

Informazioni

- **Ricevimento:** Martedì 11 dic.: dalle ore 14:00 alle ore 15:00.
- **Mercoledì 12 dicembre** ore 14.30
 - esame (P200, Edificio Paolotti)
 - [Iscriversi all'esame](#) (vedi link sulla mia pagina web)
- **Giovedì 13 dicembre** ore 10:30 (aula L1)
 - [correzione dell'esame](#)
- **Martedì 18 dicembre** ore 10:30 (aula G)
 - [visione compiti](#)
 - [Registrazione a febbraio](#) (metterò la data esatta sulla mia pagina web)

Esercizi (soluzioni)

- Quanti asterischi stampa il seguente ciclo?

```
for (int star=9; star<0; star++){cout << '*';}
```

Questo ciclo non stampa nessun asterisco.

- Cosa stampano in output le seguenti istruzioni?

```
int s = 1;
for (int x=0; x<=5; x++){
    s = s+x;
    cout << s << " ";
}
```

Stampano 1 2 4 7 11 16

Ciclo while -- 1

- Il **ciclo while** è un **ciclo condizionale** in cui un blocco di istruzioni viene indefinitamente **ripetuto finché non si verifica una condizione** che determina l'uscita dal ciclo
- Lo schema generale di un ciclo while è il seguente:

```
inizializzazione della condizione
while (condizione) {
    corpo
}
```
- Prima di entrare nel ciclo while, in generale ci sono delle **istruzioni che inizializzano le variabili coinvolte nella condizione** (cioè una espressione booleana) che governa il ciclo

Ciclo while -- 2

- **inizializzazione della condizione**

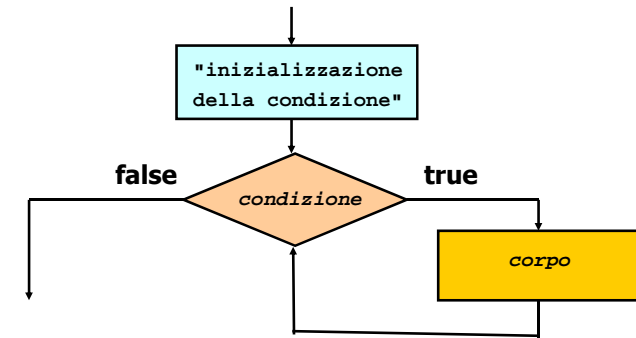
```
while (condizione) {
    corpo
}
```
- Il ciclo inizia verificando la **condizione**
- Se la **condizione** si valuta a **true** allora viene eseguito il **corpo** del ciclo while e di seguito il controllo ritorna all'inizio del ciclo, ovvero viene nuovamente verificata la **condizione**, il cui valore potrebbe essere cambiato in virtù dell'esecuzione del **corpo**.
- Questo processo iterativo è ripetuto fino a quando la **condizione** si valuta a **false**: in tal caso, si esce dal ciclo ed il controllo passa **all'istruzione che immediatamente segue il ciclo while**

Ciclo while -- 3

- **inizializzazione della condizione**
`while (condizione) {`
 corpo
`}`
- Due considerazioni molto importanti
 - la **condizione** del ciclo deve essere **inizializzata**: se la condizione non e' inizializzata, il ciclo potrebbe usare valori errati o addirittura indefiniti
 - il **valore della condizione** deve **mutare durante il ciclo**: se il valore della condizione non cambia durante il ciclo, non ci sara' alcuna possibilita' di uscita dal ciclo, che continuera' indefinitamente

Ciclo while

```
"inizializzazione della condizione"  
while (condizione) {  
    corpo  
}
```



Esempio 1

Cosa stampa il seguente programma?

```
#include<iostream>  
using namespace std;  
  
main(){  
    int k = -1;  
    while (k <= 4) {  
        k = k+2; cout << k << " ";  
    }  
}
```

stampa: 1 3 5

Esempio 2

Cosa stampa il seguente programma?

```
#include<iostream>  
using namespace std;  
  
main(){  
    int j = 9;  
    while (j > -1) {  
        j = j-3; cout << 2*j << " ";  
    }  
}
```

stampa: 12 6 0 -6

Ciclo do.. while..

- Come il ciclo **while** ma la condizione viene testata all fine, quindi il **corpo** viene **eseguito** almeno una volta.

- Lo schema generale di un ciclo do-while e' il seguente:

```
inizializzazione della condizione
do{
    corpo
}while(condizione);
```

Esempio

```
#include<iostream>
using namespace std;

//ciclo while
main(){
    int k = -1;
    while (k > 4) {
        k = k+2; cout << k << " ";
    }
    // non viene stampato niente in output

//ciclo do-while
main(){
    int k = -1;
    do{
        k = k+2; cout << k << " ";
    } while(k>4);
    // stampa: 1
```

Esercizi (cicli for e while)

- Scrivere un ciclo while equivalente al ciclo for infinito

```
for(int i = 0; i > -2; i++)
{cout<< i;}
```
- Trasformare il seguente ciclo while in un ciclo for equivalente:

```
int i = 1;
while (i < 20) {cout << i; i++}
```
- Cosa stampano in output i seguenti cicli?
 - ```
int i = -7, k = 1;
while (i < -2) {i = i + k; cout << i << " "; k++;}
```
  - ```
int i = -4, k = 3;
while (!(i == k + 1)) {i++; k--; cout << (k-i) << " "};
```
 - ```
for (int n = 0; n < 20; n = n + 2)
{ cout << n << " "};
```

## Sol. esercizi (cicli for e while) -- 1

- Scrivere un ciclo while equivalente al ciclo for infinito  

```
for(int i = 0; i > -2; i++)
{cout<< i;}
```

Soluzione: 

```
int i=0;
while(i>-2)
{ cout<<i; i++; }
```
- Trasformare il seguente ciclo while in un ciclo for equivalente:  

```
int i = 1;
while (i < 20) {cout << i; i++}
```

Soluzione: 

```
for(int i=1; i<20; i++)
{ cout<<i;}
```

## Sol. esercizi (cicli for e while) -- 2

- Cosa stampano in output i seguenti frammenti di programmi?
  - ```
int i = -7, k = 1;
while (i < -2) {i = i + k; cout << i << " "; k++;}
```

Stampa -6 -4 -1
 - ```
int i = -4, k = 3;
while (!(i == k + 1))
{i++; k--; cout << (k-i) << " "};
```

Stampa 5 3 1 -1
  - ```
for (int n = 0; n < 20; n = n + 2)
{ cout << n << " "};
```

Stampa 0 2 4 6 8 10 12 14 16 18