

PENSA UN NUMERO...

Imparare a programmare è un po' come imparare il gioco del calcio.

Teoricamente, si potrebbe studiare ogni singola mossa sulla carta e scendere in campo soltanto quando si è in grado di padroneggiarle tutte, ma ciò, oltre ad essere scarsamente divertente, sarebbe anche molto faticoso e lento.

Lo stesso si può dire della programmazione: si possono leggere e rileggere un'infinità di testi, ma il metodo migliore è proprio quello di 'entrare in campo'. Da dove iniziare?

INDOVINARE UN NUMERO

Il più semplice gioco per computer è quello nel quale la macchina 'pensa' un numero a caso e il giocatore deve indovinarlo.

D+R

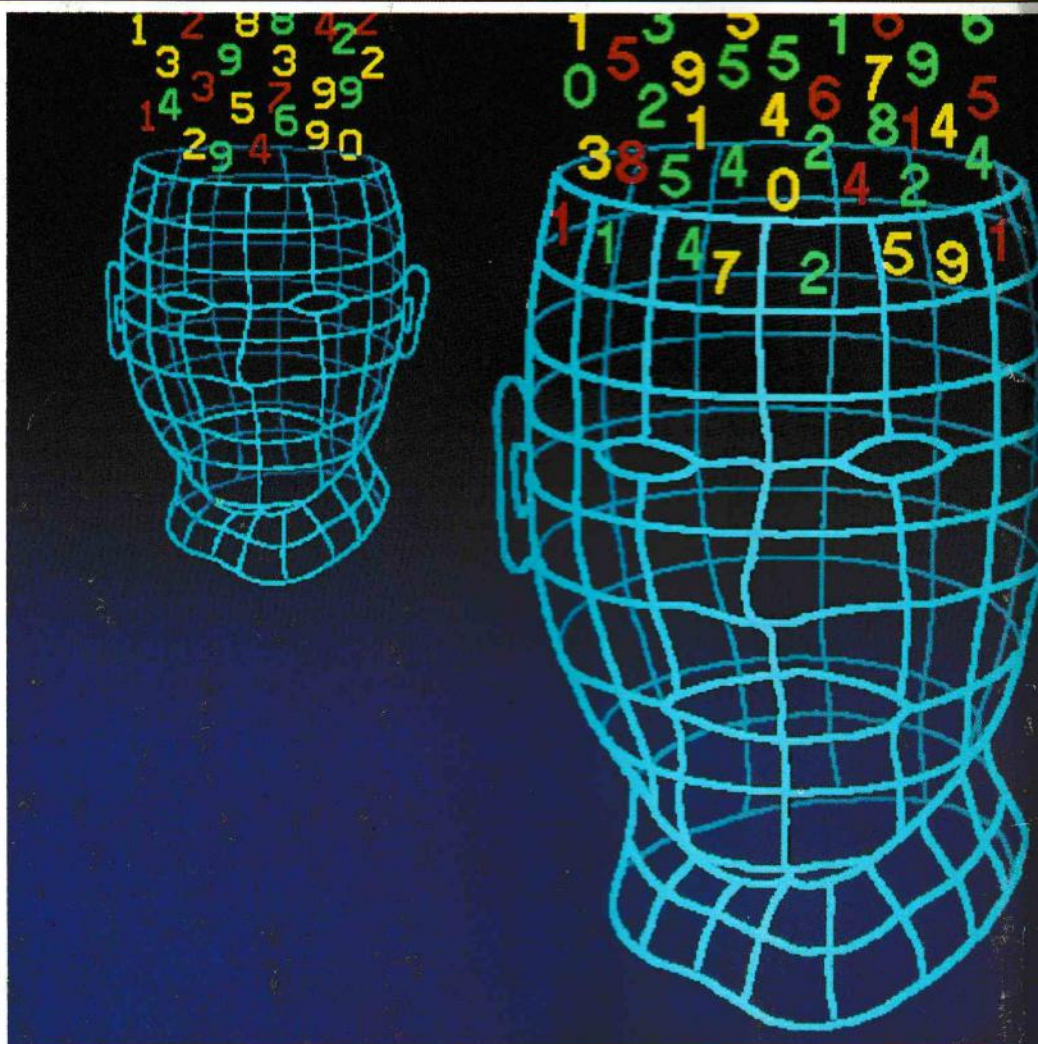
Su alcuni computer si usa l'istruzione **RANDOMIZE**.
A cosa serve?

Sullo Spectrum, usando **RANDOMIZE 1** (o qualsiasi altro numero), il computer genera sempre la stessa sequenza di numeri casuali.

Questa circostanza si rivela utile durante la ricerca di eventuali errori nel programma.

Sugli altri computer, è possibile ottenere il medesimo effetto per mezzo dell'istruzione **RND (-1)**. Anche in questo caso, il valore del numero, purché negativo, non è rilevante.

Usando, sullo Spectrum, **RANDOMIZE 0** o omettendo del tutto il numero, si ottiene l'effetto opposto, ossia che le sequenze dei numeri siano quanto più casuali possibile.



LA FUNZIONE RND

Tutti gli home computer sono dotati di un generatore casuale di numeri, utile in questo genere di giochi. In BASIC vi si accede mediante la funzione **RND**.

I numeri prodotti, tuttavia, non sono nella forma per noi più conveniente, essendo dei valori frazionari compresi tra 0 e 0,99999999. Verifichiamolo con un semplice programma, preceduto da un comando **NEW**, onde ripulire la memoria da eventuali programmi preesistenti:

SS

```
10 LET X=RND
20 PRINT X
30 GOTO 10
```

MT

```
10 LET X=RND (0)
20 PRINT X
30 GOTO 10
```

```
10 LET X=RND (1)
20 PRINT X
30 GOTO 10
```

(Ricordarsi di premere il tasto **RETURN** o **ENTER** al termine di ciascuna linea di programma).

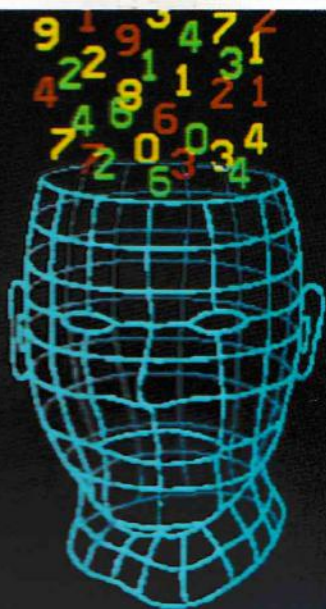
Adesso, impartendo un comando **RUN**, tutto ciò che vedremo sarà un'interminabile serie di numeri frazionari (si veda, a tal proposito, la tabella a pagina 7), poco utili per il nostro gioco.

Per ovviare a questo inconveniente, viene impiegata la funzione **INT** (abbreviazione di *intero*) e il valore viene opportunamente moltiplicato:

La maggior parte dei giochi usa la generazione di numeri casuali per creare sequenze di eventi apparentemente irregolari.

■	LA FUNZIONE RND
■	COME GENERARE NUMERI CASUALI
■	USO DELLE VARIABILI
■	L'ISTRUZIONE INPUT

■	L'USO DI IF ... THEN NEI CONFRONTI
■	DUE PROGRAMMI BASATI SUI NUMERI
■	LA GAMMA DEI NUMERI CASUALI



[ENTER], al termine di ciascuna linea).

Qualsiasi computer si possieda, l'arco di valori non è limitato a quelli tra 0 e 5, ma il gioco sarebbe arduo se scegliessimo una gamma tra 10 e 10 000!

USO DELLE VARIABILI

Nei brevi programmi appena presentati, non solo abbiamo generato un numero, ma gli abbiamo anche assegnato un nome: X.

Da questo momento in poi, se adoperiamo di nuovo X, il computer sa di dover fornire il numero 'scelto a caso'.

Questa X, che serve al computer per identificare un particolare numero e usarlo in operazioni aritmetiche oppure in un confronto, si chiama *variabile*.

L'ISTRUZIONE INPUT

Avendo generato il numero casuale, il passo successivo è quello di far sì che il computer accetti un nostro tentativo d'indovinarlo. A questo scopo ci serviamo dell'istruzione INPUT.

La sola parola INPUT non basta: occorre specificare anche un nome di variabile, nella quale il computer depositerà il valore immesso. Liberi di scegliere una lettera, o anche una combinazione di lettere qualsiasi (purché non X, già impegnata), usiamo una G.

L'istruzione, per intero (da non immettere ancora), è quindi:

```

10 LET X=INT (RND*6)
INPUT G

```

Adesso, però, è necessario che il computer confronti il valore da lui generato con quello da noi immesso. Per far ciò impieghiamo una combinazione particolare del BASIC:

```

10 LET X=RND (6)-1
20 PRINT X
30 GOTO 10
IF X=G THEN PRINT "RISPOSTA ESATTA"

```

La frase IF ... THEN (*se ... allora*) è molto utile e frequentemente impiegata nei programmi.



1. Il cursore (qui rappresentato da un carattere di sottolineatura, ma spesso da un rettangolino) indica la posizione dove verrà visualizzato il prossimo carattere immesso nel computer dall'operatore.

In alcuni computer (Acorn e Dragon), ne esiste una versione più sofisticata, ossia IF ... THEN ... ELSE (*se ... allora ... altrimenti*), la quale fornisce un'alternativa qualora la condizione non sia soddisfatta.

Negli altri computer, privi di ELSE (Spectrum e Commodore), se la condizione non è soddisfatta, il programma passa semplicemente alla linea successiva.

IMPOSTAZIONE DEL PROGRAMMA

Adesso, immettiamo il programma per intero: (si noti che < > significa rispettivamente 'minore di e maggiore di', in altre parole *'diverso da'*).

```

20 LET X=INT (RND*6)
30 PRINT "IL COMPUTER HA SCELTO UN NUMERO TRA LO 0 ED IL 5. RIESCI AD INDOVINARLO?"
40 INPUT G
60 IF G=X THEN PRINT "RISPOSTA ESATTA"
80 IF G <> X THEN PRINT "ERRATO - RIPROVARE"

```

```

20 LET X=RND(6)-1

```

```

10 LET X=INT (RND*6)

```

```

10 LET X=INT (RND(1)*6)

```

In questo caso, i numeri generali sono compresi tra 0 e 5.

Su alcuni computer non occorre usare la INT, ad esempio:

```

10 LET X=RND (6)-1
20 PRINT X
30 GOTO 10

```

(Ricordarsi di premere il tasto [RETURN] o


```
30 PRINT "IL COMPUTER HA SCELTO UN
NUMERO TRA LO 0 ED IL 5. RIESCI AD
INDOVINARLO?"
```

```
40 INPUT G
```

```
60 IF G=X THEN PRINT "RISPOSTA
ESATTA" ELSE PRINT "ERRATO -
RIPROVARE"
```



```
20 LET X=RND(6)-1
```

```
30 PRINT "IL COMPUTER HA SCELTO UN
NUMERO TRA LO 0 ED IL 5. RIESCI AD
INDOVINARLO?"
```

```
40 INPUT G
```

```
60 IF G=X THEN PRINT "RISPOSTA
ESATTA" ELSE PRINT "ERRATO -
RIPROVARE"
```

Microtip

Una via d'uscita

Talvolta, i principianti trovano difficoltà nell'interrompere un programma in corso d'esecuzione, magari per tornare a lavorare sul listato. Ecco alcuni consigli su come procedere:



In primo luogo, si premano, contemporaneamente, **[CAPS SHIFT]** e **[SPACE]**. Se questo non dovesse funzionare, è molto probabile che il computer stia eseguendo un'istruzione **INPUT**. In questo caso, usare i tasti per il movimento del cursore e il tasto **[DELETE]** per rimuovere eventuali apici o doppi apici. Successivamente, premere **[STOP]** (**[SHIFT+A]**), seguito da **[ENTER]**. Sullo schermo dovrebbe apparire la scritta 'stop in INPUT'. Adesso si preme **[ENTER]** per esaminare il listato.



Si provi premendo il tasto **[RETURN]**, seguito dal comando **LIST**. Se questo non funziona, allora tenendo premuto **[RUN/STOP]**, usare il tasto **[RESTORE]**. Poi usare il comando **LIST**.



Premere **[ESCAPE]**, poi digitare il comando **LIST**. Se non funziona: premere **[BREAK]**, digitare **OLD**, seguito da **LIST**.



Premere **[BREAK]**, poi digitare il comando **LIST**. Se non funziona: premere il tasto **[RESET]**, quindi usare il comando **LIST**.



```
20 LET X=INT(RND(1)*6)
```

```
30 PRINT "IL COMPUTER HA SCELTO UN
NUMERO TRA LO 0 ED IL 5. RIESCI AD
INDOVINARLO?"
```

```
40 INPUT G
```

```
60 IF G=X THEN PRINT "RISPOSTA ESATTA"
```

```
80 IF G<>X THEN PRINT "ERRATO
-RIPROVARE"
```

Impartendo un **RUN**, il programma funziona, ma il gioco si esaurisce in una sola 'mandata' e sullo schermo rimangono scritte superflue.

A questo secondo inconveniente si ovvia facilmente, basta aggiungere:



```
10 CLS
```

```
50 CLS
```



```
10 PRINT "♡"
```

```
50 PRINT "♡"
```

Questo accorgimento serve per 'ripulire' lo schermo prima di eseguire il resto del programma.

Per ottenere più di un 'lancio', invece, la cosa si presenta leggermente più complicata. Per il momento possiamo usare, semplicemente:

```
90 GOTO 10
```

che fa ripartire automaticamente il programma.

LE VARIABILI ALFANUMERICHE

Un metodo migliore, anche se apparentemente più complicato, consiste nel chiedere al giocatore se intende continuare nel gioco. Iniziamo coll'immettere il programma per intero (sullo ZX81, tralasciare **OR A\$="s"** nella linea 110):



```
10 CLS
```

```
20 LET X=INT(RND*6)
```

```
30 PRINT "IL COMPUTER HA SCELTO UN
NUMERO TRA LO 0 ED IL 5. RIESCI AD
INDOVINARLO?"
```

```
40 INPUT G
```

```
50 CLS
```

```
60 IF G=X THEN PRINT "RISPOSTA
ESATTA"
```

```
70 IF G=X THEN GOTO 90
```

```
80 IF G<>X THEN PRINT "ERRATO
-RIPROVARE"
```

```
90 PRINT "PER UN ALTRO TENTATIVO
DIGITARE UNA S, SEGUITA DA ENTER"
```

```
100 INPUT A$
```

```
110 IF A$="S" OR A$="s" THEN GOTO 10
```

```
120 GOTO 100
```



```
10 CLS
```

```
20 LET X=RND(6)-1
```

```
30 PRINT "IL COMPUTER HA SCELTO UN
NUMERO TRA LO 0 ED IL 5. RIESCI AD
INDOVINARLO?"
```

```
40 INPUT G
```

```
50 CLS
```

```
60 IF G=X THEN PRINT "RISPOSTA
ESATTA" ELSE PRINT "ERRATO
-RIPROVARE"
```

```
90 PRINT "PER UN ALTRO TENTATIVO
DIGITARE UNA S, SEGUITA DA ENTER"
```

```
100 INPUT AS
```

```
110 IF AS="S" THEN GOTO 10
```

```
120 GOTO 100
```



```
10 CLS
```

```
20 LET X=RND(6)-1
```

```
30 PRINT "IL COMPUTER HA SCELTO UN
NUMERO TRA LO 0 ED IL 5. RIESCI AD
INDOVINARLO?"
```

```
40 INPUT G
```

```
50 CLS
```

```
60 IF G=X THEN PRINT "RISPOSTA
ESATTA" ELSE PRINT "ERRATO
-RIPROVARE"
```

```
90 PRINT "PER UN ALTRO TENTATIVO
DIGITARE UNA S, SEGUITA DA ENTER"
```

```
100 INPUT AS
```



2. L'importanza della punteggiatura in un programma. Le linee nella parte alta dello schermo sono le istruzioni dell'esecuzione: la virgola (,) serve per 'tabulare', il punto e virgola (;) per una visualizzazione compattata, senza separazioni, mentre un apice ('), nel Commodore, nello Spectrum e negli Acorn, serve per passare alla successiva riga dello schermo.

$$\begin{array}{rcl}
 12 \times 9 & = & 108 \\
 11 \times 9 & = & 99 \\
 10 \times 9 & = & 90 \\
 9 \times 9 & = & 81 \\
 8 \times 9 & = & 72 \\
 7 \times 9 & = & 63 \\
 6 \times 9 & = & 54 \\
 5 \times 9 & = & 45 \\
 4 \times 9 & = & 36 \\
 3 \times 9 & = & 27 \\
 2 \times 9 & = & 18 \\
 1 \times 9 & = & 9
 \end{array}$$

```

110 IF A$="S" THEN GOTO 10
120 GOTO 100

```



```

10 PRINT "□"
20 LET X=INT (RND(1)*6)
30 PRINT "IL COMPUTER HA SCELTO UN
   NUMERO TRA LO 0 ED IL 5. RIESCI AD
   INDOVINARLO?"
40 INPUT G
50 PRINT "□"
60 IF G=X THEN PRINT "RISPOSTA
   ESATTA": GOTO 90
80 PRINT "ERRATO—RIPROVARE"
90 PRINT "PER UN ALTRO TENTATIVO
   DIGITARE UNA S, SEGUITA DA ENTER"
100 INPUT A$
110 IF A$="S" THEN GOTO 10
120 GOTO 100

```

Come si può notare, prima si chiede al giocatore se desidera giocare ancora (linea 90) poi, per immettere la risposta, si adopera una INPUT nella linea 100.

Questa volta, però, c'è una differenza sostanziale: alla linea 40 viene richiesta

l'immissione di un valore numerico, mentre qui, invece, occorre digitare una 'S' o una 'N', ossia un carattere alfabetico. Ciò significa che dobbiamo adoperare un diverso tipo di variabile, adatto a contenere lettere (o parole intere) invece di numeri: scriviamo, pertanto, INPUT A\$.

Il simbolo del dollaro comunica al computer che la variabile chiamata A deve contenere lettere e A\$ è chiamata una *variabile 'stringa'*.

La differenza tra i due tipi di variabili diverrà più chiara in seguito. Per adesso basta ricordare che: per immettere *valori numerici* si usa INPUT A, INPUT B, INPUT X, e così via, mentre per immettere *lettere o parole*, occorre usare INPUT A\$, INPUT B\$, INPUT X\$, e così via.

La linea 120 serve per ripetere la domanda (e l'immissione della risposta, qualora questa sia diversa da 'S').

Infatti, l'esecuzione viene dirottata alla linea 100 finché il carattere immesso dalla tastiera non corrisponde a una S (maiuscola o minuscola a seconda del programma). In tal caso, il programma riparte dall'inizio con un nuovo indovinello.

IMPARIAMO LE MULTIPLICAZIONI

La funzione RND si rivela utile in molteplici occasioni. Supponiamo di voler insegnare a un bambino la tabellina del 9. Potremmo scrivere un programma così concepito:

```

10 PRINT "QUANTO FA 1 PER 9?"
20 INPUT A
30 IF A=9 THEN PRINT "CORRETTO"
40 PRINT "QUANTO FA 2 PER 9?"
50 INPUT B
60 IF B=18 THEN PRINT "CORRETTO"

```

... e così via, ma esso sarebbe decisamente chilometrico e non prevederebbe nemmeno il caso di una risposta errata!

Un metodo migliore, applicabile molto più in generale, è basato sull'impiego della funzione RND, con la quale, oltretutto, è possibile evitare una precisa sequenza nelle domande, ottenendo uno strumento didattico di maggior efficacia.

Nella stesura di un programma è sempre bene partire da un nucleo centrale, lasciando gli abbellimenti a un secondo tempo. Proviamo il seguente programma (digitando un NEW per cancellare eventuali programmi precedenti):



```
10 LET N=INT (RND*12+1)
20 PRINT "QUANTO FA□"; N; "□PER 9?"
30 INPUT A
40 IF A=N*9 THEN PRINT "CORRETTO"
```

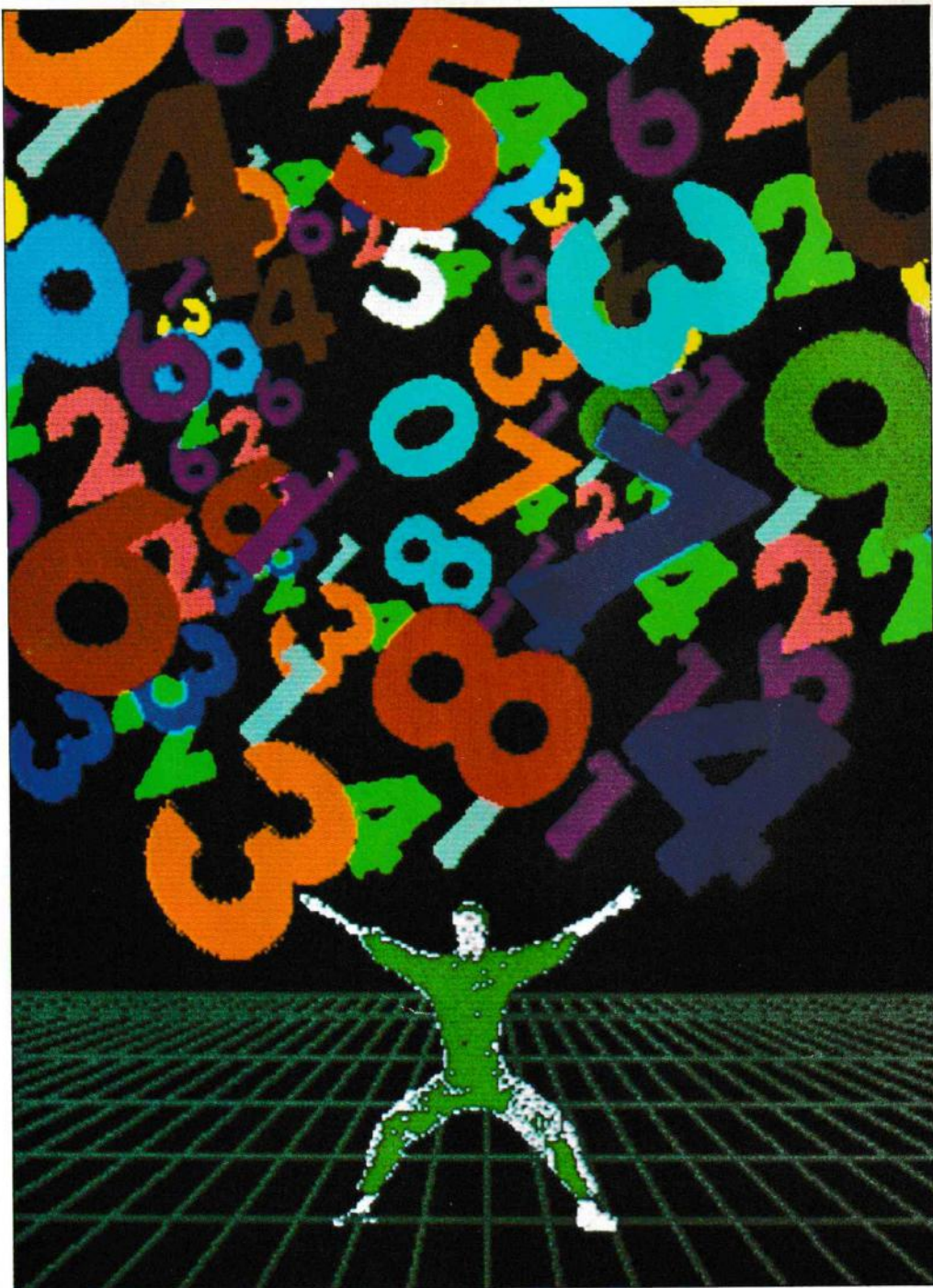


```
10 LET N=RND(12)
20 PRINT "QUANTO FA □"; N; "□ PER 9?"
30 INPUT A
40 IF A=N*9 THEN PRINT "CORRETTO"
```



```
10 N=INT (RND(0)*12)+1
20 PRINT "QUANTO FA□"; N; "□PER 9?"
30 INPUT A
40 IF A=N*9 THEN PRINT "CORRETTO"
```

La RND viene usata in modo molto simile a quello impiegato per il precedente gioco. Nella linea 10 si assegna a una variabile di nome N (il nome, si ricordi, è scelto a nostro piacimento) un valore generato a caso.



```
>PRINT (1+2)+3
9
>PRINT (4+5)/6
1.5
>PRINT (8-9)+2
4
>PRINT SQB (2)
1.41421356
>
```

3. I simboli per le operazioni aritmetiche. Il segno * (e non la x) serve per la moltiplicazione, il segno / per la divisione e ^ per l'elevazione a potenza. Per la somma e la sottrazione si usano, rispettivamente, + e -.

Il numero generato, in questo caso, è compreso tra 1 e 12. (Sullo Spectrum e sul Commodore occorre sommare 1, altrimenti otterremmo valori tra 0 e 11).

Nella linea 20 si chiede al giocatore di moltiplicare per nove il numero generato dal computer. La linea 40 segnala al computer che il numero casuale va moltiplica-

to per 9 e che il risultato va confrontato con la risposta del giocatore. Se questa è giusta, viene visualizzato (grazie alla PRINT) il messaggio 'ESATTO'. Si provi a 'lanciare' il programma (impartendo il comando RUN): esso funzionerà solamente una volta. Per far in modo che esso continui, si può aggiungere:

```
50 GOTO 10
```

ma perché non far le cose per bene, ossia:



```
10 PRINT "CIAO. COME TI CHIAMO?"
20 INPUT A$
30 CLS
```

```
40 PRINT "CIAO,□";A$,"HO ALCUNE","DOMANDE PER TE"
```

```
50 PAUSE 200
```

```
60 CLS
```

```
70 LET N=INT (RND*12)+1
```

```
80 PRINT "QUANTO FA□";N;"□PER 9?"
```

```
90 INPUT A
```

```
100 IF A=N*9 THEN GOTO 150
```

```
110 CLS
```

```
120 PRINT A;"□?"
```

```
130 PRINT "SPIACENTE, RIPROVA"
```

```
140 GOTO 80
```

```
150 PRINT "CORRETTO,□"; A$,"ECCO LA PROSSIMA"
```

```
160 PAUSE 150
```

```
170 GOTO 60
```




```

10 PRINT "CIAO. COME TI CHIAMI?"
20 INPUT AS
30 CLS
40 PRINT "CIAO,□";AS: PRINT "HO ALCUNE
DOMANDE PER TE"
50 FOR X=1 TO 6000:NEXT X
60 CLS
70 LET N=RND(12)
80 PRINT "QUANTO FA□";N;"□PER 9?"
90 INPUT A
100 IF A=N*9 THEN GOTO 150
110 CLS
120 PRINT A;"?"
130 PRINT "SPIACENTE, RIPROVA"
140 GOTO 80
150 PRINT"CORRETTO,□";AS:
PRINT "ECCO LA PROSSIMA"
160 FOR X=1 TO 4000:NEXT X
170 GOTO 60

```



```

10 PRINT "CIAO. COME TI CHIAMI?"
20 INPUT AS
30 PRINT "□"
40 PRINT "CIAO,□";AS: PRINT "HO ALCUNE
DOMANDE PER TE"
50 FOR X=1 TO 2000:NEXT X
60 PRINT "□"
70 N=INT(RND(1)*12)+1
80 PRINT "QUANTO FA□";N;"□PER 9?"
90 A=0:INPUT A
100 IF A=N*9 THEN GOTO 150
110 PRINT "□"
120 PRINT A;"□?"
130 PRINT "SPIACENTE, RIPROVA"
140 GOTO 80
150 PRINT "CORRETTO,□";AS:PRINT "ECCO
LA PROSSIMA"
160 FOR X=1 TO 2000:NEXT X
170 GOTO 60

```

La funzione della maggior parte delle linee aggiunte appare evidente se si esegue il programma.

Si noti, comunque, che:

le linee 30, 60 e 110 servono per ripulire lo schermo da tutte le scritte superflue e da eventuali risposte errate (per verificarlo, si provi a lanciare il programma dopo



4. I numeri posti all'inizio di ciascuna linea di programma sono molto importanti. Senza di essi, il computer, in BASIC, non saprebbe in quale ordine eseguire le istruzioni. È utile numerare le linee di 10 in 10, permettendo, così, eventuali inserimenti successivi.

aver cancellato queste tre linee); le linee 50 e 160 servono soltanto per assicurare delle pause nell'esecuzione e ciò si ottiene semplicemente 'facendo contare' il computer da 1 fino a un certo numero sufficientemente grande, prima di proseguire col programma (il metodo è spiegato nella seconda lezione).

Nel programma per lo Spectrum, le virgole nelle linee 40 e 150 sono state inserite unicamente per evitare che le parole vengano troncate alla fine di una riga di schermo. Nelle altre versioni, questo si ottiene aggiungendo un '.' e una ulteriore PRINT.

A questo punto, il programma ha un solo difetto: non si ferma più!

Per ovviare a ciò:



Premere [STOP], poi [ENTER].



Premere [BREAK].



Premere [ESCAPE].



Premere [RUN/STOP].

Modificare il programma per la tabellina del 5, del 7, o di qualsiasi altro numero, è di una facilità estrema.

Inoltre, rispetto alla prima versione, c'è il vantaggio che non occorre conoscere a priori le risposte esatte per scrivere il programma!



Come si specifica un arco di valori nella generazione dei numeri casuali?

RND sullo Spectrum, RND(1) sugli Acorn e RND(0) sul Dragon e sul Commodore, generano un numero casuale compreso tra 0 e 0,999999. Se si desidera un arco di valori diverso, occorre moltiplicare il valore ottenuto con la funzione RND. Per esempio, moltiplicando per 40, si ottiene un numero compreso tra 0 e 39,999999.

Volendo ottenere soltanto numeri interi (senza valori dopo la virgola), si usa la funzione INT e l'espressione risulta: INT (appropriata funzione RND*40), che genera valori interi tra 0 e 39. La funzione INT elimina, infatti, la parte decimale del numero.

Se, infine, volessimo un numero compreso tra 1 e 40, basta sommare 1 al valore appena ottenuto. Il Dragon e gli Acorn sono dotati di una scorciatoia: la funzione RND (40), nel nostro caso, fornisce già un valore intero. Si osservi la tabella sottostante per i vari esempi.

La generazione di numeri casuali

Genera un numero tra 0 e 0,999999
 Genera un numero tra 0 e n*0,999999
 Genera un numero tra -10 e +10
 Genera un numero tra 0 e 39
 Genera un numero tra 1 e 40



RND(1)
 RND(1)*n
 RND(21) - 11
 INT(RND(1)*40)
 RND(40)



RND(0)
 RND(0)*n
 INT(RND(0)*21) - 10
 INT(RND(0)*40)
 INT(RND(0)*40) + 1



RND(0)
 RND(0)*n
 RND(21) - 11
 INT(RND(0)*40)
 RND(40)



RND
 RND*n
 INT(RND*21) - 10
 INT(RND*40)
 INT(RND*40) + 1