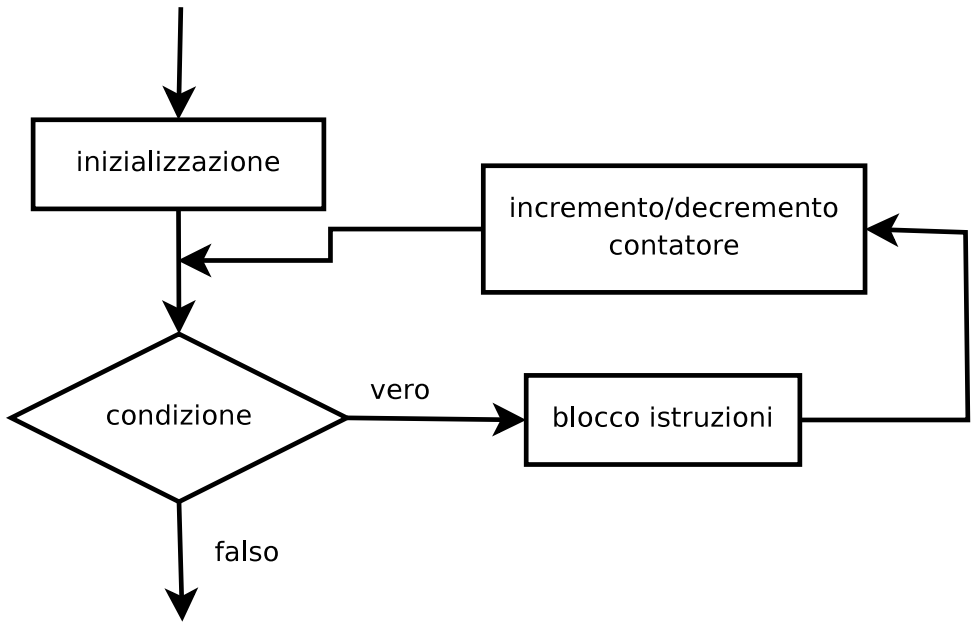
Il ciclo for

```
for ( inizializzazione contatore;
      test;
      incremento/decremento contatore
    ) { blocco istruzioni }
```



Operatori logici

- &&: AND;
- ||: OR;
- !: NOT.

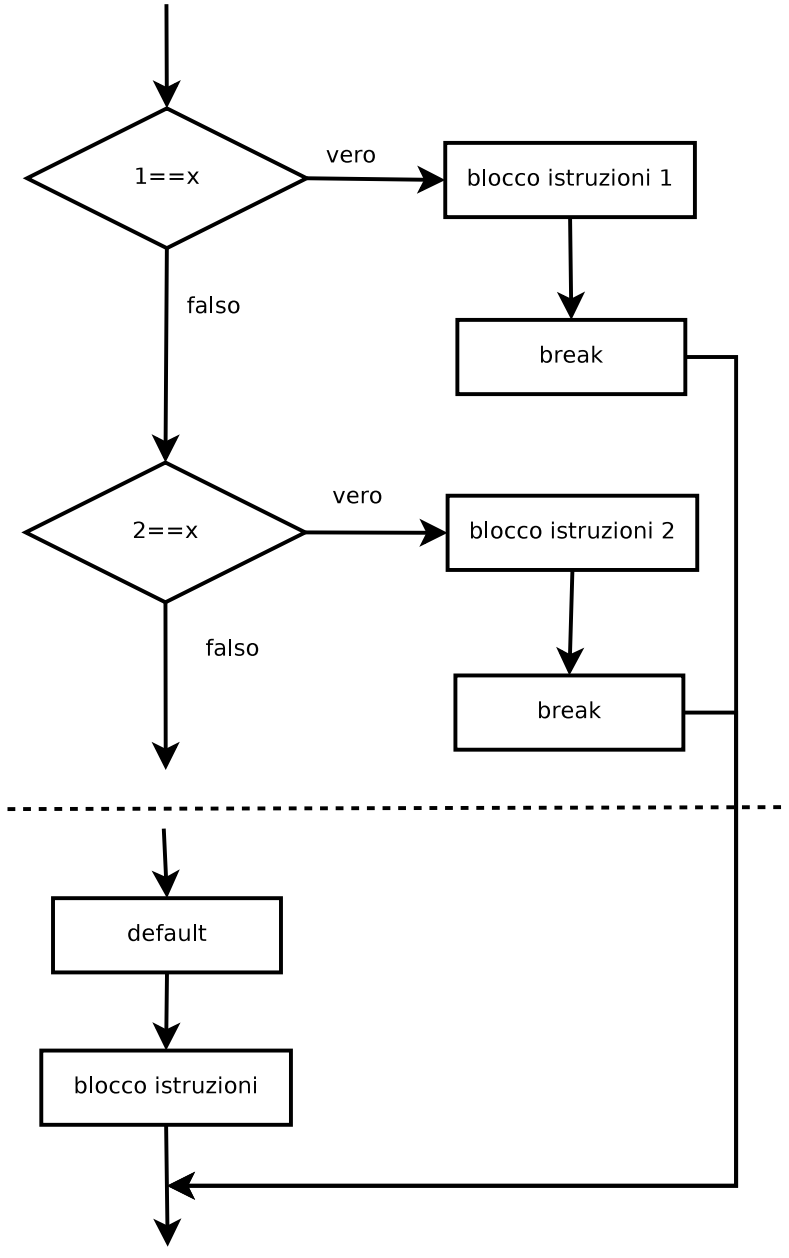
AND e OR sono ovviamente operatori binari, mentre il NOT è un operatore unario.
NOT ha priorità più alta degli operatori di relazione e confronto e degli altri operatori logici.
AND ha priorità più alta di OR. Entrambi hanno priorità più bassa degli operatori di relazione e confronto.

Esercizio

Una classe ha un certo numero di studenti; ogni studente viene interrogato un certo numero di volte e gli viene assegnato un voto da 1 a 10 ad ogni interrogazione.
Scrivere un programma in C che richieda quanti studenti ci sono nella classe e, per ogni studente, richieda il sesso (m o M per maschio e f o F per femmina), il numero di interrogazioni e, per ogni interrogazione, il voto assegnato.
Uno studente viene promosso se la media dei voti è maggiore o uguale a 6 (e in questo caso il programma stampa la media), altrimenti viene bocciato.
Al termine della sessione il programma stampa la percentuale dei promossi, la media dei promossi e le stesse informazioni divise per sessi.
Se la media complessiva dei voti degli studenti maschi è maggiore di quella delle studentesse, stampa "I maschi sono più bravi", altrimenti stampa "Le femmine sono più brave".

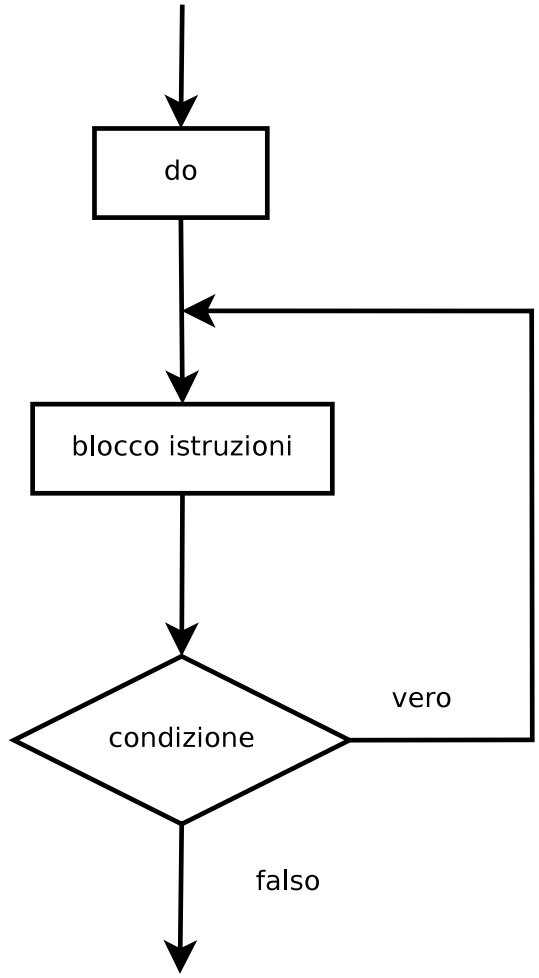
Il comando switch

```
switch x {
    case 1:
        blocco istruzioni 1;
        break;
    case 2:
        blocco istruzioni 2;
        break;
    ...
    case n:
        blocco istruzioni n;
        break;
    default:
        blocco istruzioni;
        break;
}
```



Il comando do while

```
do {
    blocco istruzioni;
} while ( condizione );
```



`break` e `continue`

Il comando `break` interrompe un ciclo (while, for), e riprende l'esecuzione dal termine del ciclo.

Il comando `continue` interrompe l'iterazione corrente di un ciclo, e riprende dall'iterazione successiva.

Le funzioni

```
int massimo (int x, int y) {
   return x > y ? x : y;
}
```

Per definire una funzione si devono fornire il tipo di ritorno, un identificatore valido (il nome della funzione), e la lista dei parametri in input.
È possibile definire una segnatura della funzione e fornire l'implementazione del corpo della funzione in seguito.

Esercizio

Scrivere due funzioni, una che calcola il fattoriale di un numero, e una che calcola il valore di 2 elevato alla potenza del numero ricevuto in input.
Il valore in input alle due funzioni è un `unsigned int`, il tipo di ritorno delle funzioni un `unsigned long`.
Nel main inserire un ciclo `do while` che stampi i valori del fattoriale di i e di 2^i per i compreso fra 0 e 10.

Esercizio

Scrivere un programma che richiede un intero positivo. Se il numero è divisibile per 3 stampa il valore inserito e " e' divisibile per tre".
Se il resto della divisione è 1, stampa il valore inserito e " -1 e' divisibile per tre".
Se il resto della divisione è 2, stampa il valore inserito e " +1 e' divisibile per tre".
Se il numero inserito è negativo il programma termina, altrimenti chiede un nuovo intero e prosegue.
(Usare il comando `switch`.)