale compagnia riuscirà a vincere la gara e posizionare il satellite nell'orbita giusta? Voi siete il responsabile del lancio per la compagnia TV Nuovo Secolo, la decisione per l'angolo di lancio e la velocità del missile dipende da voi. Coraggio!



## Come funziona il programma

- ```
10 PRINT "GIOCHI INTERGALATTICI"
S▲20 LET H=INT(RND(1)*100+1)
30 PRINT "DOVETE POSIZIONARE IN
   ORBITA UN SATELLITE"
40 PRINT "AD UNA ALTEZZA DI ";H
50 FOR G=1 TO 8

60 PRINT "INSERIRE L'ANGOLO (0-90)"
70 INPUT A
80 PRINT "INSERIRE LA VELOCITA' (0-40000)"
90 INPUT V

100 LET A=A-ATN(H/3)*180/3.14159
110 LET V=V-300*SQR(H+1/H)

120 IF ABS(A)<2 AND ABS(V)<100 THEN GOTO 210
130 IF A<-2 THEN PRINT "TROPPO VICINO"
140 IF A>2 THEN PRINT "TROPPO LONTANO"
150 IF V<-100 THEN PRINT "TROPPO LENTO"
160 IF V>100 THEN PRINT "TROPPO VELOCE"

170 NEXT G
180 PRINT "HAI SBAGLIATO"
190 PRINT "SI E' INCENDIATO"
200 STOP
210 PRINT "CE L'HAI FATTA!"
220 PRINT "GRAZIE! - LA NCTV HA VINTO"
230 STOP
```
- Sceglie l'altezza alla quale dovrete lanciare il vostro satellite, mette tale valore in H e lo stampa.
- Inizio del ciclo che vi fornisce 8 possibilità.
- Vi chiede di inserire un valore per l'angolo, e lo mette in A.
- Vi chiede un valore per la velocità, e lo mette in V.
- Usa H per calcolare quale dovrebbe essere il valore dell'angolo, e sottrae questo dal vostro valore, per vedere quanto vi ci siete avvicinati.
- Calcola quanto dovrebbe essere la velocità, e la sottrae dal valore da voi fornito.
- Controlla se vi ci eravate avvicinati abbastanza da avere vinto, e se è così, salta all'istruzione 210.
- Stampa un commento opportuno per aiutarvi nel vostro nuovo tentativo.
- Torna all'inizio del ciclo per un nuovo tentativo.
- Stampa questo dopo 8 tentativi sbagliati.
- Stampa questo se avete vinto.

Il listato visto funziona sul VIC 20 e sul C 64. Per altri computers, effettuate le modifiche riportate sotto.

```
■20 LET H=INT(RND(0)*100+1)
S▲20 LET H=INT(RND*100+1)
```

## Aggiunte al programma

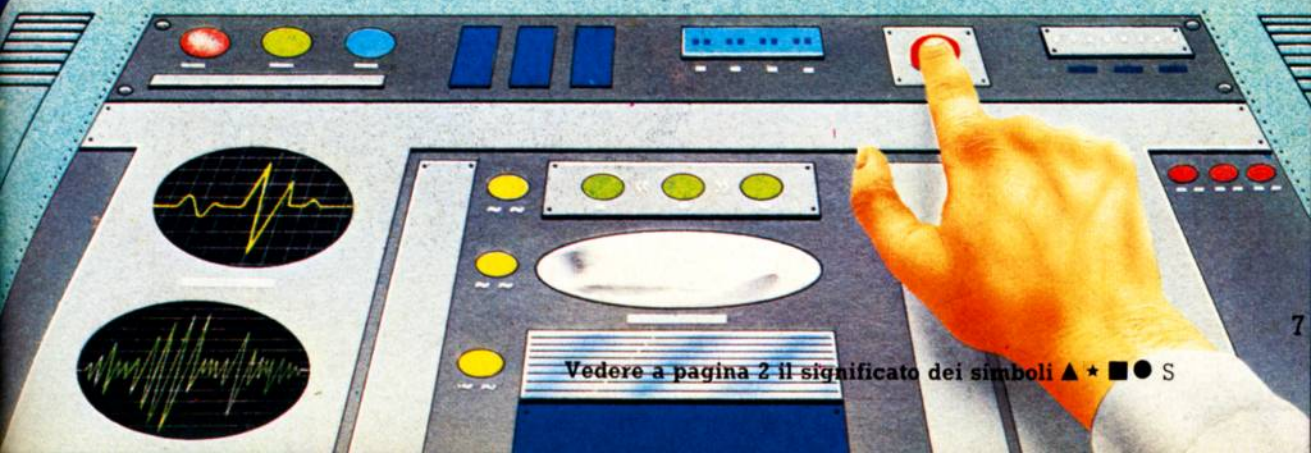
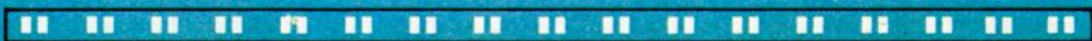
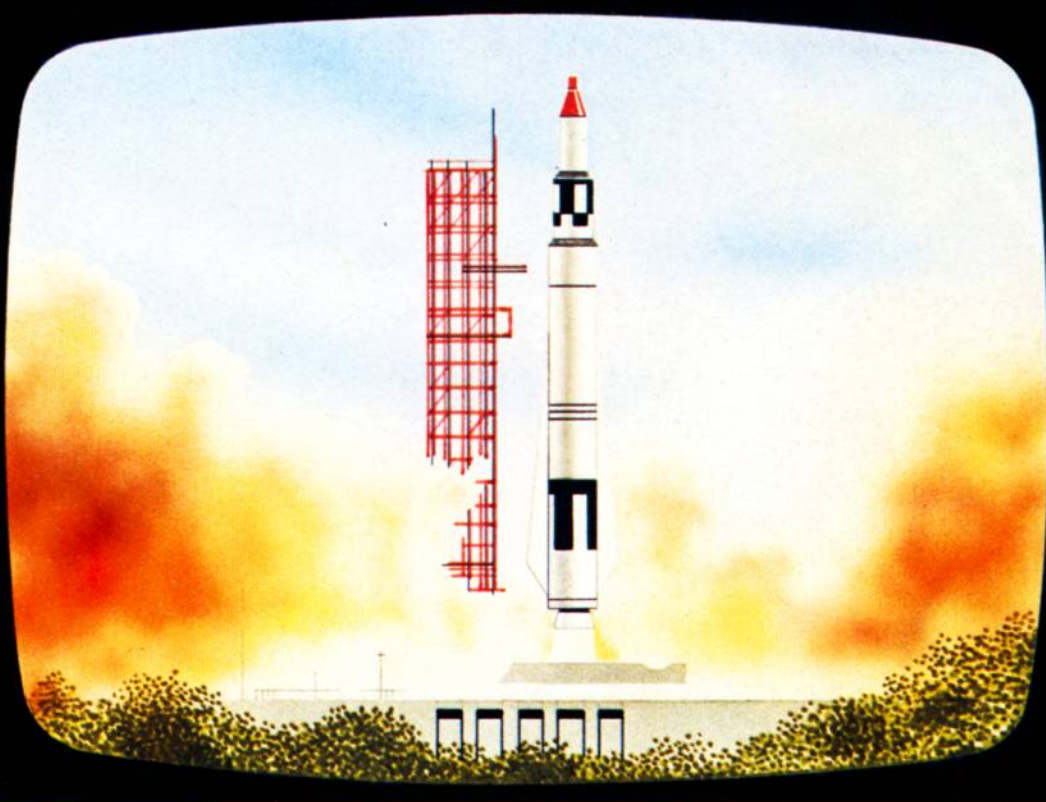
Queste altre tre linee faranno in modo che il computer vi attribuisca dei punti di bonus, in funzione di quanto rapidamente abbiate fatto un lancio riuscito.

```
222 LET B=INT(1000/G)
225 PRINT "HAI GUADAGNATO UN"
227 PRINT "BONUS DI ";B;" CREDITI"
```



## Angolo del quesito

Modificate il programma in modo che se vincete, vi offra automaticamente la possibilità di giocare di nuovo, aggiungendo il bonus che avete già guadagnato ad un nuovo eventuale bonus? (suggerimento: dovrete modificare due linee ed aggiungerne un'altra). Quanto riuscite a giocare prima che la NCTV vi licenzi?



Vedere a pagina 2 il significato dei simboli ▲ ★ ● ● S



# L'alieno malvagio

Sotto di voi, nello spazio piú buio e piú profondo, è in agguato Eiron, l'Alieno Malvagio.

Siete riusciti a disattivare tutte le sue armi, meno quelle a corto raggio, ma egli è ancora in grado di rendere invisibile la sua astronave.

Voi sapete che egli si trova da qualche parte entro il reticolo tridimensionale visibile sullo schermo della vostra astronave, (vedere sotto), ma dov'è? Avete quattro

bombe spaziali. Ciascuna può venir fatta esplodere in una posizione nel

reticolo specificata da tre

numeri compresi da 0 e 9, che

vi verranno richiesti dal vostro computer. Riuscirete a liberare

lo spazio bombardando il

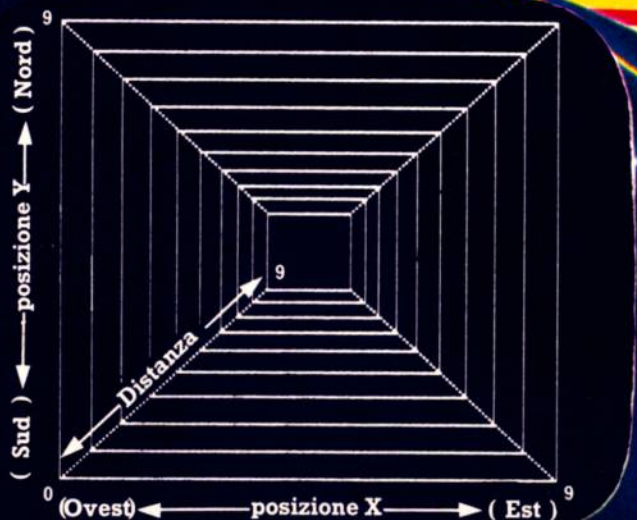
Malvagio Elron prima che

questi riesca ad

avvicinarsi

furtivamente e a

catturarvi?





## Come funziona il programma

```

S ⊕ ● ▲ 5 PRINT CHR$(147)
10 PRINT "L'ALIENO MALVAGIO"
20 LET S=10
30 LET G=4

```

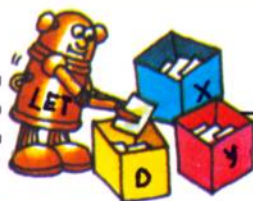
Fissa le dimensioni del reticolo.

Fissa il numero di turni a vostra disposizione.

```

S ■ ▲ 40 LET X=INT(RND(1)*S)
S ■ ▲ 50 LET Y=INT(RND(1)*S)
S ■ ▲ 60 LET D=INT(RND(1)*S)

```



La posizione di Elron viene fissata da queste tre linee, che determinano 3 numero compresi tra 0 e la dimensione del lato del reticolo.

```
70 FOR I=1 TO G
```

Inizio di un ciclo che fa ripetere al computer per G volte le successive 15 linee.

```

80 PRINT "POSIZIONE X (0-9)"
85 INPUT X1
90 PRINT "POSIZIONE Y (0-9)"
100 INPUT Y1
110 PRINT "DISTANZA (0-9)"
120 INPUT D1

```

Questa sezione richiede i vostri tre numeri, e li immagazzina in X1, Y1, e D1.

```

130 IF X=X1 AND Y=Y1 AND D=D1
    THEN GOTO 300

```

Controlla se i numeri erano quelli giusti, e in tal caso salta all'istruzione 300.

```

140 PRINT "L'OBIETTIVO ERA A ";
150 IF Y1>Y THEN PRINT "NORD";
160 IF Y1<Y THEN PRINT "SUD";
170 IF X1>X THEN PRINT "EST";
180 IF X1<X THEN PRINT "OVEST";
190 PRINT
200 IF D1>D THEN PRINT "TROPPO LONTANO"
210 IF D1<D THEN PRINT "NON LONTANO
    ABBASTANZA"

```

I numeri da voi inseriti vengono confrontati con la posizione di Elron e viene stampato un messaggio.



```
220 NEXT I
```

Fine del ciclo. Il programma lo ripete per un altro tentativo.

```

230 PRINT "IL TEMPO E' SCADUTO!"
240 STOP
300 PRINT "*BOOM* L'HAI PRESO!"
310 STOP

```

Stampa questo se avete effettuato tutti i tentativi a vostra disposizione senza successo.

Stampa questo se avete indovinato.

Il listato visto funziona sul VIC 20 e sul C 64. Per altri computers, effettuate le modifiche riportate sotto.

```

● 5 HOME
S ⊕ ■ ▲ 5 CLS
■ 40 LET X=INT(RND(0)*S)
S ▲ 40 LET X=INT(RND*S)
■ 50 LET Y=INT(RND(0)*S)
S ▲ 50 LET Y=INT(RND*S)
■ 60 LET D=INT(RND(0)*S)
S ▲ 60 LET D=INT(RND*S)

```

## Rendete il gioco piú difficile

Questo programma è stato scritto in un modo che vi permette di modificarne facilmente la difficoltà cambiando le dimensioni del reticolo. Per far questo, attribuite un valore diverso a S nella linea 20. Se aumentate le dimensioni del reticolo, vi serviranno piú bombe spaziali per avere una buona possibilità di riuscire a bombardare Elron. Le potrete avere modificando il valore di G alla linea 30.

## Angolo del quesito

Siete capaci di modificare il programma in modo che il computer vi chieda di inserire un numero, come livello di difficoltà, da attribuire a S, invece di avere S prefissato? (Suggerimento: limitate il valore di S all'intervallo tra 6 e 30, ed usate INT(S/3) come valore di G alla linea 30).



## Spara agli occhi del rettile

Siete in trappola! Dovunque vi voltiate, riuscite a cogliere il freddo luccichio metallico degli occhi di un rettile spaziale, prima che questi scivoli di nuovo dietro una roccia. Lentamente i rettili si fanno strada verso di voi, attendendo il momento favorevole per avvolgervi con la loro bava appiccicosa. Fortunatamente avete con voi un cannone a protoni. Gli occhi dei rettili spuntano in quattro punti diversi dello schermo e questi corrispondono ai tasti da 1 a 4. Premete il tasto giusto mentre gli occhi del rettile sono sullo schermo e lo distruggerete. Vi sono in tutto 10 rettili - più ne distruggete e maggiori saranno le vostre possibilità di fuggire.

### Come funziona il programma

```
10 PRINT "GLI OCCHI DEL RETTILE"
20 LET S=0
30 FOR T=1 TO 10
```

Azzerare il punteggio all'inizio del gioco.

Inizio del ciclo che vi permette dieci tentativi.

```
S * 40 PRINT CHR$(147)
```

Cancella il video.

```
S ■ 50 FOR I=1 TO INT(RND(1)*300+200)
60 NEXT I
```

Ciclo di ritardo, che ha una durata variabile, in funzione del valore di RND.

```
S ■ 70 LET R=INT(RND(1)*4+1)
```

Determina un numero tra 1 e 4, e lo mette in R.

```
▲ 80 ON R GOSUB 240,270,300,330
```

Salta ad una di quattro subroutine, in funzione del valore di R. Questa determina due numeri che corrispondono ad una posizione sullo schermo — "A" spazi in orizzontale, e "D" linee verso il basso — quindi salta di nuovo, all'istruzione 350, per muovere il cursore in questa posizione.

```
90 PRINT "00"
```

Stampa gli "occhi del rettile" in questa posizione.

```
S ▲ 100 FOR I=1 TO 150
* ■ 110 GET R$
▲ 120 IF R$(<>)" " THEN GOTO 140
130 NEXT I
140 IF VAL("0"+R$)<>R THEN GOTO 210
150 LET S=S+1
```

Vengono effettuati dei cicli per vedere se state premendo un tasto.

Se è così, il computer salta all'istruzione 140 per controllare se è quello giusto.

Aumenta di 1 il punteggio.



Il comando GOSUB fa saltare il computer dall'esecuzione del programma principale a quella di una "subroutine" (vedere alla pagina seguente). Il comando RETURN alla fine delle subroutine lo rimanda al programma principale.



S ■ ● ▲ 160 PRINT CHR\$(147) \_\_\_\_\_

170 GOSUB 350 ]  
180 PRINT "\*" ]

▲ 190 FOR J=1 TO 300 ]  
200 NEXT J ]

210 NEXT T \_\_\_\_\_

220 PRINT "HAI DISTRUTTO ";  
S;"/10 RETTILI" ]

230 STOP

240 LET D=5

250 LET A=1

260 GOTO 350

270 LET D=1

280 LET A=9

290 GOTO 350

300 LET D=5

310 LET A=18

320 GOTO 350

330 LET D=10

340 LET A=7

350 FOR I=1 TO D

360 PRINT

370 NEXT I

380 PRINT TAB(A);

390 RETURN

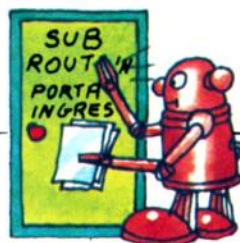
Cancella il video.

Torna a mandare il cursore alla stessa posizione, e stampa un asterisco.

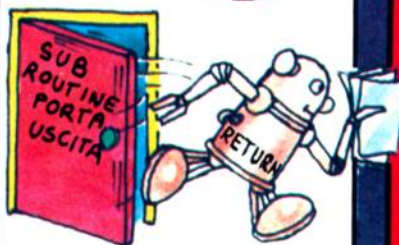
Ciclo di ritardo che fa restare l'asterisco sul video abbastanza a lungo perché voi lo possiate vedere.

Ripete il ciclo per il tentativo successivo.

Stampa il punteggio.



Subroutines.



### Come cambiare la velocità

Potete rendere il gioco più veloce cambiando l'ultimo numero alla linea 100 con un valore più basso.

### Come sfruttare una maggiore ampiezza del video

Questo programma si adatta alla minima ampiezza del video utilizzata (che è quella del VIC 20). Per gli altri computers, potete aumentare il valore di A alle linee 250, 280, 310 e 340. Il vostro manuale vi dirà qual'è la massima ampiezza che il vostro computer può utilizzare.



Il listato visto funziona sul VIC 20 e sul C 64. Per altri computers, effettuate le modifiche riportate sotto.

● 40. 160 HOME

S ■ ● ▲ 40. 160 CLS

■ 50 FOR I=1 TO INT(RND(0)\*300+200)

S ▲ 50 FOR I=1 TO INT(RND\*30+20)

■ 70 LET R=INT(RND(0)\*4+1)

S ▲ 70 LET R=INT(RND\*4+1)

▲ 80 GOSUB 210+30\*R

S ▲ 100 FOR I=1 TO 20

● 105 R\$=""

■ 110 R\$=INKEY\$(1)

● 110 IF PEEK(-16384)>127 THEN GET R\$

S ▲ 110 LET R\$=INKEY\$

S ▲ 190 FOR J=1 TO 40

### Angolo del quesito

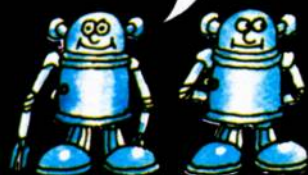
Provate a cambiare il programma in modo che i rettili compaiano in più di quattro punti dello schermo? E di aggiungere anche altri rettili?

# Modulo lunare

Siete ai comandi di un modulo lunare che trasporta un gruppo di astronauti sulla luna. Per poter arrivare sulla superficie lunare senza rischio dovrete rallentare la vostra discesa, ma ciò richiede il consumo di carburante, e la vostra scorta è limitata. Il computer vi comunicherà la quota iniziale, la velocità e la scorta di carburante, e vi chiederà quanto carburante volete bruciare. Quindi ricalcolerà la vostra nuova quota e la velocità. Bruciando una quantità di carburante di 5 unità manterrete costante la velocità; una quantità maggiore la ridurrà. Cercate di avere una velocità il più possibile vicina allo zero quando toccate il suolo. Sarete capaci di atterrare?

```
★■●▲ 10 PRINT CHR$(147)
        20 PRINT "MODULO LUNARE"
        30 LET T=0
        40 LET H=500
        50 LET V=50
        60 LET F=120
        70 PRINT "TEMPO";T,"ALTEZZA";H
        80 PRINT "VEL. ";V,"CARBUR.";F
```

Notare le virgole e i punti e virgola alle linee 70 e 80. Fate delle prove, non inserendole o scambiandole di posto, per vedere cosa succede.



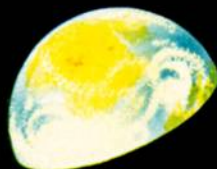
```
90 IF F=0 THEN GOTO 140
```

```
100 PRINT "CARBURANTE (0-30)"
110 INPUT B
120 IF B<0 THEN LET B=0
130 IF B>30 THEN LET B=30
140 IF B>F THEN LET B=F
```

```
150 LET V1=V-B+5
```

```
160 LET F=F-B
```

```
170 IF (V1+V)/2>=H THEN GOTO 220
```



```
180 LET H=H-(V1+V)/2
```

```
190 LET T=T+1
```

```
200 LET V=V1
```

```
210 GOTO 70
```

```
220 LET V1=V+(5-B)*H/V
```

```
230 IF V1>5 THEN PRINT "TI SEI SCHIANTATO"
```

```
240 IF V1>1 AND V1<=5 THEN PRINT "OK - MA CON FATICA"
```

```
250 IF V1<=1 THEN PRINT "BUON ATTERRAGGIO!"
```

```
260 STOP
```

Il listato visto funziona sul VIC 20 e sul C 64. Per altri computers, effettuate le modifiche riportate sotto.

● 10 HOME  
★■▲ 10 CLS



## Come funziona il programma

Stabilisce i valori iniziali per il tempo, l'altezza, la velocità e il carburante, e li stampa.

Se non avete più carburante, il computer salta più avanti nel programma, tralasciando la sezione che vi richiede quanto carburante bruciare. Quindi passa a stampare una serie di dati progressivi sull'andamento della vostra discesa verso la superficie della luna.

Vi richiede un numero pari alla quantità di carburante che volete bruciare, e controlla che sia entro i limiti richiesti.

Calcola la vostra nuova velocità  $V_1$ .

Calcola il nuovo livello della vostra scorta di carburante.

Controlla se la distanza percorsa a seguito della vostra ultima azione sia maggiore o uguale all'altezza che vi separa dalla luna. Se è così, siete atterrati. Il computer quindi passa più avanti nel programma, per vedere che tipo di atterraggio abbiate fatto.

Calcola la vostra nuova quota.

Incrementa di 1 il tempo.

Scriva il nuovo valore della velocità in  $V$ , così questa verrà stampata con la linea 80, nel turno successivo.

Ritorna all'inizio del ciclo per il turno successivo.

Calcola la vostra velocità al momento dell'impatto, e ne valuta le conseguenze sul tipo di atterraggio.

## Aggiunte al programma

Se aggiungete le seguenti linee, vedrete un asterisco stampato ad ogni vostro tentativo. La distanza dell'asterisco dal limite sinistro dello schermo corrisponde alla vostra quota sopra la luna.

```
85 FOR I=2 TO H/500*NN
86 PRINT " ";
87 NEXT I
88 PRINT "*"
```

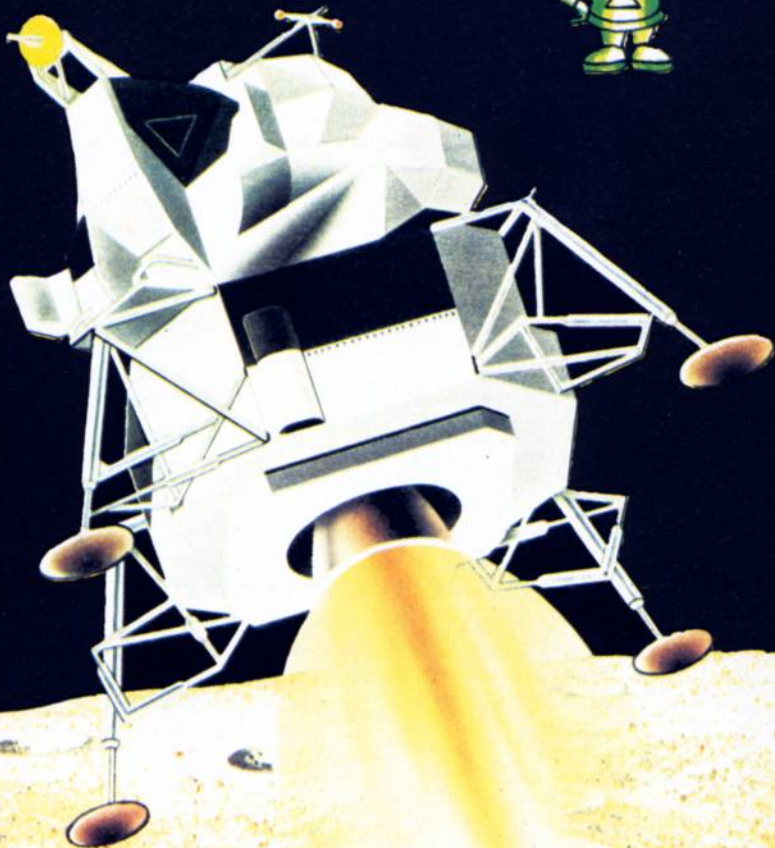
Sostituite ad nn l'ampiezza del vostro schermo.

## Modifiche da provare

Provate a modificare i valori di H, V e F alle linee da 40 a 60, e guardate cosa succede.

## Angolo del quesito

Potreste rendere il gioco più facile aumentando la massima velocità permessa perché l'atterraggio sia ben riuscito. Come cambiereste il programma, a tale proposito?



# I mostri di Galacticon

L'atterraggio su Galacticon è stato facile ma nessuno vi aveva avvertito che qui avreste trovato alcuni tra i più pericolosi mostri di tutto l'Universo conosciuto.

Sotto la minaccia di ciascuno dei mostri, dovrete usare una delle vostre armi — una pistola a raggi, un cannone a tryptoni, o una spada a laser — da usare contro di esso. Saprete scegliere quella giusta? In tal caso, potrete sopravvivere e conquistare Galacticon.



## Come funziona il programma

```
10 PRINT "I MOSTRI DI GALACTICON"  
20 DIM M$(4)
```

Predisporre uno spazio di memoria (una "matrice") chiamata M\$, con 4 caselle — M\$(1), M\$(2), M\$(3) e M\$(4) — una per nome di ogni mostro.

```
30 LET N=4  
40 LET M=5  
50 LET M$(1)="SULFACIDOR"  
60 LET M$(2)="FLAMGONDAR"  
70 LET M$(3)="BALNOLOTIN"  
80 LET M$(4)="GOLANDOR"
```

Fissa a 4 il numero dei mostri.

Fissa a 5 il numero di persone del vostro gruppo.

Pone i nomi dei 4 mostri nella matrice.

Potete usare questa routine di mescolamento in qualsiasi programma dove vogliate mescolare qualcosa.

Queste linee mescolano i nomi dei mostri. Il computer effettua un ciclo di queste istruzioni per n volte. Ad ogni ciclo, sceglie due numeri compresi tra 1 ed N, e scambia i nomi nelle caselle di tali numeri. T\$ è una variabile stringa usata solo durante l'operazione di scambio.

```
90 FOR I=1 TO N  
S ■▲ 100 LET A=INT(RND(1)*N+1)  
S ■▲ 110 LET B=INT(RND(1)*N+1)  
120 LET T$=M$(A)  
130 LET M$(A)=M$(B)  
140 LET M$(B)=T$  
150 NEXT I
```

Inizio del ciclo di 8 turni.

```
160 FOR T=1 TO 8  
S ■●▲ 170 PRINT CHR$(147)
```

Cancella il video.

```
S ■▲ 180 LET R=INT(RND(1)*N+1)  
190 PRINT "ARRIVA IL MOSTRO..."  
200 PRINT "E' UN ";M$(R)  
210 PRINT "QUALE ARMA USI (R,S,T) "  
220 INPUT R$
```

Scegli uno dei mostri e ne stampa il nome.

```
S ▲ 230 LET W=ASC(R$)-81+R  
S ▲ 240 LET W=W+3*(W>3)+3*(W>6)  
250 IF W=2 THEN GOTO 300
```

Se W vale 2, il computer salta all'istruzione 300 per comunicare che avete ucciso il mostro.

Se W vale 3, il computer salta all'istruzione 320 per comunicare: "nessun effetto".





```

260 IF W=3 THEN GOTO 320
270 PRINT "NON IN USO. UNA PERSONA DEL TUO GRUPPO"
280 PRINT "E' STATA MANGIATA!"
290 GOTO 360
300 PRINT "L'HAI UCCISO"
310 GOTO 380
320 PRINT "NESSUN EFFETTO"
S ■ 330 IF RND(1) > .4 THEN GOTO 380
340 PRINT "HAI FATTO ARRABBIARE IL ";M$(R); " CHE"
350 PRINT "HA UCCISO UNO DEL TUO GRUPPO"
360 LET M=M-1
370 IF M<1 THEN GOTO 440
S ▲ 380 FOR I=1 TO 300
390 NEXT I
400 NEXT T
410 PRINT "SEI SOPRAVVISSUTO"
420 PRINT "PER CONQUISTARE GALACTICON"
430 STOP
440 PRINT "SIETE TUTTI MORTI!"
450 STOP

```

Stampa questo se W non vale né 2 né 3. Poi salta all'istruzione 360 per diminuire di 1 il numero di persone nel gruppo.

A seguito della linea di "nessun effetto" vi è una probabilità casuale che il mostro si arrabbi e uccida qualcuno.

Controlla se vi sono delle persone. In caso contrario, salta all'istruzione 440 per comunicare.

Ciclo di ritardo per far rimanere i messaggi sul video abbastanza a lungo perché voi possiate

Torna all'inizio del ciclo per il turno successivo.

Stampa questo se vi sono rimaste delle persone alla fine dei turni concessi.

**Il listato visto funziona sul VIC 20 e sul C 64. Per altri computers, effettuate le modifiche riportate sotto.**

```

■ 100 LET A=INT(RND(0)*N+1)
S ▲ 100 LET A=INT(RND*N+1)
■ 110 LET B=INT(RND(0)*N+1)
S ▲ 110 LET B=INT(RND*N+1)
● 170 HOME
S ⊕ ▲ 170 CLS
■ 180 LET R=INT(RND(0)*N+1)
S ▲ 180 LET R=INT(RND*N+1)
S 230 LET W=CODE(R$)-81+R
▲ 230 LET W=CODE(R$)-54+R
S ▲ 240 LET W=W-3*(W>3)-3*(W>6)
■ 330 IF RND(0) > .4 THEN GOTO 380

```

```

S ▲ 330 IF RND > .4 THEN GOTO 380
S ▲ 380 FOR I=1 TO 20

```


### Angolo del quesito



**Ci sono almeno 4 modi per rendere questo gioco più difficile. Riuscite a vedere quali sono?**



# Pattuglie Aliene



Siete il comandante di un incrociatore interstellare che, per danni subiti ad uno dei motori iperspaziali, ha sconfinato in un'area proibita. Le navicelle di pattuglie aliene vi stanno attaccando e, per complicare le cose, stanno usando un dispositivo di disturbo del vostro radar, che gli fa fornire delle informazioni sbagliate. Fortunatamente il vostro computer conosce un codice che potrete usare per calcolare le posizioni esatte delle navicelle nemiche. Ma dovete essere veloci — esse non rimangono a lungo nello stesso posto! Il vostro computer stamperà una lettera (la falsa locazione del nemico) ed un numero di codice. Dovrete contare mentalmente per un numero di lettere dell'alfabeto pari a quello dato, a partire dalla lettera stampata, e inserire la lettera a cui siete arrivati. Per es., leggendo M4 dovreste inserire Q, leggendo C2 dovreste inserire E, e così via. L'inserimento di una lettera fa sparare automaticamente la vostra pistola laser, perciò se la vostra lettera è quella giusta, segnerete un colpo a vostro favore. Potrete scegliere ogni volta la difficoltà del gioco. Questa è data da un numero tra 1 e 10, e rappresenta il massimo numero di lettere che potrete dover aggiungere a quella data. Troverete 10 pattuglie aliene ogni volta. Scoprite quante ne riuscite a colpire.

## Come funziona il programma

```
S ■ ● ▲ 10 PRINT CHR$(147)
20 PRINT "PATTUGLIE ALIENE"
30 PRINT
40 PRINT "DIFFICOLTA' (1-10)"
50 INPUT D
60 IF D<1 OR D>10 THEN GOTO 50
70 LET S=0
80 FOR G=1 TO 10
```

Vi richiede un livello di difficoltà, lo mette in D, e controlla che sia entro i limiti ammessi.

Pone il punteggio iniziale uguale a zero.

Inizio del ciclo che vi permetterà dieci tentativi.

Sceglie una lettera compresa tra A e la lettera che dista un numero di posti pari al livello di difficoltà da voi scelto, dalla fine dell'alfabeto.

Scelgite un numero compreso tra 1 e D.

```
S ■ ▲ 90 LET L$=CHR$(INT(RND(1)*(26-D)+65))
S ■ ▲ 100 LET N=INT(RND(1)*D+1)
```

```
■ ● ▲ 110 PRINT CHR$(147)
120 PRINT
130 PRINT L$,N
S ▲ 140 FOR I=1 TO 200+D*50
S ● ▲ 150 GET I$
S ■ ● ▲ 160 IF I$<>" " THEN GOTO 190
170 NEXT I
180 GOTO 200
```

Stampa la lettera e il numero.

Controlla se stiate premendo un tasto, nel qual caso salta all'istruzione 190.

Se non avete premuto nessun tasto, il computer salta all'istruzione 200, che lo rimanda all'inizio del ciclo per il turno successivo.

Controlla se abbiate premuto il tasto giusto, nel qual caso, incrementa di 1 il vostro punteggio.

Fine del ciclo. Torna all'inizio per il turno successivo.

Stampa il punteggio dopo 10 tentativi.



```
S ▲ 190 IF I$=CHR$(ASC(L$)+N)
THEN LET S=S+1
200 NEXT G
210 PRINT "NE HAI COLPITE ";S;" / 10"
220 STOP
```

Il listato visto funziona sul VIC 20 e sul C 64. Per altri computers, effettuate le modifiche riportate sotto.

```
● 10,110 HOME
■ ● S ▲ 10,110 CLS
■ 90 LET L$=CHR$(INT(RND(0)*(26-D)+65))
S 90 LET L$=CHR$(INT(RND*(26-D)+65))
▲ 90 LET L$=CHR$(INT(RND*(26-D)+38))
■ 100 LET N=INT(RND(0)*D+1)
S ▲ 100 LET N=INT(RND*D+1)
● 140 FOR I=1 TO 100*D+50
S ▲ 140 FOR I=1 TO 200*D+5
● 145 I$=""
■ 150 LET I$=INKEY$(1)
● 150 IF PEEK(-16384)>127 THEN GET I$
S ▲ 150 LET I$=INKEY$
S ▲ 190 IF I$=CHR$(CODE(L$)+N) THEN LET S=S+1
```

## Come cambiare la velocità

Se per voi il gioco è troppo veloce, mettete un numero più alto a metà della linea 140 (cioè al posto di 20 o 200). Con un numero più basso invece potrete accelerarlo.



## Come rendere il gioco più difficile

Potreste cambiare l'1 alle linee 40 e 60 in un 3, per permettere solo livelli di difficoltà di 3 o maggiori.

## Angolo del quesito

Sapreste modificare il sistema di punteggio perché i punti corrispondano al numero in codice, così guadagnate 1 punto se il codice vale 1, 2 punti se vale 2, ecc.?

# La fascia degli Asteroidi

State viaggiando attraverso la Fascia di Asteroidi. Per evitare di scontrarvi con gli asteroidi, dovete distruggerli, e la forza richiesta per far ciò dipende dalle loro dimensioni.

Gli asteroidi compaiono sul video del vostro computer come gruppi di stelle. Per distruggerli dovrete premere il tasto con il numero corrispondente al numero di stelle. Siate pronti - gli asteroidi vi vengono incontro fitti e veloci.

## Come funziona il programma

|                                             |                                                                                                      |
|---------------------------------------------|------------------------------------------------------------------------------------------------------|
| 10 PRINT "LA FASCIA DEGLI ASTEROIDI"        | Pone a zero il punteggio iniziale.                                                                   |
| 20 LET S=0                                  |                                                                                                      |
| 30 FOR G=1 TO 10                            | Inizia un ciclo di 10 turni.                                                                         |
| S*■▲● 40 PRINT CHR\$(147)                   |                                                                                                      |
| S■▲ 50 LET A=INT(RND(1)*18+1)               | Sceglie un numero per la posizione orizzontale degli asteroidi sul video. Mette in A questo numero.  |
| S■▲ 60 LET D=INT(RND(1)*12+1)               | Sceglie un numero (da 1 a 12) per la posizione verticale degli asteroidi sul video, e lo mette in D. |
| S■▲ 70 LET N=INT(RND(1)*9+1)                | Sceglie un numero (da 1 a 9) per il numero di stelle che compongono l'asteroide.                     |
| 80 FOR I=1 TO D                             | Muove il cursore di D linee verso il basso sullo schermo.                                            |
| 90 PRINT                                    |                                                                                                      |
| 100 NEXT I                                  |                                                                                                      |
| 110 FOR I=1 TO N                            | Compie un ciclo per N volte, stampando ogni volta un asterisco nella posizione giusta.               |
| 120 IF I<>1 AND I<>4 AND I<>7 THEN GOTO 150 |                                                                                                      |
| 130 PRINT                                   |                                                                                                      |
| 140 PRINT TAB(A);                           |                                                                                                      |
| 150 PRINT "*";                              |                                                                                                      |
| 160 NEXT I                                  |                                                                                                      |
| 170 PRINT                                   | Effettua un ciclo per vedere se state premendo un tasto, nel qual caso salta all'istruzione 240.     |
| S*■▲ 180 FOR I=1 TO 100                     | Stampa questo se avete esaurito il tempo a disposizione per sparare.                                 |
| S*■▲ 190 GET Q                              |                                                                                                      |
| 200 IF Q<>0 THEN GOTO 240                   | Controlla se il vostro numero sia uguale ad N, e in caso contrario salta all'istruzione 270.         |
| 210 NEXT I                                  |                                                                                                      |
| 220 PRINT "TI SEI SCHIANTATO"               |                                                                                                      |
| 225 PRINT "CONTRO L'ASTEROIDE"              |                                                                                                      |
| 230 GOTO 290                                |                                                                                                      |
| 240 IF Q<>N THEN GOTO 270                   |                                                                                                      |

```

250 PRINT "L'HAI DISTRUTTO!"
260 LET S=S+1
270 IF Q<N THEN PRINT "TROPPO DEBOLE"
280 IF Q>N THEN PRINT "TROPPO FORTE"

```

Stampa questo se avete premuto il numero giusto.

Incrementa di 1 il vostro punteggio.

Confronta il numero da voi inserito con N, e stampa un messaggio di conseguenza.

Ciclo di ritardo per mantenere i messaggi sullo schermo abbastanza a lungo perché voi li possiate leggere.

Torna all'inizio del ciclo per il turno successivo.

Stampa il vostro punteggio dopo 10 turni.

```

S * ■ ▲ 290 FOR I=1 TO 250
300 NEXT I
310 NEXT G
320 PRINT "COLPITI ";S;" SU 10"
330 STOP

```

Il listato visto funziona sul VIC 20 e sul C 64. Per altri computers, effettuate le modifiche riportate sotto.

```

● 40 HOME
S * ■ ▲ 40 CLS
■ 50 LET A=INT(RND(0)*18+1)
S ▲ 50 LET A=INT(RND*18+1)
■ 60 LET D=INT(RND(0)*12+1)
S ▲ 60 LET D=INT(RND*12+1)
■ 70 LET N=INT(RND(0)*9+1)
S ▲ 70 LET N=INT(RND*9+1)
● 175 Q=0
■ 180
S ▲ 180 FOR I=1 TO 10
■ 190 Q=INKEY(100)-48
● 190 IF PEEK(-16384)>127 THEN GET Q
S ▲ 190 LET Q=VAL("0"+INKEY$)
■ 210
■ 290 FOR I=1 TO 500
S ▲ 290 FOR I=1 TO 50

```

### Modificare la velocità del gioco

La linea 180 (la 190 per il BBC) controlla quanto tempo avete a disposizione per premere un tasto. Per accelerare il gioco, cambiate l'ultimo numero alla linea 180, o il numero tra parentesi alla linea 190 per il BBC, sostituendolo con uno minore.

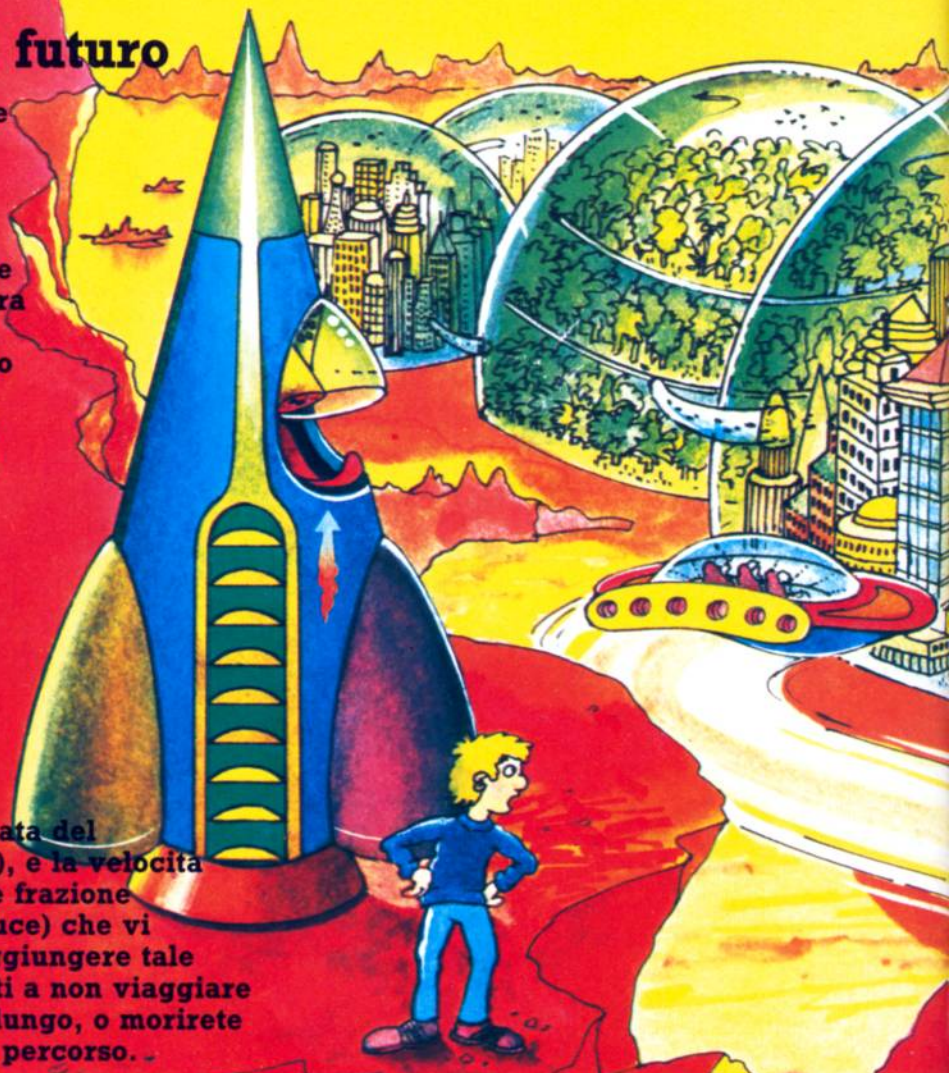
### Angolo del quesito

Sapreste modificare il sistema di punteggio così che, per ciascun asteroide, possiate guadagnare tanti punti quante sono le stelle in esso contenute?



# Viaggio nel futuro

Immaginate di essere su un'astronave in viaggio a velocità quasi pari a quella della luce. Il tempo scorre più lentamente all'interno della vostra navicella che al di fuori. Perciò, essendo partito per un lungo viaggio nello spazio, potrete tornare sulla terra più avanti nel tempo di quanto non indichino gli orologi nella navicella. In questo gioco, il computer vi comunica quanti anni devono trascorrere sulla terra prima del vostro ritorno. Starà a voi decidere la durata del viaggio (in anni luce), e la velocità della navicella (come frazione della velocità della luce) che vi permetteranno di raggiungere tale obiettivo. State attenti a non viaggiare lentamente troppo a lungo, o morirete di vecchiaia lungo il percorso.



## Come funziona il programma

```

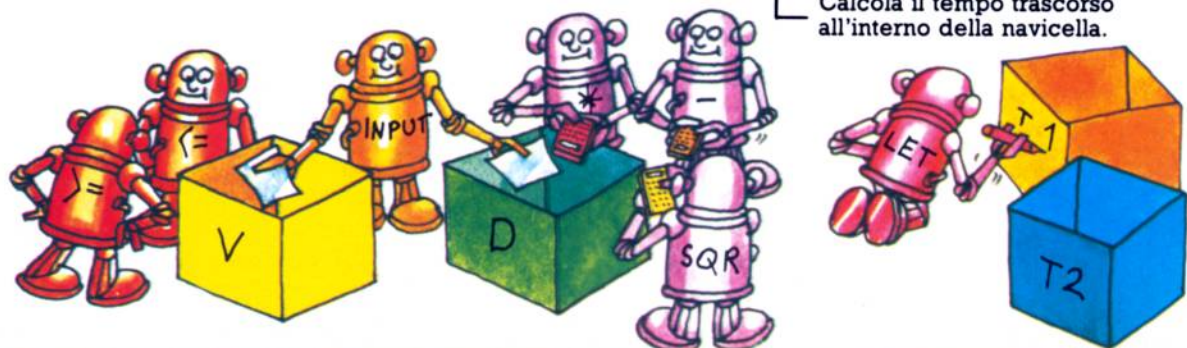
S ■ ● ▲ 10 PRINT CHR$(147)
    20 PRINT "VIAGGIO NEL FUTURO"
S ■ ▲ 30 LET T=INT(RND(1)*100+25)
    40 PRINT "DEVI TORNARE FRA ";T
    50 PRINT "ANNI."
    60 PRINT
    70 PRINT "VELOCITA' DELL'ASTRONAVE (0-1)"
    80 INPUT V
    90 IF V>=1 OR V<=0 THEN GOTO 70
    100 PRINT "DISTANZA DEL VIAGGIO"
    110 INPUT D
    120 LET T1=D/V
    
```

Sceglie un numero intero, compreso tra 25 e 124, di anni che devono trascorrere prima del vostro ritorno, e lo stampa.

Vi richiede un valore per la velocità, e controlla che sia entro i limiti del viaggio.

Vi richiede un valore per la durata del viaggio.

Calcola il tempo trascorso all'interno della navicella.





```

130 LET T2=T1/SQR(1-V*V)
140 PRINT "IMPIEGHERAI ";T1;"ANNI"
150 PRINT "E ARRIVERAI FRA ";T2;" ANNI."
170 IF T1>50 THEN GOTO 210

```

Calcola il tempo trascorso all'esterno della navicella (cioè sulla Terra).

Controlla se ci abbiate messo di più degli anni che vi rimangono da vivere (50 anni). In tale caso, salta alla linea 210.

```

180 IF ABS(T-T2)<=5 THEN PRINT "ARRIVERAI IN TEMPO"
190 IF ABS(T-T2)>5 THEN PRINT "TROPPO TARDI"
200 STOP
210 PRINT "SEI MORTO DURANTE IL VIAGGIO"
220 STOP

```

Controlla se siete arrivati entro 5 anni dal termine richiesto, e stampa un messaggio.

Il listato visto funziona sul VIC 20 e sul C 64. Per altri computers, effettuate le modifiche riportate sotto.

```

● 10 HOME
s ⊠▲ 10 CLS
■ 30 LET T=INT(RND(0)*100+25)
s ▲ 30 LET T=INT(RND*100+25)

```

### Angolo del quesito

Modificate il programma per fare quanto segue:

- 1) Fornire un intervallo di valori maggiore per gli anni che trascorreranno prima del vostro ritorno sulla Terra.
- 2) Aumentare la precisione richiesta per il momento del ritorno, da un errore di 5 anni ad uno di 2 anni.
- 3) Aumentare o diminuire il numero di anni che vi restano da vivere.



# La Valle della Morte

C'è un solo modo per sfuggire alle armate dei malvagi Dissectitrons: dovrete pilotare il vostro veicolo monoposto Dardo Veloce lungo la gola frastagliata e senza fondo, chiamata Valle della Morte.

Il computer vi chiederà qual'è la larghezza della valle. Provate ad inserire 15(\*) all'inizio, e poi diminuite gradatamente. Guidate il vostro Dardo Veloce premendo Q per andare a sinistra e P a destra, e cercate di attraversare la Valle della Morte senza danni.

(\*) Se state usando un VIC 20, allora utilizzate i valori di larghezza da 6 a 10.

## Come funziona il programma

```

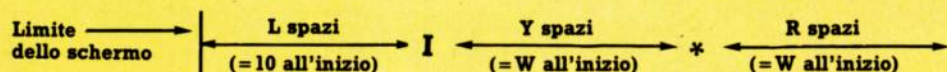
10 PRINT "LA VALLE DELLA MORTE"
20 LET S=0
30 LET M=200
40 PRINT "LARGHEZZA"
50 INPUT W
60 LET W=INT(W/2)
70 LET L=4
80 LET Y=W
90 LET R=W
    
```

Pone a zero il numero di turni impiegati, all'inizio del gioco.

M è il massimo numero di mosse ammesso.

Vi chiede di fornire un valore per la larghezza, lo divide per due e usa INT per eliminare l'eventuale frazione 1/2.

L, Y ed R sono le distanze tra le pareti e il Dardo Veloce.

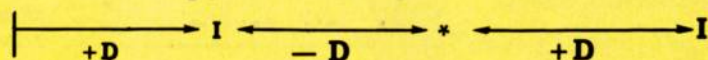


```

100 LET D=INT(RND(1)*3-1)
110 IF L+D<0 OR L+D>20 THEN GOTO 100
120 LET L=L+D
130 LET Y=Y-D
140 LET R=R+D
    
```

Sceglie un valore tra -1,0 e +1, e lo scrive in D. Controlla che L+D non sia così grande o così piccolo da far uscire le pareti dai lati dello schermo.

Modifica gli spazi tra le colonne in funzione del valore di D.



Muove il cursore di L spazi in orizzontale, e stampa I. (Il punto e virgola serve a evitare che il cursore passi alla linea successiva).

Muove il cursore per altri Y spazi, e stampa \*.

Muove il cursore per altri R spazi, e stampa un'altra I. (Non c'è il punto e virgola stavolta, per cui il cursore successivamente passa alla linea sottostante).

Controlla se state premendo un tasto.

Se il tasto che state premendo non è Q, il computer salta all'istruzione 280.

Se il tasto è Q, allora Y viene decrementato di 1, ed R viene aumentato di 1, cosicché l'asterisco si sposta a sinistra.

Controlla se il tasto premuto è P.

```

150 LET N=L
160 GOSUB 1000
170 PRINT "I";
180 LET N=Y
190 GOSUB 1000
200 PRINT "*";

210 LET N=R
220 GOSUB 1000
230 PRINT "I"

240 GET I$
250 IF I$<>"Q" THEN GOTO 280
260 LET Y=Y-1
270 LET R=R+1
280 IF I$<>"P" THEN GOTO 310
    
```

```

290 LET Y=Y+1
300 LET R=R-1
310 IF Y<1 OR R<1 THEN GOTO 370
320 LET S=S+1

```

In tal caso, l'asterisco viene mosso verso destra aumentando Y di 1, e diminuendo R di 1.

Controlla se vi siete scontrati con una parete. Salta all'istruzione 370, se ciò è successo, per comunicarvelo.

Incrementa di 1 il numero delle mosse utilizzate.

```

330 IF S<M THEN GOTO 100
340 PRINT "BRAVO! SEI PASSATO"
350 PRINT "ATTRAVERSO LA VALLE DELLA MORTE"
360 STOP
370 PRINT "TI SEI SCHIANTATO CONTRO"
380 PRINT "LA PARETE DISINTEGRANDOTI"
390 STOP

```

Torna all'inizio del ciclo per il turno successivo, se avete utilizzato meno di M mosse.

Stampa questo se avete effettuato M turni senza scontrarvi.

```

1000 IF N=0 THEN RETURN
1010 FOR I=1 TO N
1020 PRINT " ";
1030 NEXT I
1040 RETURN

```

Subroutine per spostare il cursore nei posti giusti per la stampa di I e \*.

Il listato visto funziona sul VIC 20 e sul C 64. Per altri computers, effettuate le modifiche riportate sotto.

```

S*■●▲70 LET L=10
  ■100 LET D=INT(RND(0)*3-1)
S▲100 LET D=INT(RND*3-1)
  ▲145 SCROLL
  ●235 I$=""
  ⊠240 LET I$=INKEY$(1)
  ●240 IF PEEK(-16384)>127 THEN GET I$
S▲240 LET I$=INKEY$

```

### Rallentare il gioco

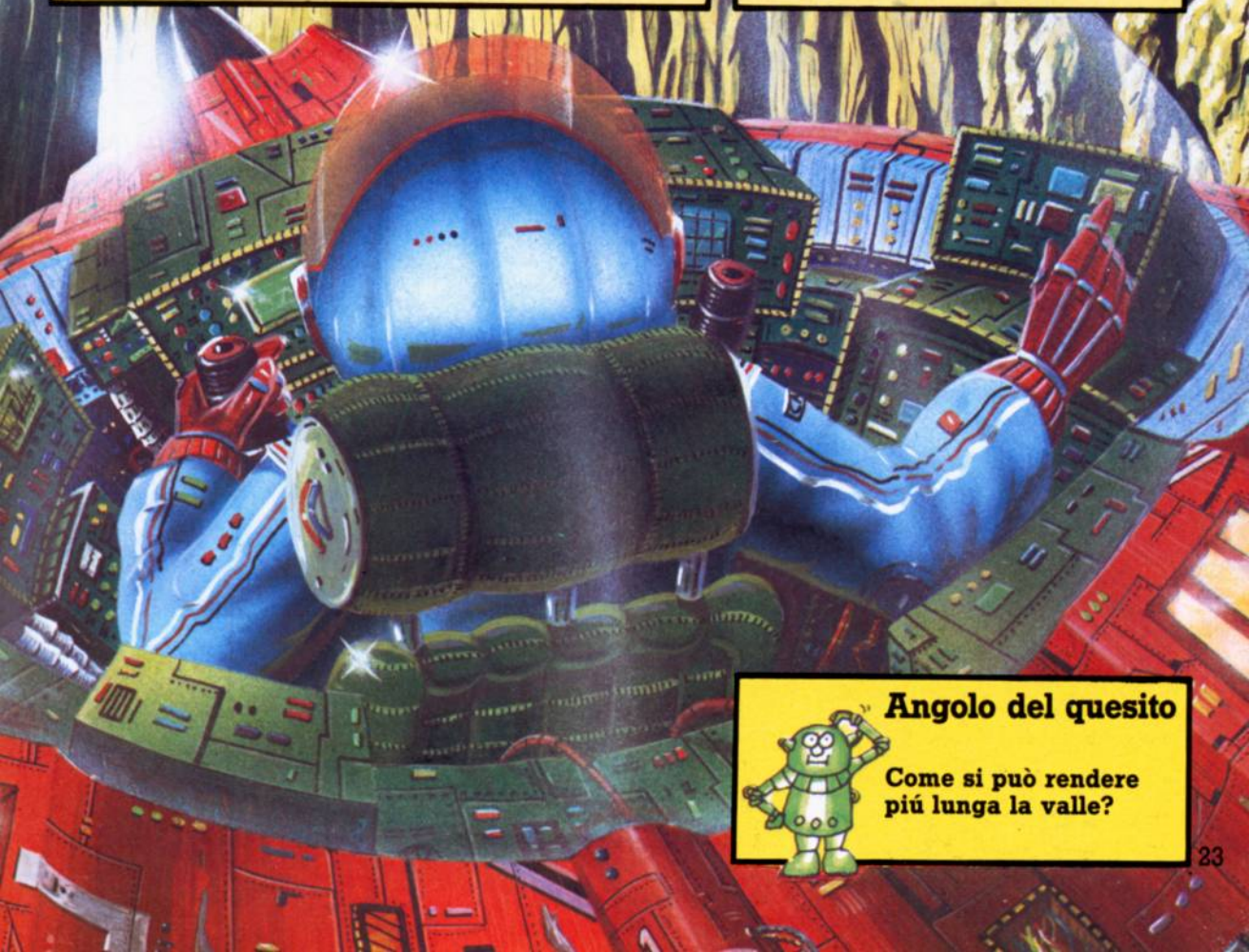
Se questo gioco è troppo veloce per voi, potrete aggiungere un ciclo di ritardo alle linee 141 e 142:

```

141 FOR J=1 TO 100
142 NEXT J

```

Cambiate il numero alla linea 141 per regolare la velocità — quanto minore è il numero, tanto più è veloce il gioco.



### Angolo del quesito



Come si può rendere più lunga la valle?



# Le miniere spaziali

Siete stati eletti capo di una colonia ricca di miniere, sul Pianeta Astron. Tutte le decisioni riguardo la vendita di minerali ai Commercianti Intergalattici, l'acquisto di viveri, e il commercio delle miniere, vengono prese da voi. Riuscirete a soddisfare il vostro popolo, e portare a termine i dieci anni di incarico, oppure la vita della colonia, sotto la vostra guida, terminerà in una catastrofe?

## Come funziona il programma

```

10 LET L=INT(RND(1)*3+5)
20 LET P=INT(RND(1)*60+40)
30 LET M=INT(RND(1)*50+10)*P
40 LET FP=INT(RND(1)*40+80)
50 LET CE=INT(RND(1)*40+80)

60 LET C=0
70 LET S=1
80 LET Y=1

90 LET LP=INT(RND(1)*2000+2000)
100 LET CP=INT(RND(1)*12+7)
110 PRINT CHR$(147)
120 PRINT "ANNO ";Y
130 PRINT
140 PRINT "CI SONO ";P;" PERSONE
    NELLA COLONIA"
150 PRINT "HAI ";L;" MINIERE E $"M
160 PRINT "FATTORE DI SODDISFAZIONE ";S
170 PRINT
180 PRINT "LE TUE MINIERE PRODUCONO
    CIASCUNA ";CE;" TONN."
190 LET C=C+CE*L
200 PRINT "MINERALI IMMAGAZZINATI=";
    C;" TONN."
210 PRINT "VENDITE"
220 PRINT "PREZZO DI VENDITA DEI
    MINERALI=";CP
230 PRINT "PREZZO DI VENDITA DELLE
    MINIERE=";LP;"/MINIERA"
240 PRINT "QUANTO MINERALE VUOI VENDERE"
250 INPUT CS
260 IF CS<0 OR CS>C THEN GOTO 240
270 LET C=C-CS
280 LET M=M+CS*CP
290 PRINT "QUANTE MINIERE VUOI VENDERE"
300 INPUT LS
310 IF LS<0 OR LS>L THEN GOTO 290
320 LET L=L-LS
330 LET M=M+LS*LP
340 PRINT
350 PRINT "HAI $"M
360 PRINT
370 PRINT "ACQUISTI"
380 PRINT "QUANTO VUOI SPENDERE PER IL
    CIBO?(CIRCA $100 PER PERSONA)"
390 INPUT FB
400 IF FB<0 OR FB>M THEN GOTO 380
410 LET M=M-FB
  
```

S ■ ▲ } Queste linee determinano il numero di miniere (L), il numero di persone (P), i fondi (M), il prezzo dei viveri (FP), e la quantità di minerali estratti da ogni miniera, all'inizio del gioco.

60 LET C=0 } Pone a zero la quantità di minerali immagazzinata all'inizio.

70 LET S=1 } Pone a 1 il fattore di soddisfazione.

80 LET Y=1 } Pone a 1 il numero di anni.

S ■ ▲ 90 LET LP=INT(RND(1)\*2000+2000) }

S ■ ▲ 100 LET CP=INT(RND(1)\*12+7) }

S ⊕ ■ ● ▲ 110 PRINT CHR\$(147) } Determina il prezzo di acquisto/vendita delle miniere.

120 PRINT "ANNO ";Y }

130 PRINT }

140 PRINT "CI SONO ";P;" PERSONE NELLA COLONIA" } Determina il prezzo di vendita dei minerali.

150 PRINT "HAI ";L;" MINIERE E \$"M }

160 PRINT "FATTORE DI SODDISFAZIONE ";S }

170 PRINT }

180 PRINT "LE TUE MINIERE PRODUCONO CIASCUNA ";CE;" TONN." } Stampa l'attuale stato di cose nella colonia.

190 LET C=C+CE\*L }

200 PRINT "MINERALI IMMAGAZZINATI=";C;" TONN." }

210 PRINT "VENDITE" }

220 PRINT "PREZZO DI VENDITA DEI MINERALI=";CP } Vi chiede che quantità di minerali volete vendere, ne mette il valore in CS, e controlla che abbiate tale quantità in magazzino.

230 PRINT "PREZZO DI VENDITA DELLE MINIERE=";LP;"/MINIERA" }

240 PRINT "QUANTO MINERALE VUOI VENDERE" }

250 INPUT CS }

260 IF CS<0 OR CS>C THEN GOTO 240 }

270 LET C=C-CS } Sottrae l'ammontare venduto da quello immagazzinato.

280 LET M=M+CS\*CP } Calcola il valore dei minerali venduti, e aggiunge tale valore al denaro in vostro possesso.

290 PRINT "QUANTE MINIERE VUOI VENDERE" }

300 INPUT LS } Ripete la stessa operazione per la vendita delle miniere.

310 IF LS<0 OR LS>L THEN GOTO 290 }

320 LET L=L-LS }

330 LET M=M+LS\*LP }

340 PRINT }

350 PRINT "HAI \$"M } Stampa la nuova quantità di denaro in vostro possesso.

360 PRINT }

370 PRINT "ACQUISTI" }

380 PRINT "QUANTO VUOI SPENDERE PER IL CIBO?(CIRCA \$100 PER PERSONA)" } Vi chiede quanto volete spendere in ci-

390 INPUT FB }

400 IF FB<0 OR FB>M THEN GOTO 380 } barie, e pone questo valore in FB.

410 LET M=M-FB } Controlla che abbiate denaro a sufficienza per pagare ciò che avete acquistato.

} Aggiorna la vostra scorta di denaro.

```

420 IF FB/P>120 THEN LET S=S+.1
430 IF FB/P<80 THEN LET S=S-.2
440 PRINT "QUANTE MINIERE VUOI COMPRARE"
450 INPUT LB
460 IF LB<0 OR LB*LP>M THEN GOTO 440
470 LET L=L+LB
480 LET M=M-LB*LP
490 IF S<.6 THEN GOTO 660

```

Aggiorna il fattore di soddisfazione in funzione di quanto avete speso per le cibarie.

Vi chiede quante miniere volete comprare, e controlla che abbiate i soldi per farlo.

Incrementa, se necessario, il numero delle miniere.

Aggiorna di nuovo la vostra scorta di denaro.

Controlla il valore del fattore di soddisfazione. Se questo è molto basso, il computer salta all'istruzione 660 per terminare il gioco.

S ■▲ 500 IF S>1.1 THEN LET CE=CE+INT(RND(1)\*20+1)

S ■▲ 510 IF S<.9 THEN LET CE=CE-INT(RND(1)\*20+1)

520 IF P/L<10 THEN GOTO 680

S ■▲ 530 IF S>1.1 THEN LET P=P+INT(RND(1)\*10+1)

S ■▲ 540 IF S<.9 THEN LET P=P-INT(RND(1)\*10+1)

Se S è alto, allora viene aumentata la quantità di minerali estratti da ciascuna miniera.

Se S è basso, la quantità estratta viene diminuita.

Se vi sono meno di 10 persone per ciascuna miniera, il gioco termina.

Se S è alto, arrivano altre persone.

Se S è basso, le persone se ne vanno.

Se vi sono meno di trenta persone, il gioco ha termine.

550 IF P<30 THEN GOTO 700

S ■▲ 560 IF RND(1)>.01 THEN GOTO 590

570 PRINT "FUGA RADIOATTIVA.....MOLTE PERDITE"

580 LET P=INT(P/2)

590 IF CE<150 THEN GOTO 620

600 PRINT "ECCEDENZA DI MERCATO - I PREZZI CROLLANO"

610 LET CE=INT(CE/2)

620 LET Y=Y+1

630 IF Y<11 THEN GOTO 90

640 PRINT "SEI SOPRAVVISSUTO AL TERMINE DEL TUO COMPITO"

Introduce una minima possibilità che metà delle persone vengono uccise.

Se la quantità estratta da ciascuna miniera è molto alta, il prezzo dei minerali viene dimezzato.

Il numero di anni viene aumentato di 1, e se è inferiore a 11 il computer salta all'istruzione 90 per il turno successivo.

650 STOP

660 PRINT "RIVOLTA POPOLARE"

670 STOP

680 PRINT "HAI SOVRACCARICATO TUTTI DI LAVORO"

690 STOP

700 PRINT "NON CI SONO PERSONE A SUFFICIENZA"

710 STOP

Stampa questo se il computer arriva a questo punto del programma, nel corso del vostro decimo tentativo.

Il listato visto funziona sul VIC 20 e sul C 64. Per altri computers, effettuate le modifiche riportate sotto.

● 110 HOME

S ■▲ 110 CLS

■ 10, 20, 30, 40, 50, 90, 100, 500, 510, 530, 540, 560  
CAMBIARE RND(1) CON RND(0)

S ▲ 10, 20, 30, 40, 50, 90, 100, 500, 510, 530, 540, 560  
CAMBIARE RND(1) CON RND



### Angolo dei quesiti

Sapreste fare in modo che il computer vi chieda se volete giocare ancora, aggiungendo la somma di denaro rimastavi alla nuova scorta di denaro fornitavi per il gioco successivo?

## Soccorso spaziale

Dovete fare un viaggio urgentemente, attraverso il braccio a spirale della Galassia, fino ad un pianeta in via di sviluppo, il quale ha bisogno di forniture mediche. Il viaggio prevede una distanza talmente grande, che per la maggior parte di esso voi verrete immersi in un sonno profondo, ma prima dovrete programmare la navicella per il viaggio. Il computer vi chiederà quanta energia volete ripartire ai motori, ai sistemi di supporto vitale e alle protezioni esterne, poi vi farà addormentare.

Al vostro risveglio, esso vi fornirà un rapporto sui fatti accaduti durante il viaggio e, se tutto è andato bene, vi troverete nell'orbita del pianeta. Ora dovrete ripartire l'energia che vi rimane ai razzi d'atterraggio e agli scudi termici, allo scopo di effettuare un buon atterraggio sul pianeta.

Se portate felicemente a termine la missione, avrete una buona probabilità di venire promossi al grado di Ammiraglio Spaziale. Buona fortuna!

```
S*■●▲10 PRINT CHR$(147)
  20 PRINT"SOCCORSO SPAZIALE"
  30 PRINT
  40 PRINT "VUOI LE ISTRUZIONI"
  50 INPUT I$
  ▲S60 IF LEFT$(I$,1)="S" THEN GOSUB 1000
  ■▲S70 LET D=INT(RND(1)*800+101)
  ■▲S80 LET E=INT(RND(1)*400+401)
  90 LET T=INT(D/SQR(E/5)+.5)
  100 PRINT "IL PIANETA E' LONTANO ";D;"UNITA'"
  110 PRINT "HAI ";E/" UNITA' DI ENERGIA"
  120 PRINT "E UN TEMPO LIMITE DI ";T;" GIORNI"
  130 PRINT
  140 PRINT "RIPARTIZIONE DELL'ENERGIA:"
  150 PRINT "AI MOTORI"
  160 INPUT P
  170 PRINT "AI SISTEMI DI SUPPORTO VITALE"
  180 INPUT L
  190 PRINT "ALLE PROTEZIONI ESTERNE"
  200 INPUT S
  210 IF P+L+S>E THEN GOTO 140
  220 LET X=E-P-L-S
  230 LET V=INT(SQR(P))
  240 LET T1=INT(D/V)
S*■●▲250 PRINT CHR$(147)
  260 PRINT "LA TUA VELOCITA' E' ";V
  270 PRINT "HAI UN PERIODO DI SONNO ";T1;" GIORNI"
  280 PRINT
  S■▲290 FOR I=1 TO INT(RND(1)*5+6)
  S■▲300 IF RND(1)>.5 THEN GOTO 430
  ▲S310 ON INT(RND(1)*4+1) GOTO 320,350,380,410
  320 PRINT "SCIAME DI ASTEROIDI - PROTEZIONI DANNEGGIATE"
```

```

S■▲330 LET S=S-20-INT(RND(1)*40+1)
340 GOTO 430
350 PRINT "GUASTO DEL COMPUTER - RITARDO PER LE RIPARAZIONI"
S■▲360 LET D=D+INT(RND(1)*20+1)
370 GOTO 430
380 PRINT "GUASTO AI MOTORI - DEVI RALLENTARE"
390 LET V=V-.5
400 GOTO 430
410 PRINT "RAGGI X - DANNEGGIATO IL SUPPORTO VITALE"
420 LET L=L-20-INT(RND(1)*40+1)
S■▲430 FOR J=1 TO 500
440 NEXT J
450 NEXT I
460 LET T1=INT(D/V)
S■●▲470 PRINT CHR$(147)
480 PRINT "SEI ARRIVATO IN ";T1;" GIORNI"
490 IF S<0 THEN PRINT "PROTEZIONI DISTRUTTE"
495 IF S<0 THEN PRINT "SEI STATO RISUCCHIATO"
500 IF L<=0 THEN PRINT "SUPPORTO VITALE INATTIVO"
505 IF L<=0 THEN PRINT "SEI MORTO"
510 IF V<=0 THEN PRINT "I MOTORI NON FUNZIONANO"
520 IF T1>T THEN PRINT "TI CI VUOLE TROPPO TEMPO"
530 IF S<0 OR L<=0 OR V<=0 OR T1>T THEN STOP
S■▲540 LET G=INT(RND(1)*10+5)
550 LET G$="ALTA"
560 IF G<12 THEN LET G$="MEDIA"
570 IF G<8 THEN LET G$="BASSA"
S■▲580 LET A=INT(RND(1)*10+5)
590 LET A$="ALTA"
600 IF A<12 THEN LET A$="MEDIA"
610 IF A<8 THEN LET A$="BASSA"
620 PRINT
630 PRINT "ORA SEI IN ORBITA ATTORNO AL PIANETA"
640 PRINT "ENERGIA IN PIU'=";X
650 PRINT "LA GRAVITA' E' ";G$
660 PRINT "L'ATMOSFERA E' ";A$
670 PRINT
680 PRINT "QUANTA ENERGIA AI RAZZI DI ATTERraggio"
690 INPUT B
700 PRINT "QUANTA ENERGIA AGLI SCUDI TERMICI"
710 INPUT S
720 IF B+S>X THEN GOTO 680
S■●▲730 PRINT CHR$(147)
740 IF B>=G*10 THEN GOTO 770

```

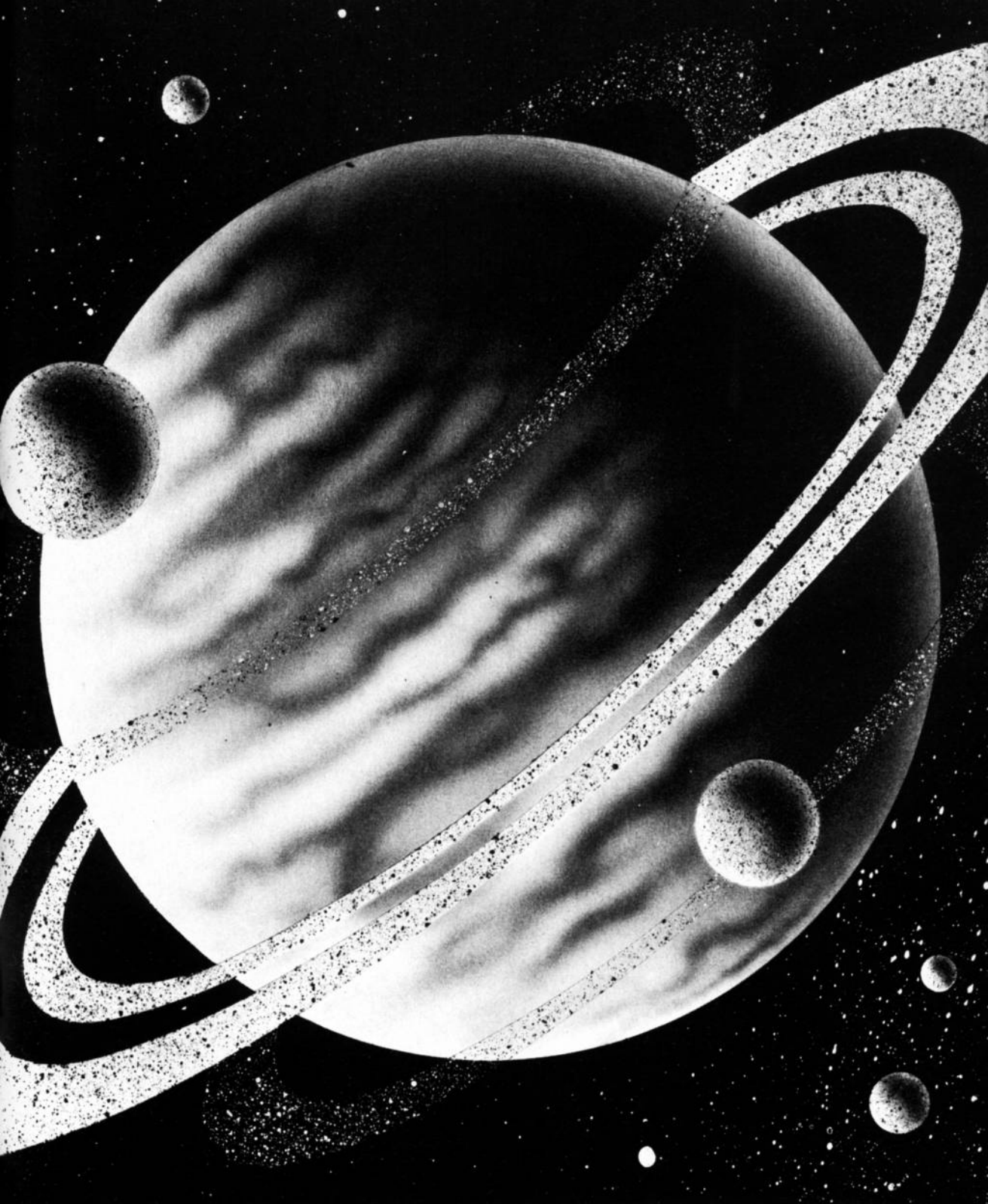
*Il programma continua nella prossima pagina.*

## Soccorso Spaziale (seguito)

```
750 PRINT "HAI APERTO UN NUOVO CRATERE"  
760 GOTO 840  
770 IF S>=A*10 THEN GOTO 800  
780 PRINT "SEI DIVENTATO UNA MERAVIGLIOSA STELLA CADENTE"  
790 GOTO 840  
800 PRINT "SEI ATTERRATO PERFETTAMENTE - COMPLIMENTI"  
810 IF X-S-B>25 THEN GOTO 840  
820 PRINT "MI SPIACE MA NON HAI ABBASTANZA"  
830 PRINT "ENERGIA PER APRIRE LA PORTA"  
840 STOP  
1000 PRINT  
1010 PRINT "STAI PER IMBARCARTI PER UNA MISSIONE IN UN PIANETA  
DISTANTE"  
  
1020 PRINT "CHE HA URGENTE BISOGNO DI FORNITURE MEDICHE"  
1030 PRINT "DEVI PRIMA PROGRAMMARE L'ASTRONAVE"  
1040 PRINT "PER IL VIAGGIO RIPARTENDO PARTE DELL'ENERGIA AI"  
1050 PRINT "MOTORI, AI SISTEMI DI SUPPORTO VITALE E ALLE PROTEZIONI  
ESTERNE."  
  
1060 PRINT "DOVRAI DORMIRE PER LA MAGGIOR PARTE DEL VIAGGIO."  
1070 PRINT "IN SEGUITO AVRAI UN RAPPORTO SU CIO' CHE E' ACCADUTO"  
1080 PRINT "POTRAI COSI' ATTERRARE SUL PIANETA....."  
1090 PRINT "PREMI UN TASTO"  
S●▲1150 GET I$:IF I$="" THEN GOTO 1150  
1160 PRINT CHR$(147)  
1170 RETURN
```

**Il listato visto funziona sul VIC 20 e sul C 64. Per altri computers, effettuate le modifiche riportate sotto.**

```
■ CAMBIARE TUTTI I RND(1) CON RND(0)  
S▲ CAMBIARE TUTTI I RND(1) CON RND  
● 10,250,470,730,1160 HOME  
S●▲ 10,250,470,730,1160 CLS  
S▲ 60 IF I$(1)="S" THEN GOSUB 1000  
S▲ 310 GOTO 320+INT(RND*4)*30  
■ 430 FOR J=1 TO 1000  
S▲ 430 FOR J=1 TO 50  
■ 1150 I=GET  
● 1150 GET I$  
S▲ 1150 IF INKEY$="" THEN GOTO 1150
```



### **Aggiunte al gioco**

**Questo gioco è composto da due parti: nella prima voi partite per il viaggio con lo scopo di raggiungere l'orbita del pianeta; nella seconda parte tentate l'atterraggio sul pianeta. Potreste cercare di aggiungere una terza parte, nella quale effettuate la traversata dal punto di atterraggio fino al Quartier Generale della Croce Rossa Intergalattica.**

# Allunaggio

Questo gioco è diverso dagli altri di questo libro, perché fa uso della grafica. Poiché i diversi computers si differenziano molto per come viene gestita la grafica, il programma è presentato separatamente per ciascuno di essi. Leggete in questa pagina le istruzioni per il gioco, poi cercate nelle pagine seguenti la versione per il vostro computer.

## Come giocare ad Allunaggio

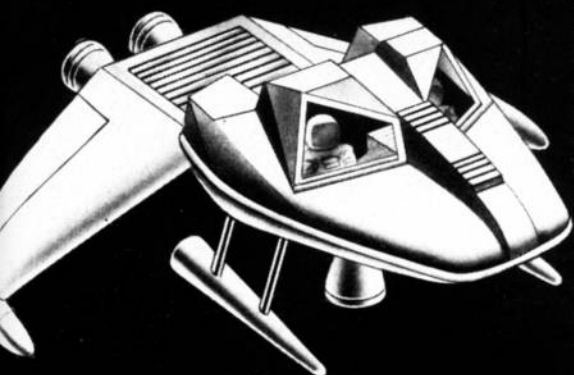
Capitan Saetta, asso fra i piloti, è seduto al vostro fianco mentre sostenete la Parte Terza del vostro Test Avanzato di Guida di un'Astronave. Il vostro veicolo d'atterraggio a due posti si sta rapidamente avvicinando alla superficie della Luna. La velocità dovrà essere, all'impatto, quasi uguale a zero. Controllate la spinta, premendo A per aumentarla e D per diminuirla, (\*) potendo vedere costantemente sullo schermo il vostro andamento. Se aumentate troppo la spinta, inizierete a risalire, se ne date troppo poca, farete un nuovo cratere sulla Luna. Riuscirete a far colpo su Capitan Saetta, con la vostra abilità?

(\*) Per il VIC 20 usate il tasto che muove in basso il cursore per aumentare la spinta, e il tasto che lo muove a destra per diminuirla.

Nel listato per VIC 20: Q rappresenta il tasto che muove in basso il cursore, J rappresenta il tasto che lo muove a destra.

## Allunaggio: versione TRS-80

```
20 CLS
30 CLEAR 200
31 B$=STRING$(25,131)
33 M1$=CHR$(194)+STRING$(2,176)
34 M2$=" "+STRING$(4,191)
35 M3$=CHR$(131)+CHR$(135)+
STRING$(2,131)+CHR$(139)+CHR$(131)
40 GOSUB 250
50 GOSUB 300
60 C=1:GOSUB 390
70 A=1:B=F:GOSUB 460
80 A=2:B=ABS(V):GOSUB 460
90 A=3:B=H:GOSUB 460
100 A=4:B=T:GOSUB 460
110 GOSUB 530
120 V1=V-T/20+G:F=F-T/10
130 H1=H-(V+V1)/10
140 C=0:GOSUB 390
150 IF H1<0 THEN 200
160 H=H1:V=V1
170 IF H<100 THEN 60
180 GOSUB 590
190 GOTO 220
200 H=0:C=1:GOSUB 390
210 GOSUB 660
220 END
250 H=100:F=100:T=0
260 V=INT(RND(0)*10+6)
270 G=INT(RND(0)*40+41)/100
280 RETURN
300 FOR X=80 TO 127
320 SET (X,47-INT(RND(0)*5))
330 NEXT
340 PRINT "GRAVITA'=";G
30 350 PRINT @192,"CARBUR.:"
355 PRINT @384,"VELOC.:"
360 PRINT @576,"ALTEZZA:"
365 PRINT @768,"SPINTA:"
370 RETURN
390 Y=818-64*INT(H/8)
400 PRINT @Y,;:IF C=1 THEN PRINT M1$;
ELSE PRINT CHR$(196);
410 PRINT @Y+64,;:IF C=1 THEN PRINT M2$;
ELSE PRINT CHR$(198);
420 PRINT @Y+128,;:IF C=1 THEN PRINT M3$;
ELSE PRINT CHR$(198)
440 RETURN
460 Y=(A*3+1)*64
470 PRINT @Y,CHR$(217);
480 PRINT @Y,LEFT$(B$,B/4);
510 RETURN
530 I$=INKEY$
540 IF I$="A" THEN T=T+4;
IF T>100 THEN T=100
550 IF I$="D" THEN T=T-4;
IF T<0 THEN T=0
560 IF T>F THEN T=F
570 RETURN
590 CLS
600 FOR I=1 TO 20
610 PRINT @INT(RND(0)*1024), "*"
620 NEXT
630 PRINT @470,"TI SEI PERDUTO
NELLO SPAZIO!"
640 RETURN
650 CLS
660 PRINT "ATTERRATO CON VEL. ";
INT((V+V1)*5)/10
670 IF (V+V1)<8 THEN PRINT "SIETE SALVI"
ELSE PRINT "SIETE MORTI TUTTI"
680 RETURN
```



## Allunaggio: versione VIC 20

```
20 PRINT CHR$(147)CHR$(5):
25 POKE 36879,8
30 DEF FNR(X)=INT(RND(1)*X+1)
40 GOSUB 250
50 GOSUB 300
60 C=1:GOSUB 390
70 A=1:B=F:GOSUB 460
80 A=2:B=ABS(V):GOSUB 460
90 A=3:B=H:GOSUB 460
100 A=4:B=T:GOSUB 460
110 GOSUB 530
120 V1=V-T/20+G:F=F-T/10
130 H1=H-(V+V1)/10
140 C=0:GOSUB 390
150 IF H1<0 THEN 200
160 H=H1:V=V1
170 IF H<=100 THEN 60
180 GOSUB 590
190 GOTO 220
200 H=0:C=1:GOSUB 390
210 GOSUB 650
220 END
250 H=100:F=100:T=0
260 V=5+FNR(10)
270 G=(FNR(40)+40)/100
280 RETURN
300 FOR X=8178 TO 8185
320 POKE X,98+2*FNR(3)
330 NEXT
340 PRINT "GRAVITA'=";G
350 PRINT "CARBUR.:"
355 PRINT "VELOC.:"
360 PRINT "ALTEZZA:"
365 PRINT "SPINTA:"
370 RETURN
390 Y=8137-22*INT(H/5)
400 IF C=0 THEN 425
405 POKE Y,108:POKE Y+1,123
410 POKE Y+22,160:POKEY+23,160
415 POKE Y+44,75:POKE Y+45,74
420 GOTO 440
425 FOR Z=0 TO 44 STEP 22
430 POKE Y+Z,32:POKE Y+Z+1,32
435 NEXT
440 RETURN
460 FOR X=0 TO 9
470 Y=A*88+X+7724
480 IF X<B/10 THEN POKE Y,102:GOTO 500
485 IF X<B/10+.5 THEN POKE Y,92:GOTO 500
490 POKE Y,32
500 NEXT
510 RETURN
530 GET I$
540 IF I$="Q" THEN T=T+4:IF T>100 THEN T=100
550 IF I$="J" THEN T=T-4:IF T<0 THEN T=0
560 IF T>F THEN T=F
570 RETURN
590 PRINT CHR$(147)
600 FOR I=1 TO 20
610 POKE 7679+FNR(506),42
620 NEXT
630 PRINT "TI SEI PERDUTO NELLO SPAZIO!"
640 RETURN
650 PRINT CHR$(147)"ATTERRATO"
660 PRINT "CON VEL. ";INT((V+V1)*5)/10
670 IF (V+V1)<8 THEN PRINT
"SEI SALVO!":RETURN
680 PRINT "SIETE MORTI TUTTI":RETURN
```

**Q** è il cursore che muove in basso

**J** è il cursore che muove a destra





## Allunaggio: versione ZX81

```
20 CLS
30 LET B$="[66666666666666666666]"
40 GOSUB 250
50 GOSUB 300
60 LET C=1
65 GOSUB 390
70 LET A=1
75 LET B=F
77 GOSUB 460
80 LET A=2
85 LET B=ABS(V)
87 GOSUB 460
90 LET A=3
95 LET B=H
97 GOSUB 460
100 LET A=4
105 LET B=T
107 GOSUB 460
110 GOSUB 530
120 LET V1=V-T/20+G
125 LET F=F-T/10
130 LET H1=H-(V+V1)/10
140 LET C=0
145 GOSUB 390
150 IF H1<0 THEN GOTO 200
160 LET H=H1
165 LET V=V1
170 IF H<=100 THEN GOTO 60
180 GOSUB 590
190 GOTO 220
200 LET H=0
205 LET C=1
207 GOSUB 390
210 GOSUB 660
220 STOP
250 LET H=100
252 LET F=100
255 LET T=0
260 LET V=5+INT(RND*10+1)
270 LET G=INT(RND*40+40)/100
280 RETURN
300 PRINT AT 21,20;"[SDADSADAS]"
340 PRINT AT 0,0;"GRAVITA'=";G
350 PRINT AT 3,0;"CARBUR.:"
355 PRINT AT 7,0;"VELOC.:"
360 PRINT AT 11,0;"ALTEZZA:"
365 PRINT AT 15,0;"SPINTA:"
370 RETURN
390 LET Y=(100-H)/6+1
400 IF C=0 THEN GOTO 425
405 PRINT AT Y,24;"[34]"
410 PRINT AT Y+1,23;"[Q W]"
415 PRINT AT Y+2,23;"[TGGY]"
420 GOTO 440
425 PRINT AT Y,24;" "
430 PRINT AT Y+1,23;" "
435 PRINT AT Y+2,23;" "
440 RETURN
460 LET Y=A*4
470 PRINT AT Y,0;"";REM 20 SPAZI
480 PRINT AT Y,0;B$( TO B/5)
```



```
510 RETURN
530 LET I$=INKEY$
540 IF I$="A" THEN LET T=T+4
545 IF T>100 THEN LET T=100
550 IF I$="D" THEN LET T=T-4
555 IF T<0 THEN LET T=0
560 IF T>F THEN LET T=F
570 RETURN
590 CLS
600 FOR I=1 TO 20
610 PRINT AT INT(RND*22),
INT(RND*32);"*"
620 NEXT I
630 PRINT AT 10,10;"TI SEI
PERDUTO NELLO SPAZIO"
640 RETURN
650 CLS
660 PRINT AT 20,0;"ATTERRATO
CON VEL. ";INT((V+V1)*5)/10
670 IF (V+V1)<8 THEN PRINT
"SIETE SALVI"
675 IF (V+V1)>=8 THEN PRINT
"SIETE MORTI TUTTI"
680 RETURN
```

### Nota speciale per chi usa lo ZX81

I caratteri grafici vengono mostrati nel listato con la seguente conversione. Trovando il carattere "[" occorre premere il tasto GRAPHICS, quindi inserire le lettere / numeri che seguono tenendo premuto SHIFT, fino a che non si ritrova il simbolo "]", allorché occorre premere di nuovo il tasto GRAPHICS e tornare all'inserimento normale.

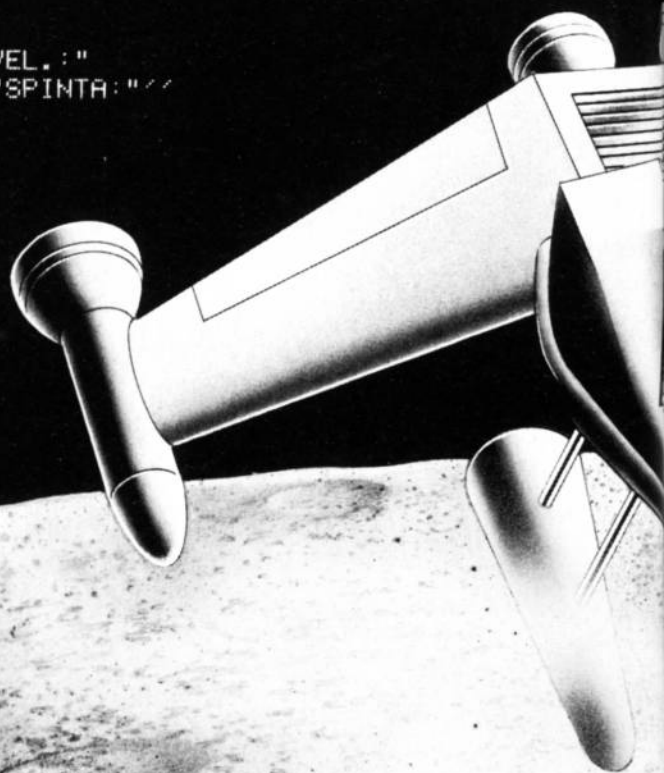
## Allunaggio: versione ZX Spectrum

```
20 CLS
30 DEF FNR(X)=INT(RND*X+1)
40 GOSUB 250
50 GOSUB 300
60 LET C=0:GOSUB 390
70 LET A=1:LET B=F:LET C=2*(F<25)
80 LET A=2:LET B=ABS V:LET C=4*(V<0)
85 GOSUB 460
100 LET A=4:LET B=T:LET C=0
105 GOSUB 460
110 GOSUB 530
120 LET V1=V-T/20+G:LET F=F-T/10
130 LET H1=H-(V+V1)/10
140 LET C=1:GOSUB 390
150 IF H1<0 THEN GOTO 200
160 LET H=H1:LET V=V1
170 IF H<=100 THEN GOTO 60
180 GOSUB 590
190 GOTO 220
200 LET H=0:LET C=0:GOSUB 390
210 GOSUB 650
220 STOP
250 LET H=100:LET F=100:LET T=0
260 LET V=5+FNR(10)
270 LET G=(FNR(40)+40)/100
280 RETURN
300 PLOT 180,8
310 FOR X=1 TO 15
320 DRAW 5,FNR(3)-2
330 NEXT X
340 PRINT "GRAVITA'=";G
350 PRINT ///"CARBUR.:"/////"VEL.:"
360 PRINT ///"ALTEZZA:"/////"SPINTA:"
370 RETURN
390 INVERSE C
400 LET Y=H*1.3+10
410 PLOT 200,Y:DRAW 34,0
420 DRAW -4,20:DRAW -13,10
430 DRAW -13,-10:DRAW -4,-20
440 RETURN
460 LET Y=172-A*32
470 INK C
480 PLOT 0,Y
490 DRAW B,0
500 DRAW INVERSE 1,100-B,0
510 RETURN
530 LET I$=INKEY$
540 IF I$="A" THEN LET T=T+4:
    IF T>100 THEN LET T=100
550 IF I$="D" THEN LET T=T-4:
    IF T<0 THEN LET T=0
560 IF T>F THEN LET T=F
570 RETURN
590 CLS
600 FOR I=1 TO 20
610 PRINT AT FNR(21),FNR(31);"*"
620 NEXT I
630 PRINT "TI SEI PERDUTO NELLO SPAZIO"
640 RETURN
650 PRINT AT 0,0;"ATTERRATO CON VEL.
    ";INT((V+V1)*5)/10
660 IF (V+V1)<8 THEN GOTO 680
670 PRINT "SIETE MORTI TUTTI":RETURN
680 PRINT "SIETE SALVI":RETURN
```



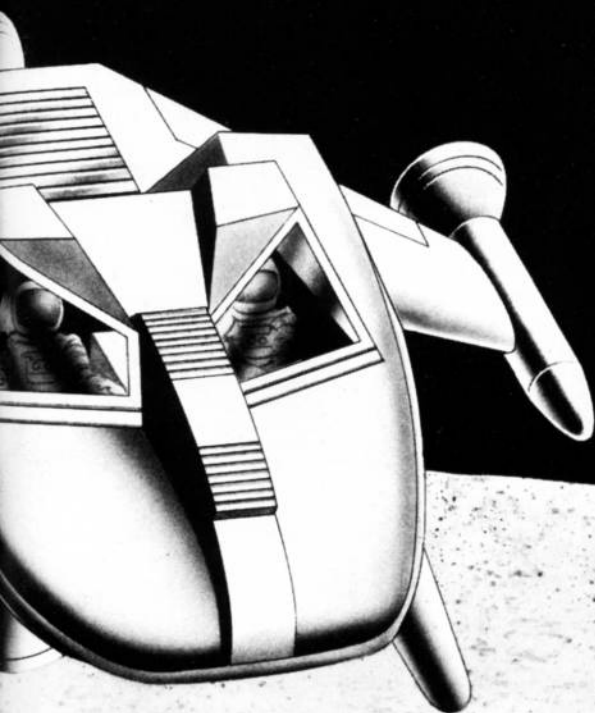
## Allunaggio: versione BBC

```
20 MODE 5
30 *FX 12,1
40 PROCSETVAR
50 PROCDISPLAY
60 PROCMODULE(H,3)
70 PROCBAR(1,F,3+2*(F<25))
80 PROCBAR(2,ABS(V,3+(V<0))
90 PROCBAR(3,H,3+2*(H<25))
100 PROCBAR(4,T,3)
110 PROCTHRUST
120 V1=V-T/20+G:F=F-T/10
130 H1=H-V*(V+V1)/10
140 PROCMODULE(H,0)
150 IF H1<0 THEN T=0
160 H=H1:V=V1
170 IF H<=100 THEN G0
180 PROCLOST
190 GOTO 220
200 PROCMODULE(0,2)
210 PROCLANDED
220 *FX 12
230 END
240 DEF PROCSETVAR
250 H=100:F=100:T=0
260 V=5+RND(10)
270 G=(RND(40)+40)/100
280 ENDPROC
290 DEF PROCDISPLAY
300 MOVE 800,30
310 FOR X=800 TO 1280 STEP 16
320 DRAW X,10+RND(40)
330 NEXT
340 PRINT "GRAVITA'=";G
350 PRINT "CARBUR.:";"VEL.:"
360 PRINT "ALTEZZA:";"SPINTA:"
370 ENDPROC
380 DEF PROCMODULE(H,C)
390 GCOL 0,C
400 Y=H*8.5+150
410 MOVE 1040,Y:PLOT 1,-40,-40
420 PLOT 1,-8,-60:PLOT 1,96,0
430 PLOT 1,-8,60:PLOT 1,-40,40
440 ENDPROC
450 DEF PROCBAR(N,V,C)
460 Y=1000-192*N
470 GCOL 0,C
480 MOVE 0,Y:MOVE 0,Y-16
490 PLOT 85,V*4,Y:PLOT 85,V*4,Y-16
500 PLOT 87,400,Y:PLOT 87,400,Y-16
510 ENDPROC
520 DEF PROCTHRUST
530 *FX 15,1
540 IF INKEY(-194) THEN T=T+4:
    IF T>100 THEN T=100
550 IF INKEY(-179) THEN T=T-4:
    IF T<0 THEN T=0
560 IF T>F THEN T=F
570 ENDPROC
580 DEF PROCLOST
590 CLS
600 FOR I=1 TO 20
610 VDU 31,RND(19),RND(31),42
620 NEXT
630 PRINT TAB(4,16)"TI SEI
    PERDUTO NELLO SPAZIO"
640 ENDPROC
650 DEF PROCLANDED
660 VDU 28,0,31,11,0,12
670 PRINT "ATTERRATO"
680 PRINT "CON VEL. ";INT((V+V1)*5)/10
690 IF (V+V1)<8 THEN PRINT "SIETE SALVI"
    ELSE PRINT "SIETE MORTI TUTTI"
700 ENDPROC
```



## Allunaggio: versione Apple

```
15 HOME
20 HGR
30 DEF FNR(X)=INT(RND(1)*X+1)
40 GOSUB 250
50 GOSUB 300
60 C=3:GOSUB390
70 A=1:B=F:GOSUB 460
80 A=2:B=ABS(V):GOSUB 460
90 A=3:B=H:GOSUB 460
100 A=4:B=T:GOSUB 460
110 GOSUB 530
120 V1=V-T/20+G:F=F-T/10
130 H1=H-V*(V+V1)/10
140 C=0:GOSUB 390
150 IF H1<0 THEN 200
160 H=H1:V=V1
170 IF C=100 THEN 60
180 GOSUB 590
190 GOTO 220
200 H=0:C=3:GOSUB 390
210 GOSUB660
220 END
250 H=100:F=100:T=0
260 V=5+FNR(10)
270 G=(FNR(40)+40)/100
280 RETURN
300 HCOLOR=3
305 HPLOT 0,155
310 FOR X=0 TO 279 STEP 5
320 HPLOT TO X,159-FNR(10)
330 NEXT
335 FOR I=1 TO 30:HPLOT FNR(279),FNR(150)
337 NEXT
340 VTAB 21:PRINT TAB(34);"G=";G
350 VTAB 21:PRINT "CARBUR.:";PRINT "VEL:"
360 PRINT "ALTEZZA:";PRINT "SPINTA:";
370 RETURN
390 HCOLOR=C
400 Y=(100-H)*1.3
410 HPLOT 140,Y TO 120,Y+10
420 HPLOT 120,Y+20:HPLOT TO 160,Y+20
430 HPLOT TO 160,Y+10:HPLOT TO 140,Y
435 HPLOT 155,Y+20 TO 160,Y+25
437 HPLOT 125,Y+20 TO 120,Y+25
440 RETURN
460 VTAB (20+A):HTAB 8
470 INVERSE
480 PRINT SPC(B/4);
490 NORMAL
500 PRINT SPC(26-B/4);
510 RETURN
530 I$="":IF PEEK(-16384)>127
    THEN GET I$
540 IF I$="A" THEN T=T+4:
    IF T>100 THEN T=100
550 IF I$="D" THEN T=T-4:
    IF T<0 THEN T=0
560 IF T>F THEN T=F
570 RETURN
590 HOME:VTAB 23
600 PRINT "TI SEI PERDUTO NELLO SPAZIO"
640 RETURN
660 HOME:VTAB 22
670 PRINT "ATTERRATO CON VEL. ";
    INT((V+V1)*5)/10
680 IF (V+V1)<8 THEN 700
690 PRINT "SIETE MORTI TUTTI":RETURN
700 PRINT "SIETE SALVI":RETURN
```



# Aggiunte ai programmi

Eccovi ora alcune idee per aggiunte che potreste fare ai programmi di questo libro o ai vostri stessi programmi. Nella maggior parte dei casi non sarete in grado di fare queste aggiunte sullo ZX81 con solo 1K di memoria, in quanto tutto tale spazio, mentre dovreste poter trovare tutto lo spazio che volete sugli altri computer.

Ricordate che dovrete limitare le vostre aggiunte ai numeri di linea disponibili nel programma, o rinumerare il programma. Se decidete di rinumerarlo, state attenti a modificare anche tutte le linee di GOTO e GOSUB.

## Farvi dire dal computer come giocare

A qualsiasi programma potete aggiungere una sezione per far stampare al computer le istruzioni su cosa dovete fare. Il modo più semplice è quello di aggiungere alcune linee, come quelle mostrate sotto, all'inizio del programma, e poi mettere la relativa subroutine in fondo.



```
10 PRINT "TITOLO DEL GIOCO"  
11 PRINT "VUOI SAPERE"  
12 PRINT "COME SI GIOCA?"  
15 INPUT I$  
s▲17 IF I$(1)="S" THEN GOSUB 1000  
VIC17 IF LEFT$(I$,1)="S" THEN GOSUB 1000
```

programma principale

```
1000 PRINT "QUELLO CHE DEVI "  
1010 PRINT "FARE E'....."  
1999 RETURN
```

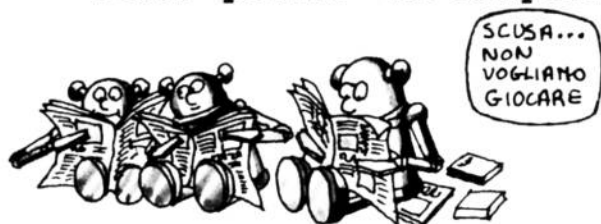
Per le istruzioni, potete aggiungere quante frasi di stampa volete, ricordatevi solo di mettere un numero e la parola PRINT all'inizio di ciascuna. Limitate la lunghezza della parte tra virgolette al numero di caratteri che il vostro computer può stampare su una linea di stampa. Non scordatevi di mettere una linea di RETURN alla fine o il programma non funzionerà.

## Far fermare e attendere il computer



Se le vostre istruzioni sono molto lunghe, potrebbe voler inserire questa subroutine, che arresta l'esecuzione del programma in un punto particolare, fintanto che non premete un tasto. In questo modo potrete evitare che le istruzioni scorrano verso l'alto sullo schermo, scomparendo prima che siate riusciti a leggerle. Ponete una linea di GOSUB nel punto dove volete che il programma si fermi, quindi mettete questa subroutine in fondo al programma.

## Farvi "parlare" dal computer



Potete farvi porre delle domande dal computer, e farlo agire in conseguenza delle vostre risposte. Ecco ad esempio un'aggiunta che causerà un rifiuto da parte del computer di giocare con voi, a meno che il vostro nome non inizi con la "J".

```
1000 PRINT "PREMI UN TASTO PER CONTINUARE"  
s▲1010 I$=GET$  
●1010 GET I$  
VIC1010 GET I$:IF I$="" THEN GOTO 1010  
1020 PRINT  
1030 RETURN
```

Segue il programma, quindi:

```
1000 PRINT "MI SPIACE, QUESTO GIOCO"  
1010 PRINT "E' SOLO PER CHI HA IL NOME"  
1020 PRINT "CHE INIZIA CON "  
1030 PRINT "LA LETTERA J"
```

Eccone ora un'altra con cui il computer vi sfida ad essere coraggiosi abbastanza da giocare.

```
10 PRINT "GIOCO VERAMENTE TERRIFICANTE"
12 PRINT "SEI COSI' CORAGGIOSO"
14 PRINT "DA AFFRONTARE"
15 PRINT "IL MOSTRO PELOSO?"
16 INPUT I$
S▲17 IF I$(1)="S" THEN GOTO 20
●●VIC17 IF LEFT$(I$,1)="S" THEN GOTO 20
18 PRINT "CODARDO!"
19 STOP
```

Potreste associare questa con la subroutine delle istruzioni, prendendo le linee 11-17 della sezione delle istruzioni mostrata prima, e inserendole come linee 20-26 in questo programma. Potrete poi far iniziare il programma principale alla linea 30 e aggiungere la subroutine delle istruzioni alla fine.

## Volete giocare ancora?

Invece di inserire RUN ogni volta che giocate, potrete farvi chiedere dal computer se volete giocare ancora. Inserite queste linee alla fine del programma, appena prima dell'ultimo comando di STOP.

```
1000 PRINT "VUOI FARE UN'ALTRA PARTITA?"
1010 INPUT I$
S▲1020 IF I$(1)="S" THEN RUN
●●VIC1020 IF LEFT$(I$,1)="S" THEN RUN
1030 PRINT "OK - CIAO!"
1040 STOP
```

Modificate i numeri di linee secondo il vostro programma.



## Aggiunta di effetti sonori

I modelli BBC, VIC 20, C 64, ZX Spectrum ed alcuni modelli Apple sono in grado di produrre suoni, e voi potete aggiungere linee ai vostri programmi per farglieli produrre in certi punti particolari. Potreste aggiungere un'esplosione, ad esempio, o un motivetto che venga suonato se vincete. Peraltro tutti i computer hanno bisogno di istruzioni diverse, per produrre suoni, per cui

dovrete esaminare il vostro manuale. In certi casi basterà aggiungere una sola linea al programma nel punto in cui volete generare il suono. In altri casi, vi potranno servire diverse linee, e vi converrà inserirle nel programma come subroutine.

Come esempio, ecco il suono di uno sparo per il modello BBC. Potrete fare delle prove per vedere dove vi conviene metterlo nel programma, ma dovrete comunque assegnargli un numero di linea per farlo funzionare:

SOUND 0, -15, 5, 10

In fondo al manuale per il VIC troverete alcune utili subroutines per produrre suoni quali "raggio laser", "esplosione" e "allarme rosso". Mettete una linea di GOSUB dove volete che si senta il suono, numerate la subroutine corrispondente, e aggiungetevi in fondo un comando di RETURN.



## Nota speciale per chi usa i modelli BBC e ZX Spectrum

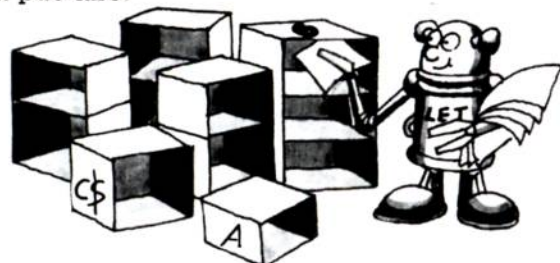


Se avete un modello BBC o uno ZX Spectrum, potrete trovare troppo veloce l'esecuzione di alcuni giochi di questo libro. Accanto a tali giochi troverete un riquadro con delle istruzioni per cambiarne la velocità. Per rallentare un gioco, vi servirà sempre usare un numero più alto. Le versioni più recenti del modello BBC possono eseguire programmi con velocità anche doppia rispetto alle versioni precedenti, e ciò potrebbe far apparire impossibili da giocare tali giochi, ad una prima esecuzione. Aspettatevi di dover apportare delle sensibili modifiche al numero che fissa la velocità.

# Scrivete i vostri programmi

Nel corso dell'esame dei programmi di questo libro, vi troverete probabilmente ad apportarvi sempre più modifiche, ed eventualmente vi verrà voglia di scrivere dei nuovi programmi da soli. Su queste pagine troverete alcuni spunti sui metodi da seguire.

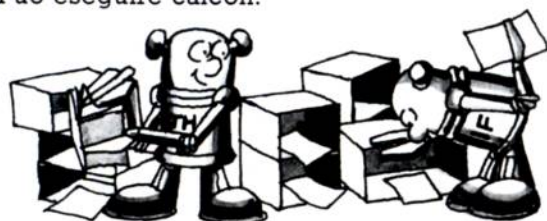
Prima di cominciare sarà bene pensare per un attimo a quello che il vostro computer può fare e a quello che non può fare.



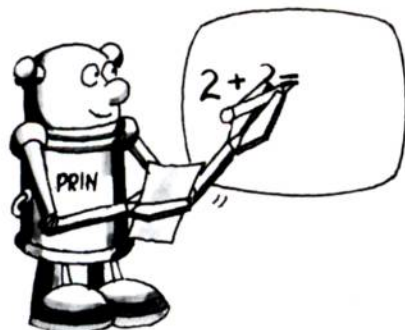
\* Può immagazzinare informazioni.



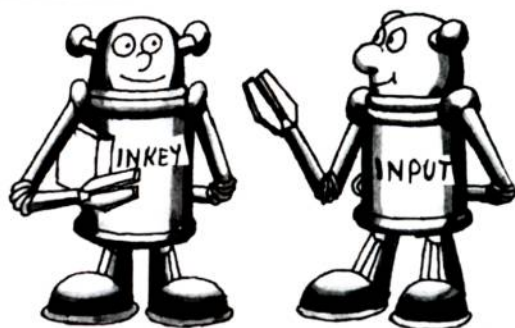
\* Può eseguire calcoli.



\* Può prendere decisioni confrontando elementi di informazioni in diversi modi.



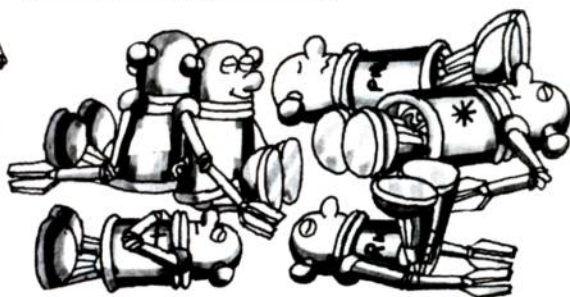
\* Vi può comunicare i risultati dei suoi calcoli e delle sue decisioni, e anche i contenuti della sua memoria.



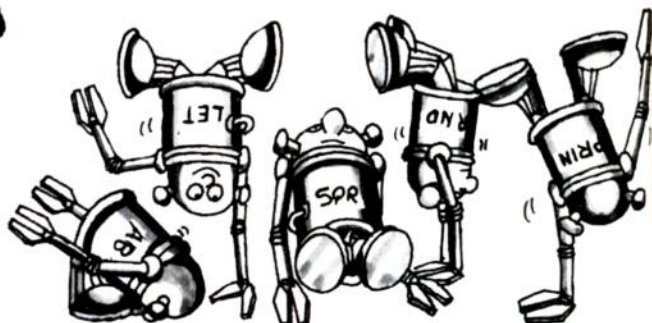
\* Vi può richiedere delle informazioni.



\* Può scegliere dei numeri a caso usando la funzione RND.



\* Non può far nulla a meno che non glielo diciate voi.



\* Purché ne usiate il linguaggio correttamente, può solo fare esattamente quello che gli dite, anche se è una sciocchezza.

Ricordatevi, nell'elaborare un gioco di non includervi nulla che il vostro computer non sia in grado di fare.

## Pianificare un gioco

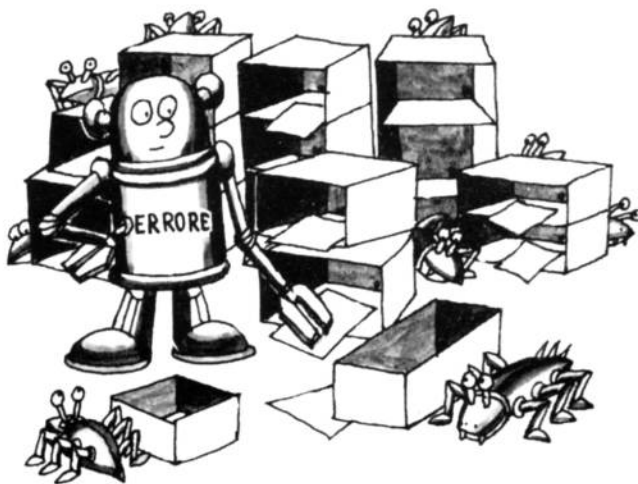
Prima di poter dire al computer come eseguire il vostro gioco, dovrete sapere voi stessi esattamente come giocarlo e quali sono le regole. Il computer avrà bisogno di una serie di semplici istruzioni logiche, perciò prima elaborate il gioco mentalmente o sulla carta, quindi suddividetelo in una serie di passi elementari. A questo punto, stendete un piano (in italiano — non mettetevi ad usare il BASIC, per ora) ordinato di tutte le varie fasi del gioco. Eccovi un piano per un semplice gioco che comporta una sparatoria, quale può essere il bombardamento di una nave pirata con palle di cannone, o lo sparo di raggi laser contro un invasore alieno, per darvi un'idea.

### PIANO

- 1) STAMPA DEL TITOLO E DELLE ISTRUZIONI
- 2) SCELTA DI UN OBIETTIVO PER QUESTO GIOCO
- 3) INIZIO DI UN GIOCO PER DARE AL GIOCATORE N TENTATIVI
- 4) RICHIESTA AL GIOCATORE DI SPARARE
- 5) CONTROLLO SE LO SPARO ABBIÀ RAGGIUNTO IL BERSAGLIO
- 6) STAMPA DI UN MESSAGGIO IN FUNZIONE DELLA PRECISIONE DELLO SPARO
- 7) RITORNO PER UN ALTRO TENTATIVO IN CASO DI SPARO FUORI BERSAGLIO

## Scrivere il programma

La fase successiva consiste nel convertire il piano in linguaggio BASIC. Ogni passo del vostro piano può richiedere parecchie linee in BASIC. Non scordate di lasciare degli intervalli tra numeri di linea successivi, così da potervi, all'occorrenza, inserire delle nuove linee. Prima scrivete un listato preliminare del programma, su carta, quindi iniziate le prove sul computer. Il vostro computer sarà in grado di rivelare errore molto più velocemente di quanto non li possiate trovare voi, e vi potrà fornire delle indicazioni su cosa c'è che non va. Ricordate che il debugging dei programmi è un'operazione lunga e noiosa anche per i programmatori esperti, quindi non aspettatevi di fare un programma perfetto al primo tentativo.

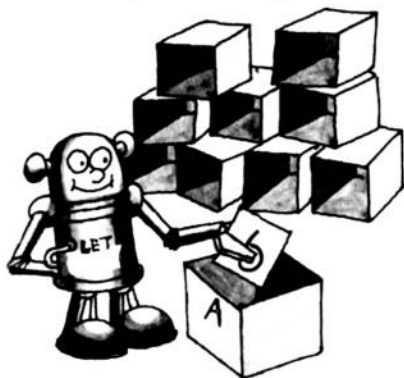


Una volta che abbiate ottenuto il nucleo del programma funzionante, potrete farvi delle aggiunte. I punti, ulteriori commenti, altri obiettivi sono tutte cose che possono essere incluse più avanti. Potreste aggiungere ai vostri giochi anche sezioni prese dai programmi di questo libro. Non pretendete di riuscire a scrivere giochi entusiasmanti e originali fin dall'inizio. Cercate di partire da idee molto semplici, e siate pronti a riadattarle strada facendo. Potreste accorgervi di aver incluso nel vostro gioco qualcosa di facile da fare per gli esseri umani, ma di molto difficile per un computer. Mano a mano che fate esperienza comincerete a capire istintivamente che cosa il vostro computer sia in grado di fare, e a trovare più facile lo scriverne i programmi.

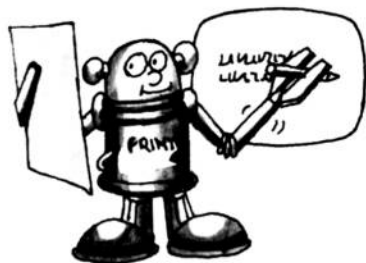


# Sommario di BASIC

In questa sezione vengono elencate comuni parole del BASIC, viene descritto ciò che fanno fare al computer e come vengono usate. Quasi tutte sono state usate nei programmi di questo libro, per cui potrete vedere come funzionano all'interno dei programmi di gioco. Non tutte le parole possono essere usate su tutti i computer citati. La tabella di conversione a pag. 46 mostra quelle che potete usare in alternativa.



**LET** dice al computer di attribuire un nome ad una sezione della sua memoria, e di mettervi dentro un particolare valore ad esempio `LET A=6` vuol dire: chiama "A" una sezione di memoria e metti dentro il valore 6. "A" viene detta una "variabile", e il mettervi dentro qualcosa viene detto "assegnare un valore ad una variabile".



Alcuni nomi di variabile vengono fatti seguire dal simbolo del dollaro, ad esempio `A$`. Questo vuol dire che servono per le "stringhe", che possono contenere un qualsiasi numero di caratteri, comprese le lettere, i numeri e i simboli.

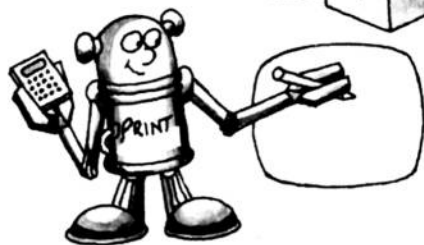


**PRINT** dice al computer di visualizzare qualcosa sullo schermo, e potete usare tale comando in molti modi diversi:

Un messaggio racchiuso tra virgolette preceduto da `PRINT`, verrà visualizzato sullo schermo esattamente come l'avete inserito. La parte tra virgolette non deve necessariamente essere scritta in BASIC, può essere qualsiasi cosa vogliate.

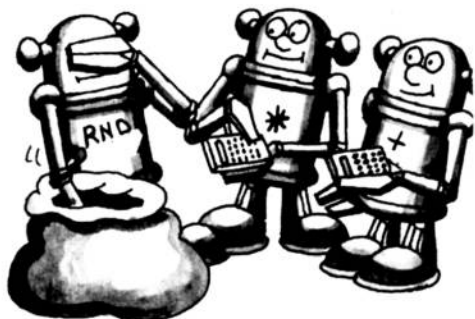
`PRINT` seguito da un nome di variabile, ad esempio `PRINT A` o `PRINT A$`, dice al computer di visualizzare sullo schermo il contenuto di tale variabile.

`PRINT` può anche effettuare dei calcoli e quindi visualizzare i risultati, ad esempio `PRINT 6*4` farà stampare 24 dal computer. Potete anche usare `PRINT` da solo per lasciare una riga vuota sullo schermo.



**RND** dice al computer di scegliere un numero a caso. Diversi computer usano diverse forme di `RND`, come potrete vedere sulla tabella di conversione a pag. 46. Sui computer Sinclair, `RND` da solo genera un numero compreso tra 0 e 0,99999999. Potete variare i limiti del numero scelto moltiplicando `RND` per un qualche numero, e aggiungendolo ad esso un qualche numero. Ad esempio `RND*20` genera un numero compreso tra 0 e 19,99999999, mentre `RND*20+1` genera un numero compreso tra 1 e 20,99999999.

Vedere il comando `INT` per sapere come si possono generare solo valori interi. Vedere `CHR$` per sapere come si possono generare lettere o altri caratteri della tastiera, a caso.



**INT** è l'abbreviazione di intero, che vuol dire numero privo di decimali. Per i numeri positivi, questo comando dice al computer di ignorare tutto quello che si trova a destra del punto decimale. Ad esempio INT (20.999) vale 20. Per i numeri negativi, viene ignorato tutto quello che si trova a destra del punto decimale, e viene incrementato di 1 il valore assoluto del numero che si trova a sinistra del punto, ad esempio INT (- 3.6) vale - 4.

INT viene spesso usato con RND, in questo modo: INT(RND × 20 + 1) che fa sapere al computer che gli volete far generare un numero intero compreso tra 1 e 20.



**CHR\$** converte i numeri in lettere. Con l'eccezione dello ZX81, tutti i computers menzionati in questo libro fanno uso della serie di caratteri da tastiera ASCII (\*), nella quale ciascun carattere corrisponde ad un certo numero. Ad esempio, la lettera A ha il numero di codice 65, e il comando PRINT CHR\$(65) visualizzerà sullo schermo una A. Potete usare CHR\$ con INT e RND, per far scegliere lettere a caso dal computer, in questo modo:

CHR\$(INT(RND(× 26 + 65)))

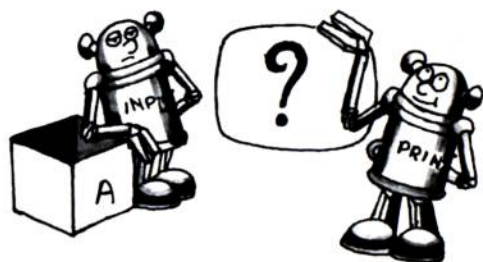
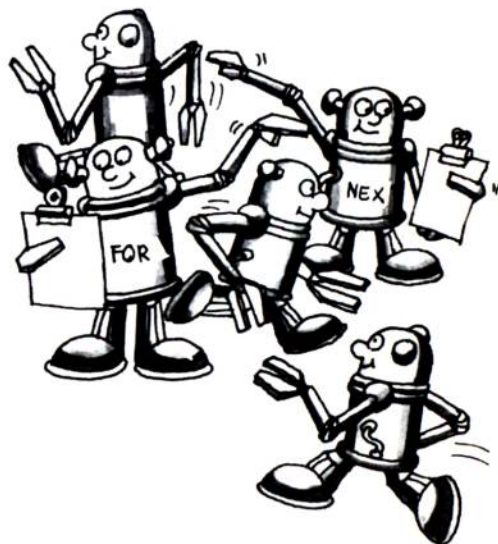
Questa linea produrrà delle lettere a caso sullo ZX Spectrum (per gli altri computer, vedere la tabella di conversione).

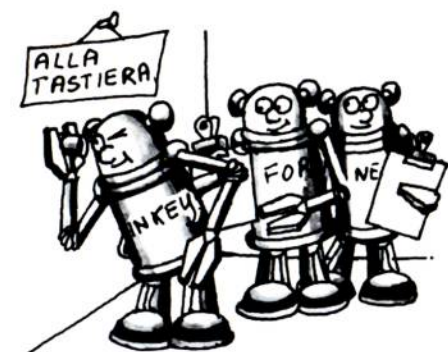
**FOR** viene utilizzato per iniziare un "ciclo", che farà ripetere al computer una parte di programma per un certo numero di volte. Deve essere seguito dal nome di una variabile, (quale ad esempio G per indicare il numero di turni (goes) permessi nel gioco), e alla variabile devono essere attribuiti dei valori iniziali e finali (ad esempio con il comando 1 TO 10).

La fine del ciclo è contrassegnata da una linea di NEXT, (NEXT G, nell'esempio), che ogni volta incrementa di 1 il valore della variabile, quindi rimanda di nuovo il computer all'esecuzione della linea di FOR. Quando la variabile ha raggiunto il suo valore finale, il computer ignora la linea di NEXT, e continua l'esecuzione dalla linea che la segue. Ad ogni FOR deve corrispondere un NEXT, o avrete un errore nel programma.

**INPUT** attribuisce un nome ad uno spazio di memoria del computer, stampa un punto di domanda, e resta in attesa che voi inseriate qualcosa da poter mettere in questo spazio di memoria. L'esecuzione del resto del programma non proseguirà finché non avrete poi premuto RETURN, ENTER, o NEWLINE.

Con il comando INPUT potete usare variabili numeriche o di tipo stringa, ma se usate una variabile numerica il computer non accetterà che gli dicitate di scriverci delle lettere.

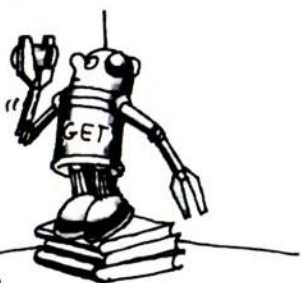




**INKEY\$** controlla la tastiera per vedere se vi è qualche tasto premuto, e in tal caso, quale tasto sia. Non aspetta che voi premiate un tasto, come succede con il comando INPUT. Viene di solito usato all'interno di un ciclo che fa controllare al computer la tastiera, più volte iterativamente. Questo perché i computer lavorano così velocemente che non riuscireste a premere un tasto nel tempo che il computer impiega ad effettuare un singolo controllo.

Se non avete premuto alcun tasto prima della fine del ciclo, il computer continua l'esecuzione, con una stringa vuota (chiamata stringa "nulla").

NB: I modelli Apple e VIC non usano il comando INKEY\$.



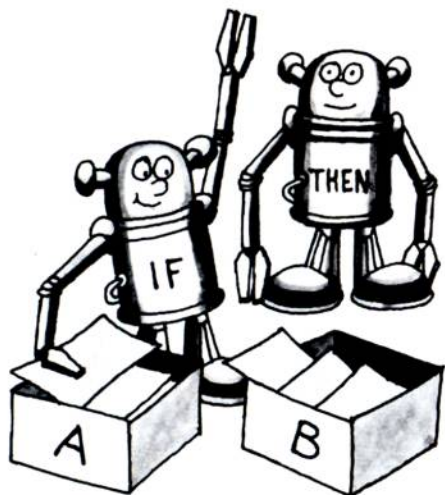
**GET** viene usato al posto di INKEY\$ sui computer VIC e Pet.



**GOTO** fa in modo che l'esecuzione "salti" su o giù nel programma ignorando le linee diverse da quella di destinazione. Dovete mettere il numero di linea a cui volete far saltare l'esecuzione, dopo l'istruzione GOTO.



**GOSUB** dice al computer di lasciare il programma principale e di andare ad eseguire una subroutine. GOSUB deve essere seguito dal numero della prima linea della subroutine. Alla fine della subroutine dovete avere una linea di RETURN. Questo comando fa tornare il computer al programma principale, alla linea immediatamente seguente quella di GOSUB. Un comando di GOSUB senza un RETURN nel programma darà un errore.

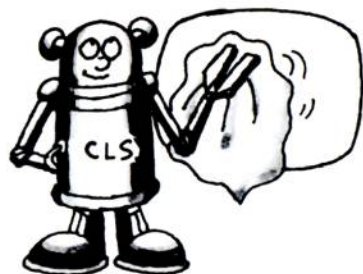


**IF ... THEN** dice al computer di decidere se un'espressione è vera o falsa e di comportarsi in modo diverso, in funzione della decisione presa. Viene usato con i seguenti simboli, oltre che con gli operatori AND e OR:

- = uguale a
- < minore di
- > maggiore di
- <= minore o uguale a
- >= maggiore o uguale a
- <> diverso da

Se il computer stabilisce che un'espressione è vera, prosegue con l'esecuzione dell'istruzione che segue la parola THEN. Se decide che è falsa, ignora il resto della linea e passa ad eseguire quella successiva.

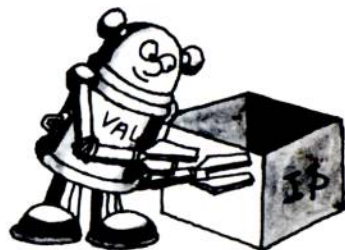
**CLS** viene usato per cancellare dallo schermo qualsiasi cosa vi si trovi, senza togliere od alterare nulla nei contenuti della memoria. È un comando utile per cancellare il listato dallo schermo all'inizio di un RUN, o nei giochi dove si vuole che il giocatore reagisca a qualcosa visto sullo schermo per un intervallo di tempo limitato. (NB: I modelli Apple e VIC non fanno uso del comando CLS-vedere la tabella di conversione).



**HOME** viene usato dai computer Apple al posto di CLS, per cancellare lo schermo.

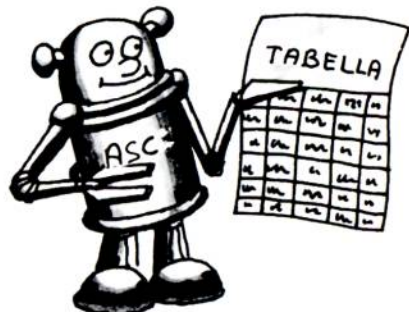


**ABS** ignora i segni "più" e "meno" davanti ai numeri e ne prende i valori "assoluti". Ad esempio ABS (- 10) vale 10, e anche ABS (+ 10) vale 10.



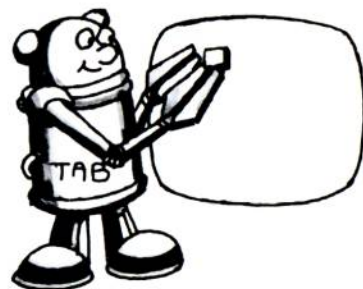
**VAL** prende il valore numerico di numeri scritti come stringhe. Di fatto dice al computer di ignorare il simbolo del dollaro, e di trattare la stringa come una normale variabile numerica. Ad esempio se I\$="60" allora VAL(I\$) equivale al numero 60.

**ASC** converte un carattere nel proprio codice numerico ASCII, ad esempio ASC("3") dà il valore 51. L'espressione tra parentesi deve essere una stringa, ad esempio ASC(A\$) o ASC("20"). NB: Lo ZX81 e lo ZX Spectrum non usano il comando ASC, benché lo Spectrum faccia uso del codice ASCII.



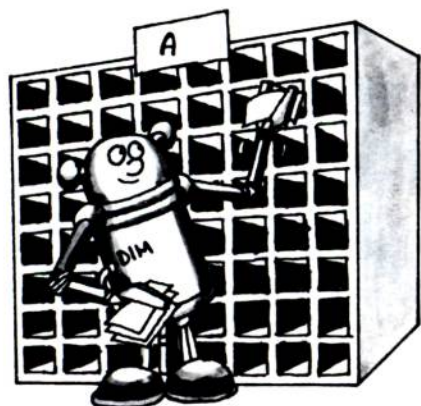
**CODE** viene usato dallo ZX81 e dallo Spectrum al posto del comando ASC. Come ASC, deve sempre venire seguito da una stringa. Ricordate che lo ZX81 utilizza i numeri di codice diversi dagli altri computers.

**TAB** muove il cursore in orizzontale sullo schermo, fino ad uno specificato numero di colonna. Viene di solito usato assieme a PRINT per visualizzare qualcosa al centro dello schermo. Il numero di spazi di cui volete spostare il cursore viene scritto tra parentesi dopo la parola TAB. Il valore massimo che possiate usare per questo numero dipende dalla larghezza di schermo gestibile dal vostro computer.





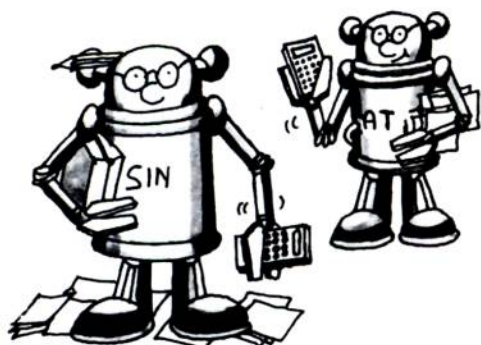
**SGN** dice al computer di individuare il segno di un numero. Fornisce come risultato  $-1$  per un numero negativo,  $0$  per lo zero, e  $+1$  per i numeri positivi. Ad esempio  $SGN(-30)$  vale  $-1$ ,  $SGN(7)$  vale  $+1$ , e  $SGN(0)$  vale  $0$ .



**DIM** dice al computer quanto spazio di memoria gli servirà per una "matrice" (di una sola riga o un reticolo). Ad esempio  $DIM X(6)$  dice al computer di riservare un'area abbastanza grande da contenere una riga di 6 elementi chiamata  $X$ .  $DIM A(8,8)$  vuol dire che serve uno spazio di memoria chiamato  $A$  e grande abbastanza da contenere 8 elementi su ciascuna riga e 8 su ciascuna colonna. Il numero di dati usati nel programma deve corrispondere ai numeri tra parentesi dopo la parola  $DIM$ , o vi troverete con un errore nel programma.



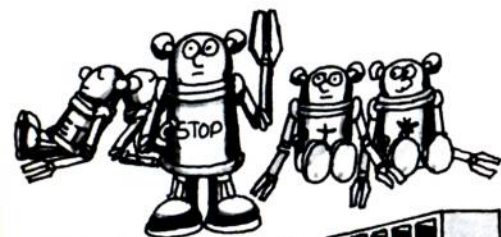
**SQR** estrae la radice quadrata dei numeri. Ad esempio  $SQR(16)$  dà la risposta  $4$ .



**SIN** calcola il seno di un angolo. In un triangolo rettangolo la lunghezza del lato opposto ad un angolo divisa per la lunghezza dell'ipotenusa (il lato opposto all'angolo retto) fornisce il seno di quell'angolo. Quando usate la funzione  $SIN$  in un programma, l'angolo con il quale la usate deve essere misurato in radianti, non in gradi.

**ATN** è una delle funzioni trigonometriche che i computers possono calcolare direttamente (vedere anche la precedente funzione  $SIN$ ).  $ATN$  sta per arcotangente ed è importante ricordare che fornisce un risultato in radianti, non in gradi. Vi servirà consultare un libro di matematica per scoprire come funziona tutto ciò, se non lo sapete già.

**STOP** dice al computer di non procedere oltre, nell'esecuzione di un programma. I computers diversi dallo ZX81 possono invece usare il comando  $END$ .



**PEEK** offre un modo per scoprire cosa si trovi in un'area particolare della memoria del computer. Dovrete usarlo con un numero che specifichi un "indirizzo" nella memoria. NB: non viene utilizzato sul modello BBC.



**POKE** fornisce un modo particolare per mettere informazioni nella memoria del computer, utilizzando un "indirizzo" di memoria. NB: non viene utilizzato sul modello BBC.

## Tabella ASCII

| Codice | Caratteri ASCII | Codice | Caratteri ASCII |
|--------|-----------------|--------|-----------------|
| 32     | space           | 62     | >               |
| 33     | !               | 63     | ?               |
| 34     | "               | 64     | (@              |
| 35     | #               | 65     | A               |
| 36     | \$              | 66     | B               |
| 37     | %               | 67     | C               |
| 38     | &               | 68     | D               |
| 39     | '               | 69     | E               |
| 40     | (               | 70     | F               |
| 41     | )               | 71     | G               |
| 42     | *               | 72     | H               |
| 43     | +               | 73     | I               |
| 44     | ,               | 74     | J               |
| 45     | -               | 75     | K               |
| 46     | .               | 76     | L               |
| 47     | /               | 77     | M               |
| 48     | 0               | 78     | N               |
| 49     | 1               | 79     | O               |
| 50     | 2               | 80     | P               |
| 51     | 3               | 81     | Q               |
| 52     | 4               | 82     | R               |
| 53     | 5               | 83     | S               |
| 54     | 6               | 84     | T               |
| 55     | 7               | 85     | U               |
| 56     | 8               | 86     | V               |
| 57     | 9               | 87     | W               |
| 58     | :               | 88     | X               |
| 59     | ;               | 89     | Y               |
| 60     | <               | 90     | Z               |
| 61     | =               |        |                 |

## Codice tabella ZX81

| Codice | Caratteri ZX81 | Codice | Caratteri ZX81 |
|--------|----------------|--------|----------------|
| 11     | "              | 41     | D              |
| 12     | £              | 42     | E              |
| 13     | \$             | 43     | F              |
| 14     | :              | 44     | G              |
| 15     | ?              | 45     | H              |
| 16     | (              | 46     | I              |
| 17     | )              | 47     | J              |
| 18     | >              | 48     | K              |
| 19     | <              | 49     | L              |
| 20     | =              | 50     | M              |
| 21     | +              | 51     | N              |
| 22     | -              | 52     | O              |
| 23     | *              | 53     | P              |
| 24     | /              | 54     | Q              |
| 25     | ;              | 55     | R              |
| 26     | ,              | 56     | S              |
| 27     | .              | 57     | T              |
| 28     | 0              | 58     | U              |
| 29     | 1              | 59     | V              |
| 30     | 2              | 60     | W              |
| 31     | 3              | 61     | X              |
| 32     | 4              | 62     | Y              |
| 33     | 5              | 63     | Z              |
| 34     | 6              |        |                |
| 35     | 7              |        |                |
| 36     | 8              |        |                |
| 37     | 9              |        |                |
| 38     | A              |        |                |
| 39     | B              |        |                |
| 40     | C              |        |                |

## Tabella dimensioni schermi

|             | Massimo numero di righe | Massimo numero di colonne |
|-------------|-------------------------|---------------------------|
| VIC 20      | 22                      | 23                        |
| TRS-80      | 64                      | 16                        |
| BBC         | 20/40/80                | 16/24/32                  |
| ZX81        | 32                      | 22                        |
| ZX Spectrum | 32                      | 22                        |
| Apple       | 40                      | 25                        |

# Tabella di conversione

Questa tabella di riferimento rapido mostra alcune delle differenze fra i BASIC usati dalle macchine in questo libro. Non comprende le istruzioni per grafica, suono e colore dal momento che variano enormemente da macchina a macchina. Notate anche che sebbene la maggior parte dei computers (escluso il BBC) usino PEEK e POKE, essi non usano lo stesso sistema di indirizzi della memoria, così i numeri usati con PEEK e POKE devono essere modificati per ogni computer.

|                                                | BBC                               | VIC/C 64                          | Apple                                           | TRS-80                            | ZX Spectrum                        | ZX81                              |
|------------------------------------------------|-----------------------------------|-----------------------------------|-------------------------------------------------|-----------------------------------|------------------------------------|-----------------------------------|
| Scegli un numero casuale fra 0 e 0,99999999    | RND(1)                            | RND(1)                            | RND(1)                                          | RND(0)                            | RND                                | RND                               |
| Scegli un numero casuale fra 1 e N             | RND(N)                            | RND(1)*N+1                        | RND(1)*N+1                                      | RND(N)                            | RND*N+1                            | RND*N+1                           |
| Scegli una lettera a caso fra A e Z            | CHR\$(RND(26)+64)                 | CHR\$(INT(RND(1)*26+65))          | CHR\$(INT(RND(1)*26+65))                        | CHR\$(RND(26)+64)                 | CHR\$(INT(RND*26+65))              | CHR\$(INT(RND*26+38))             |
| Cancella lo schermo                            | CLS                               | PRINT CHR\$(147)                  | HOME                                            | CLS                               | CLS                                | CLS                               |
| Controlla se è premuto un tasto qualsiasi      | INKEY\$(N)                        | GET X\$                           | X\$=" "<br>IF PEEK(-16384)<br>>127 THEN GET X\$ | INKEY\$                           | INKEY\$                            | INKEY\$                           |
| Converti i caratteri in numeri di codice       | ASC("X")<br>(usando codice ASCII) | ASC("X")<br>(usando codice ASCII) | ASC("X")<br>(usando codice ASCII)               | ASC("X")<br>(usando codice ASCII) | CODE("X")<br>(usando codice ASCII) | CODE("X")<br>(usando codice ZX81) |
| Muovi il cursore in alto                       | PRINT CHR\$(11)                   | PRINT CHR\$(145)                  | CALL -998                                       | PRINT CHR\$(27)                   | PRINT CHR\$(11)                    | PRINT CHR\$(112)                  |
| Muovi il cursore in basso                      | PRINT CHR\$(10)                   | PRINT CHR\$(17)                   | PRINT CHR\$(10)                                 | PRINT CHR\$(26)                   | PRINT CHR\$(10)                    | PRINT CHR\$(113)                  |
| Muovi il cursore a sinistra                    | PRINT CHR\$(8)                    | PRINT CHR\$(157)                  | PRINT CHR\$(8)                                  | PRINT CHR\$(24)                   | PRINT CHR\$(8)                     | PRINT CHR\$(114)                  |
| Muovi il cursore a destra                      | PRINT CHR\$(9)                    | PRINT CHR\$(29)                   | PRINT CHR\$(21)                                 | PRINT CHR\$(25)                   | PRINT CHR\$(9)                     | PRINT CHR\$(115)                  |
| Prendi i primi N caratteri di una stringa      | LEFT\$(A\$,N)                     | LEFT\$(A\$,N)                     | LEFT\$(A\$,N)                                   | LEFT\$(A\$,N)                     | A\$(1 TO N)                        | A\$(1 TO N)                       |
| Prendi gli ultimi N caratteri di una stringa   | RIGHT\$(A\$,N)                    | RIGHT\$(A\$,N)                    | RIGHT\$(A\$,N)                                  | RIGHT\$(A\$,N)                    | A\$(N TO )                         | A\$(N TO )                        |
| Prendi gli N caratteri centrali di una stringa | MID\$(A\$,N1,N2)                  | MID\$(A\$,N1,N2)                  | MID\$(A\$,N1,N2)                                | MID\$(A\$,N1,N2)                  | A\$(N1 TO N2)                      | A\$(N1 TO N2)                     |



## Risposte

Potrete trovare che le risposte da voi date ad alcuni dei quesiti sono diverse da quelle presentate qui. Ciò non ha molta importanza, fintanto che esse funzionino sul vostro computer, ma verificate comunque che siano altrettanto ordinate e semplici come le risposte del libro.

### Pagina 5 Il decollo dell'astronave

Le linee 30 e 40 fissano i numeri che determinano l'entità della forza. Per aumentare l'intervallo di forze possibili, potrete aumentare il 20 alla linea 30, o il 40 alla linea 40 o entrambi tali numeri. L'ampliamento dell'intervallo di forze renderà ovviamente il gioco più difficile.

### Pagina 7 Giochi Intergalattici

Modificate le linee 22 e 230 come segue:

```
222 LET B=B+INT(1000/G)
230 GOTO 20
```

ed aggiungete una nuova linea 15:

```
15 LET B=0
```

### Pagina 9 L'Alieno Malvagio

Modificate le linee 20 e 30, e aggiungete una nuova linea 25, come segue:

```
20 PRINT "DIFFICOLTA'? (6-30)"
25 INPUT S
30 LET G=INT(S/3)
```

### Pagina 11 Spara agli occhi del rettile

Per far apparire i rettili in più di quattro punti sullo schermo, dovete scrivere un numero maggiore di 4 a metà della linea 70, modificare la linea 80 e aggiungere delle nuove subroutines in fondo al programma una per ogni posizione in più.

Ecco le modifiche da apportare per far apparire i rettili in 5 punti diversi:

```
70 LET R=INT(RND*5+1)
```

```
S▲80 GOSUB 220+20*R
●VIC80 ON R GOSUB 240,260,280,300,320
```

```
240 LET D=5
245 LET A=1
250 GOTO 350
260 LET D=1
265 LET A=9
270 GOTO 350
280 LET D=5
285 LET A=18
290 GOTO 350
300 LET D=10
305 LET A=7
310 GOTO 350
320 LET D=15
325 LET A=15
330 GOTO 350
```

Potete usare qualsiasi numero vi piaccia, per A e D, purché rientrino nelle dimensioni del vostro schermo.



Per aggiungere altri rettili, sostituite il 10 alle linee 30 e 220 con un numero maggiore. (Assicuratevi di usare lo stesso numero in entrambe le linee).

### Pagina 13 Modulo Lunare

Per aumentare il valore massimo di velocità permessa per un buon atterraggio, dovrete apportare delle modifiche alle linee 230, 240 e 250. Potrete usare i numeri che preferite tanto più alti saranno, tanto più facile sarà il gioco. Nell'esempio che segue, vi è permessa una velocità di 2 per un buon atterraggio, e di 7 per un atterraggio accettabile:

```
230 IF V1>7 THEN PRINT "TI SEI SCHIANTATO
- SIETE TUTTI MORTI"
240 IF V1>V2 AND V1<=7 THEN PRINT "
OK - MA CON FATICA!"
250 IF V1<=2 THEN PRINT "BUON ATTERRAGGIO"
```

### Pagina 15 I Mostri di Galaction

Quattro modi per rendere più difficile questo gioco sono:

1. Iniziare il gioco con un minor numero di persone nel vostro gruppo, scrivendo un numero minore di 5 alla linea 40.
2. Aumentare il numero dei mostri cambiando il 4 alle linee 20 e 30. Aggiungete i nomi dei nuovi mostri alle linee 81-89, usando M\$(5) e M\$(6).



- Ridurre il numero di turni permessi, modificando l'8 alla linea 160.
- Aumentare la probabilità che il mostro si arrabbi, alla linea 330, aumentando un po' il 0.4.

## Pagina 17

### Pattuglie Aliene

In questo gioco, N è il numero di codice. Per modificare il sistema di punteggio per adattarlo al numero di codice, dovrete aumentare ogni volta il punteggio di N punti, anziché di uno. Per cui modificate la linea 190 come segue:

```
S ▲ 190 IF I$=CHR$(CODE(L$)+N) THEN
      LET S=S+N
● VIC 190 IF I$=CHR$(ASC(L$)+N) THEN
      LET S=S+N
```

## Pagina 19

### La Fascia di Asteroidi

Dovrete modificare la linea 260, cosicché il computer aggiunga al vostro punteggio il numero di stelle, anziché 1. Il numero di stelle è controllato dal valore scelto per N alla linea 70, quindi, come nel quesito precedente, dovrete aggiungere N al punteggio. Dovrete anche cambiare la linea 320.

```
260 LET S=S+N
320 PRINT "HAI OTTENUTO ";S;" PUNTI"
```

## Pagina 21

### Viaggio nel Futuro

- Per aumentare il valore dell'intervallo di anni da trascorrere prima del vostro ritorno sulla Terra, sostituite il 100 alla linea 30 con un numero più alto, ad esempio 150, in questo modo:

```
30 LET T=INT(RND*150+25)
```

- Per aumentare la precisione richiesta da 5 a 2 anni, sostituite i 5 alle linee 180 e 190 con dei 2, in questo modo:

```
180 IF ABS(T-T2)<=2 THEN PRINT
      "SEI ARRIVATO IN TEMPO"
190 IF ABS(T-T2)>2 THEN PRINT
      "TROPPO TARDI!"
```

- La linea 170 contiene il numero che determina la durata di tempo che vi resta da vivere. Sostituite il 50 con un valore più alto, per avere una vita più lunga.

## Pagina 23

### La Valle della Morte

Potrete rendere più lunga la valle sostituendo il numero alla linea 30 con un valore maggiore di 200.

## Pagina 25

### Le miniere Spaziali

Aggiungete queste linee per farvi chiedere dal computer se volete giocare ancora:


```
645 PRINT "UN ALTRO GIOCO? (S/N)"
646 INPUT A$
647 IF A$="S" THEN GOTO 10
```

Dovete poi aggiungere una nuova linea 5, e modificare la linea 30, per sommare la quantità di denaro rimastavi al termine del gioco al denaro assegnatovi per il gioco successivo:

```
5 LET M=0
30 LET M=M+INT(RND*50+10)*P
```

(Accertatevi di usare la forma di RND propria del vostro computer).

© Copyright per l'edizione originale Usborne Publishing Ltd — 1982  
© Copyright per l'edizione italiana Gruppo Editoriale Jackson — 1985

Il nome Usborne e il marchio  sono marchi registrati dalla Usborne Publishing Ltd., 20 Garrick Street, London WC2E 9BJ, England.

Tutti i diritti sono riservati. Nessuna parte di questo libro può essere riprodotta, memorizzata in sistemi di archivio, o trasmessa in qualsiasi forma o mezzo, elettronico, fotocopia, registrazione o altri senza per la preventiva autorizzazione scritta dell'editore.

Stampato in Italia da: Grafika 78 - Via Trieste, 20 - Pioltello  
Fotocomposizione: Composit s.a.s. - Via Giordano Bruno, 8 - 56100 Pisa

# Speedy Computer

Il calcolatore è anche (o soprattutto?) una macchina divertente. Si può giocare con lui, gli si possono fare delle domande, lo si può usare per scrivere una poesia o per suonare.

Questa colorata serie di libri vi propone alcune delle cose più eccitanti che si possono fare con un calcolatore e vi spiega come farle.

Scritti in linguaggio chiaro e comprensibile a chiunque, arricchiti da una moltitudine di illustrazioni, questi libri rappresentano una spiritosa introduzione al mondo dei computer per chi comincia da zero.

## Giochi Spaziali

Chi ha detto che i computer servano solo per cose serie? Oltretutto, divertendosi si può imparare più volentieri.

## altri volumi di questa collana

### COMPUTER GRAFICA

Si può dire che il computer ne sa una più del diavolo. Per scoprirlo leggete questo libro.

### COSTRUISCI PROGRAMMI DI ADVENTURE PER IL TUO COMPUTER

Ragazzi, volete partire per viaggi avventurosi, tutti quelli che la vostra immaginazione può pensare? Compratevi un computer e questo libro!

### APPLICAZIONI PRATICHE DEL PERSONAL COMPUTER

Un libro che vi farà scoprire piacevolmente che esiste qualcosa di diverso dai giochi, più utile e ugualmente appassionante, nel mondo dei computer.

### INTRODUZIONE AL LINGUAGGIO MACCHINA

Un argomento "difficile" affrontato con linguaggio piacevole; preziose illustrazioni consentono l'immediata comprensione anche dei più complessi aspetti tecnici.