



# Costrutti iterativi: while, do while e for

Alessandra Giordani

[agiordani@disi.unitn.it](mailto:agiordani@disi.unitn.it)

Lunedì 2 aprile 2012

<http://disi.unitn.it/~agiordani/>

# Il ciclo while

## Sintassi

```
while (<condizione>) { /* istruzioni */ }
```

- 1 Se `<condizione> != 0`, vai a (2)
  - 2 Esegui il blocco `istruzioni`
  - 3 Se `<condizione> != 0`, vai a (2)
- Il blocco viene ripetuto finché la condizione é vera
  - Il blocco non viene mai eseguito se la condizione é falsa quando viene valutata la prima volta
  - Se l'esecuzione delle istruzioni non modifica il valore dalla condizione il programma continua a ripetere le istruzioni all'infinito ⇒ **infinite loop!**

# Esempio: somma dei primi n numeri

```
#include <stdio.h>
#include <stdlib.h>

main()
{
    int n, somma = 0;

    printf("Introdurre un intero positivo:\n");
    scanf("%d", &n);
    printf("Ecco la somma dei primi %d numeri: ", n);
    while (n)
    {
        somma+=n;
        n--;
    }
    printf("%d\n", somma);
    exit(0);
}
```



# Esempio: fattoriale di n

```
#include <stdio.h>
#include <stdlib.h>

main()
{
    int n;
    unsigned long fatt;

    printf("Introdurre un intero positivo:\n");
    scanf("%d", &n);
    fatt = 1;
    printf("%d! = ", n);
    while (n > 1)
    {
        fatt = fatt * n;
        n--;
    }
    printf("%ld\n", fatt);
    exit(0);
}
```



## Nota:

- Abbiamo dato per scontato che l'utente immettesse correttamente un numero positivo!
- Sarebbe meglio controllare usando un **if**

```
if (n<1) printf("Errore\n");  
    else {...}
```

# Il ciclo do..while

## Sintassi

```
do { /* istruzioni */ } while (<condizione>)
```

- 1 Esegui il blocco `istruzioni`
- 2 Se `<condizione> != 0`, vai a (1)

Analogo a `while`, però:

- Il blocco viene ripetuto finché la condizione é vera
- Il blocco viene comunque eseguito almeno una volta

# Esempio: conta numero di spazi

- Leggiamo una stringa carattere per carattere
  - getchar() prende un carattere dal buffer input
  - confronto se è uguale a spazio (o acapo)

```
#include <stdio.h>
int main()
{
    int ch, spazi=0;
    printf("Introdurre una frase:\n");
    do
    {
        ch = getchar();
        if (ch == ' ')
            spazi++;
    } while (ch != '\n');
    printf("Il numero di spazi è: %d.\n", spazi);
    exit(0);
}
```

# Il ciclo for

## Sintassi

```
for (<inizializzazione> ; <controllo>; <incremento>) {  
    /* istruzioni */  
}
```

- ① Esegui <inizializzazione>
  - ② Se <controllo> != 0, vai a (3), altrimenti esci
  - ③ Esegui il blocco di istruzioni
  - ④ Esegui <incremento>
  - ⑤ Vai a (2)
- Prima della prima iterazione, esegue <inizializzazione>
  - Prima di ogni iterazione, verifica che <controllo> != 0
  - Dopo ogni iterazione, esegue <incremento>



# Esempio: fattoriale di n, vers.2

```
#include <stdio.h>
#include <stdlib.h>

/* Calcola il fattoriale di n */

main()
{
    int n;
    unsigned long fatt;

    printf("Introdurre un intero positivo:\n");
    scanf("%d", &n);
    if (n<1)
        printf("Errore\n");
    else {
        printf("%d! = ", n);
        for (fatt = 1; n > 1; n--)
            fatt = fatt * n;
        printf("%ld\n", fatt);
    }
    exit(0);
}
```

# Esempio: da stringa ad intero

- Leggi una stringa di cifre
- Convertila in un numero intero

```
#include <stdio.h>
#include <ctype.h>
int main()
{
    int num=0, digit;
    printf("Scrivi un numero: ");
    digit = getchar();
    for(; isdigit(digit); digit=getchar())
    {
        num = num * 10;
        num = num + (digit - '0');
    }
    printf("Il numero intero e' %d\n", num);
    exit(0)
}
```

# Esempio: da stringa ad intero

- Usando un ciclo while...
- Migliore: una sola chiamata a getchar()

```
#include <stdio.h>
#include <ctype.h>
int main()
{
    int num=0, digit;
    printf("Scrivi un numero: ");
    while (isdigit(digit=getchar()))
    {
        num = num * 10;
        num = num + (digit - '0');
    }
    printf("Il numero intero e' %d\n", num);
    exit(0)
}
```



# exit VS return

## ■ return

- È un'istruzione (restituire un valore di funz.)

- Anche il main è una funzione

```
int main() { return 0; }
```

## ■ exit()

- È una funzione (della libreria standard)

- Si può usare solo nel main

```
#include <stdlib.h>
```

```
int main() { exit(0); }
```