

Basic: esempi di programmazione



Somma tra due numeri positivi	2542
Moltiplicazione di due numeri positivi attraverso la somma .	2542
Divisione intera tra due numeri positivi	2543
Elevamento a potenza	2543
Radice quadrata	2544
Fattoriale	2544
Ricerca sequenziale	2545

In questo capitolo si raccolgono solo alcuni esempi molto semplici di programmazione in Basic. Infatti, questo linguaggio di programmazione non si presta per la rappresentazione di algoritmi complessi.

Somma tra due numeri positivi	2542
Moltiplicazione di due numeri positivi attraverso la somma .	2542
Divisione intera tra due numeri positivi	2543
Elevamento a potenza	2543
Radice quadrata	2544
Fattoriale	2544
Ricerca sequenziale	2545

Somma tra due numeri positivi



Il problema della somma tra due numeri positivi, attraverso l'incremento unitario, è descritto nella sezione [62.3.1](#).

```
1000 REM =====
1010 REM somma.bas
1020 REM Somma esclusivamente valori positivi.
1030 REM =====
1040 REM
1050 INPUT "Inserisci il primo valore "; X
1060 INPUT "Inserisci il secondo valore "; Y
1070 LET Z = X
1080 FOR I = 1 TO Y
1090 LET Z = Z + 1
1100 NEXT
1110 PRINT X; "+"; Y; "="; Z
1120 END
1130 REM =====
```

Moltiplicazione di due numeri positivi attraverso la somma



Il problema della moltiplicazione tra due numeri positivi, attraverso la somma, è descritto nella sezione [62.3.2](#).

```
1000 REM =====
1010 REM moltiplica.bas
1020 REM Moltiplica esclusivamente valori positivi.
1030 REM =====
1040 REM
1050 INPUT "Inserisci il primo valore "; X
1060 INPUT "Inserisci il secondo valore "; Y
1070 LET Z = 0
1080 FOR I = 1 TO Y
1090 LET Z = Z + X
1100 NEXT
1110 PRINT X; "*"; Y; "="; Z
1120 END
1130 REM =====
```

Divisione intera tra due numeri positivi

Il problema della divisione tra due numeri positivi, attraverso la sottrazione, è descritto nella sezione [62.3.3](#). «

```
1000 REM =====
1010 REM dividi.bas
1020 REM Divide esclusivamente valori positivi.
1030 REM =====
1040 REM
1050 INPUT "Inserisci il primo valore "; X
1060 INPUT "Inserisci il secondo valore "; Y
1070 LET Z = 0
1080 LET I = X
1090 IF I < Y THEN GOTO 1130
1100 LET I = I - Y
1110 LET Z = Z + 1
1120 GOTO 1090
1130 PRINT X; "/"; Y; "="; Z
1140 END
1150 REM =====
```

Elevamento a potenza

Il problema dell'elevamento a potenza tra due numeri positivi, attraverso la moltiplicazione, è descritto nella sezione [62.3.4](#). «

```
1000 REM =====
1010 REM exp.bas
1020 REM Eleva a potenza.
1030 REM =====
1040 REM
1050 INPUT "Inserisci il primo valore "; X
1060 INPUT "Inserisci il secondo valore "; Y
1070 LET Z = 1
1080 FOR I = 1 TO Y
1090 LET Z = Z * X
1100 NEXT
1110 PRINT X; "^"; Y; "="; Z
1120 END
1130 REM =====
```

Radice quadrata



Il problema della radice quadrata è descritto nella sezione [62.3.5](#).

```
1000 REM =====
1010 REM radice.bas
1020 REM Radice quadrata intera.
1030 REM =====
1040 REM
1050 INPUT "Inserisci il valore "; X
1060 LET Z = 0
1070 LET T = 0
1080 REM Inizio del ciclo di calcolo
1090 LET T = Z * Z
1100 IF T > X THEN GOTO 1130
1110 LET Z = Z + 1
1120 GOTO 1080
1130 REM Riprende il flusso normale
1140 LET Z = Z - 1
1150 PRINT "radq("; X; ") ="; Z
1160 END
1170 REM =====
```

Fattoriale



Il problema del fattoriale è descritto nella sezione [62.3.6](#).

```
1000 REM =====
1010 REM fatt.bas
1020 REM Fattoriale.
1030 REM =====
1040 REM
1050 INPUT "Inserisci il valore "; X
1060 LET Z = X
1070 FOR I = (X - 1) TO 1 STEP -1
1080 LET Z = Z * I
1090 NEXT
1100 PRINT "fatt("; X; ") ="; Z
1110 END
1120 REM =====
```

Ricerca sequenziale

Il problema della ricerca sequenziale all'interno di un array, è descritto nella sezione [62.4.1](#). «

```
1000 REM =====
1010 REM ricercaseq.bas
1020 REM Ricerca sequenziale.
1030 REM =====
1040 REM
1050 INPUT "Inserisci il numero di elementi "; N
1060 DIM A(N)
1070 FOR I = 0 TO N-1
1080 PRINT "A("; I; ") ="
1090 INPUT A(I)
1100 NEXT
1110 INPUT "Inserisci il valore da cercare "; X
1120 FOR I = 0 TO N-1
1130 IF X = A(I) THEN GOTO 1170
1140 NEXT
1160 GOTO 1190
1170 PRINT "L'elemento A("; I; ") contiene il valore "; X
1180 END
1190 PRINT "Il valore "; X; " non è stato trovato"
1200 END
1210 REM =====
```

