

Le istruzioni di iterazione

Nella risoluzione di problemi capita spesso di dover ripetere una stessa serie di istruzioni per un numero molto grande (ma finito) di volte.

Il linguaggio C/C++ dispone di tre tipi diversi di istruzioni iterative:

- **while;**
- **do ... while;**
- **for.**

Confronto fra while, do..while e for

while

prevede che il controllo della ripetizione delle istruzioni sia precedente alla loro esecuzione.

Si tratta di un costrutto **pre-condizionale**.

Il corpo del ciclo potrebbe non essere mai eseguito se la condizione è falsa alla prima esecuzione.

```
while(<Condizione>
{
  Istruzioni
}
```

do .. while

valuta la condizione alla fine del ciclo iterativo.

È un costrutto **post-condizionale**.

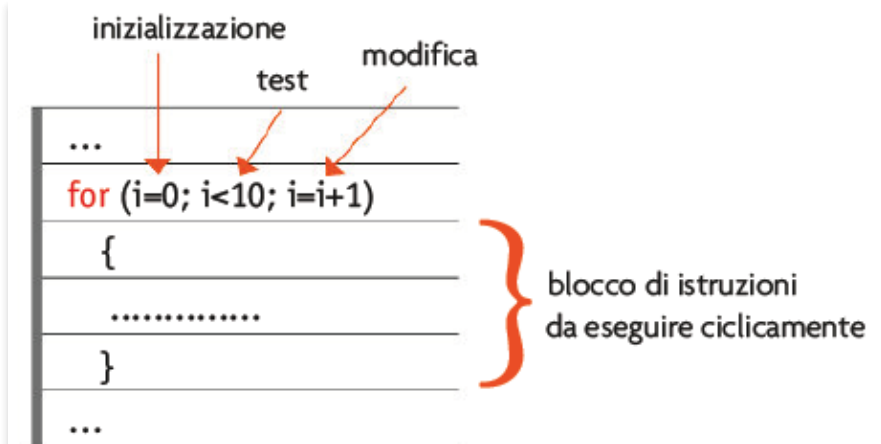
Per questo motivo, il corpo del ciclo do ... while viene sempre eseguito almeno una volta.

```
do
{
  Istruzioni
} while(<Condizione>;
```

for

ripetizione enumerativa o ripetizione con contatore: cioè istruzioni cicliche che devono essere ripetute per un numero prestabilito di volte.

```
for(<Inizializzazione_Indice>;
<Condizione_di _permanenza_del_ciclo>;
[<Incremento>])
{
  Istruzioni
}
```



Quale struttura iterativa conviene usare ?

- Le strutture «**while**» e «**do .. while**» sono molto simili ed in generale intercambiabili, è più conveniente usare il «do .. while» quando ci serve eseguire almeno una volta il corpo del ciclo (istruzioni interne) il «while» invece è più conveniente quando vogliamo verificare la condizione prima di eseguire il corpo del ciclo (se la condizione risulta immediatamente falsa l'esecuzione delle istruzioni del corpo del ciclo non viene mai fatta al contrario del «do while»).
- La struttura «**for**» si utilizza in generale quando sappiamo in anticipo quante volte ripetere le istruzioni del corpo del ciclo (ad esempio sappiamo di dover ripetere 10 volte una certa azione conviene usare il «for»). Nel for possiamo anche incrementare il contatore con numeri superiori ad uno (nel caso in cui ci serva contare con dei «salti» di numeri) e decrementarlo (nel caso ci serva contare al contrario)