

## ISTRUZIONI CONDIZIONALI

In C è possibile eseguire determinate operazioni solo se si verificano di alcune condizioni tramite i rami if/else:

```
if condizione(){  
    corpo;  
}  
else{  
    corpo;  
}
```

Se la condizione è vera si entra nel ramo if e vengono eseguite tutte le istruzioni al suo interno; dopodiché si esce dall'if ignorando l'else.

Se invece la condizione è falsa si passa direttamente al ramo else ignorando l'if. Notare che il ramo else non è obbligatorio.

È inoltre possibile valutare più condizioni con gli operatori logici AND (&&) ed OR(||).

```
int a = 42;  
  
if(a >= 0){  
    printf("il numero inserito è positivo\n");  
}  
else{  
    printf("il numero inserito è negativo\n");  
}  
  
//ovviamente tale frammento di programma stamperà a schermo il  
//messaggio "il numero inserito è positivo"
```

## IF ANNIDATI

L'annidazione degli if consiste nel l'inserire un if dentro un altro al fine di verificare condizioni sempre più specifiche

```
int min, ora, giorno;

if (min==59){
    if(ora==23){
        giorno++;
    }
    ora++;
}
else{
    min++;
}
```

## CICLI

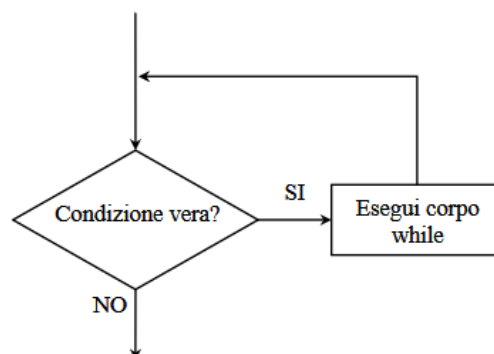
Le istruzioni di ciclo permettono di ripetere ciclicamente le stesse operazioni fino al verificarsi di una determinata condizione che mette fine al ciclo.

### CICLO WHILE

Il ciclo while esegue tutte le istruzioni al suo interno fin tanto che la condizione del ciclo stesso è vera. Una volta eseguite tutte le istruzioni si torna a valutare la condizione: se è ancora vera si ricomincia con l'esecuzione del corpo, altrimenti si esce dal ciclo.

```
int a = 0;

while(a<10){
    a++;
    printf("%d\n", a);
}
```



Appena entrati nel ciclo viene valutata la condizione  $a < 10$ ; essendo vera, si entra nel ciclo dove in ordine: incrementiamo il valore di  $a$ , e lo stampiamo.

Dopo 10 iterazioni la condizione  $a < 10$  sarà false e usciremo dal ciclo.

## CICLO FOR

Il ciclo for sostanzialmente lavora come il ciclo while: esegue tutte le istruzioni al suo interno finché una determinata condizione è vera.

Nel caso del for, però, tale condizione è data da un contatore che viene incrementato ad ogni ciclo, fino al raggiungimento di un determinato valore.

```
int a=0;
for(int i=0; i<10; i++){
    printf("%d\n", a);
}
```

Nella condizione del in sequenza:

inizializziamo una variabile `i=0`, controlliamo che sia minore di 10, e la incrementiamo ad ogni iterazione. Dunque al raggiungimento del valore 10 usciremo dal ciclo. In particolare, questo esempio è del tutto equivalente al precedente esempio sul ciclo while.

N.B. è sempre possibile tradurre in ciclo while in un ciclo for e viceversa.