

# Programmazione - lezione 9

Alessandro Mazzei

[alessandro.mazzei@unito.it](mailto:alessandro.mazzei@unito.it)

Link al moodle :

[informatica.i-learn.unito.it/course/view.php?id=2982](http://informatica.i-learn.unito.it/course/view.php?id=2982)

**17/10/2024**

## Contenuti

### Strutture e funzioni

La funzione main chiama una funzione per eseguire un compito e questa funzione può eventualmente restituire un output. Una funzione solitamente prende qualcosa in input.

```
#include <stdio.h>

int quadrato(int n);

int main(void) {
    int n;
    scanf("%d", &n);
    int quadrato = quadrato(n);
    printf("Il valore inserito e' %d e il suo quadrato e' %d", n,
quadrato);
}

int quadrato(int n) {
    return n*n;
}
```

In caso di input n = 5, verrà scritto sulla console il seguente messaggio :

```
5
Il valore inserito e' 5 e il suo quadrato e' 25 (output del printf)
```

5 è valore inserito dall'utente tramite scanf. "Il valore inserito e' 5 e il suo quadrato e' 25" è l'output del printf.

Dire "int quadrato(int n)" (leggendo da sinistra verso destra) indica che :

1. La funzione restituirà un valore int.
2. Il nome della funzione è "quadrato";
3. Tra le parentesi, il/i valore/i (parametro/i per essere precisi) che dobbiamo passare alla funzione per far sì che funzioni. Questi valori (quando scritti nella definizione della funzione) si chiamano "**Parametri formali**".

Questo è un prototipo, ovvero una definizione iniziale della funzione per farla "conoscere" al main e alle altre funzioni e poi si scrive l'effettivo corpo della funzione sotto il main.

## Le funzioni e lo stack

Ogni chiamata a funzione comporta l'allocazione di un nuovo frame (record di attivazione) nella memoria stack. Anche la funzione main ha un frame nello stack.

C'è tutto un discorso sullo stack, scriverlo qui sarebbe lungo e probabilmente inutile, ti consiglio di andare a vedere le slide del prof, se ti serve posso aiutarti a leggerle, perché sono molte immagini dove ha parlato molto, c'è quindi poco testo. È importante per l'esame, gli esercizi su questa cosa (che è abbastanza semplice in realtà) valgono 8 punti su 32.

## Passaggio per valore

Quando passiamo un qualcosa ad una funzione, stiamo passando sempre un valore, un numero. Non dobbiamo pensare di star passando la variabile, ma dobbiamo capire che stiamo passando solamente il **valore** di quella variabile.

```
#include <stdio.h>

int quadrato(int n);

int main(void) {
    int n;
    scanf("%d", &n);
    int quadrato = quadrato(n);
    printf("Il valore inserito e' %d e il suo quadrato e' %d", n,
quadrato);
}

int quadrato(int n) {
    return n*n;
}
```

In questo codice ad esempio (prendiamo un input = 5), quando chiamiamo "int quadrato = quadrato(n)" noi non stiamo passando n alla funzione quadrato, ma stiamo passando 5, si

potrebbe vedere quindi come "int quadrato = quadrato(5)".

## **Passaggio per riferimento**

Se dovessimo aver bisogno di passare una intera variabile alla funzione, usiamo quindi il passaggio per riferimento. Questo si fa utilizzando i puntatori Non è stato spiegato ancora altro dal professore riguardo questo argomento (slide 65).