

# ISTRUZIONE WHILE

while (<condizione>)  
<istruzione>

```
while (<condizione>) {  
    istruzione 1;  
    ...  
    istruzione n;  
}
```

## Esempio

Programma che prende in input un numero  $n$  e stampa la radice quadrata di  $n$  intera, approssimata per eccesso.

0  $\mapsto$  0

1  $\mapsto$  1

2  $\mapsto$  1,4142...  $\mapsto$  2

3  $\mapsto$  2

4  $\mapsto$  2

$\sqrt{35}$

proviamo con 0! NON VA BENE perché  $0^2 < 35$

" " 1 " " perché  $1^2 = 1 < 35$

" " 2

" " 3

" " 4

5 NON VA BENE perché  $5^2 = 25 < 35$

6 OK! perché  $6^2 = 36 > 35$

Esempio 2 while che simula un for

```
for ( int <var> = <val. iniziale>; <var> <= <valore finale>; <var.>++)  
    <istruzione>;
```

numero di ripetizioni :  $<valore\ finale> - <valore\ iniziale> + 1$

```
for ( int i = 1; i <= 10; i++)  
    <istruzione>;
```

```
→ int i = 1;  
while ( i <= 10 ) {  
    <istruzione>;  
    i++;  
}
```

ISTRUZIONE FOR, CASO GENERALE

```
for ( <inizializzazione>; <condizione>; <aggiornamenti> ) {  
    <istruzione 1>  
    ⋮  
    <istruzione n>  
}
```

istruzione qualunque



```
<inizializzazione>  
while ( <condizione> ) {  
    <istruzione 1>  
    ⋮  
    <istruzione n>  
    <aggiornamenti>  
}
```

## ISTRUZIONE DO - WHILE

```
do  
  <istruzione>;  
while (<condizione>);
```



```
do {  
  <istruzione 1>;  
  ⋮  
  <istruzione n>;  
} while (<condizione>);
```

Come il while ma la condizione viene controllata alla fine del corpo e non all'inizio.

### Esempio

Scrivere un programma che simuli il lancio di un dado e conti quante volte viene lanciato prima di ottenere 6.