


# Laboratorio 1

---

Sara Montagna



13 Marzo 2020

## 1 UNA INTRODUZIONE OPERATIVA A CODE::BLOCKS

Aprire Code::Blocks, l'ambiente di sviluppo che useremo nelle lezioni di laboratorio in questo corso. Trovate sul Desktop un collegamento con icona 

**APRIRE UN PROGRAMMA** Per aprire un file già presente del File System selezionare File → Open (oppure Ctrl-O) e navigare il File System posizionandosi nella cartella in cui il file è stato salvato. Selezionare il file e poi premere il bottone Open.

**CREARE E SALVARE UN NUOVO PROGRAMMA** Per creare un nuovo file selezionare File → New → Empty file. Per salvarlo File → Save file, scegliendo la cartella sul File System in cui si vuole salvare il file e specificando poi il nome del file con estensione “.c”, seguendo le convenzioni: *(i)* prima lettera minuscola, *(ii)* non lasciare spazi se il nome è composto, ma utilizzare eventualmente un \_ come separatore —e.g. `nome_file.c`

**COMPILARE ED ESEGUIRE UN NUOVO PROGRAMMA** Per compilare il file sorgente premere sulla barra degli strumenti il bottone Build (icona con rotella gialla ) e per eseguire il bottone Run (icona con freccia verde ). A seguito della Build – nella finestra Log & others – selezionare Build messages, dove vengono riportati eventuali errori (sintattici o semantici, lo vedremo tra qualche lezione) presenti nel programma. Risolverli (e ricompilare) prima di eseguire.

## 2 ANALISI DEI PRIMI PROGRAMMI IN C

UTILIZZO DELLA FUNZIONE PER LA STAMPA A VIDEO Si apra il file `helloworld.c`, che riporta l'esempio visto a lezione, dove il `main()` utilizza la funzione `printf()` per stampare a video la stringa "Hello, world!", quindi vada a capo. Si compili ed esegua il programma.

CALCOLO DELL'AREA DI UN TRIANGOLO Analizzare il sorgente `area_triangolo.c`, compilarlo ed eseguirlo, verificando che il risultato dell'esecuzione sia la stampa attesa.

## 3 CREAZIONE DEI PRIMI PROGRAMMI IN C

CALCOLO DELL'AREA DI UN RETTANGOLO A partire dal sorgente `area_triangolo.c`, creare un programma che calcola l'area di un rettangolo:

1. Creare un nuovo sorgente `area_rettangolo.c`
2. Copiarci internamente il contenuto del sorgente `area_triangolo.c`
3. Modificare opportunamente il metodo `main()` affinché computi l'area di un rettangolo
4. Fare la stessa cosa dei punti 1-3 per computare l'area del cerchio all'interno di un sorgente `area_cerchio.c`. Nota: si consiglia di definire una variabile chiamata `PI_GRECO` di valore iniziale 3.14

## 4 FACCIAMO ANCORA QUALCHE ESERCIZIO

ESERCIZIO 1 Si scriva un programma che legge da tastiera due valori interi  $a$  e  $b$  e visualizzi la loro somma. Si faccia riferimento al file `pitagora.c` visto a lezione (e allegato a questa esercitazione) per vedere come si fa a leggere un valore intero inserito dall'utente mediante la funzione `scanf()`

ESERCIZIO 2 Si scriva un programma in grado di stampare il risultato delle quattro operazioni (somma, sottrazione, moltiplicazione e divisione) tra due interi  $a$  e  $b$  inseriti da tastiera. Dopo che sono stati inseriti i due numeri, il programma dovrà visualizzare il risultato delle quattro operazioni. Cosa succede tentando di calcolare il rapporto  $a/b$  se  $b = 0$ ?

ESERCIZIO 3 Si scriva un programma che legga da tastiera un valore intero e visualizzi il valore intero precedente e il successivo.

ESERCIZIO 4 Si scriva un programma che legga da tastiera due valori reali e visualizzi la loro media aritmetica. SUGGERIMENTO: per acquisire da tastiera un valore reale si usi l'istruzione `scanf("%f", ...)`;

ESERCIZIO 5 Si scriva un programma che legga da tastiera un numero e stampi un messaggio che indichi se tale numero sia positivo oppure negativo.

## 5 DALL'ESAME TEORICO

ESERCIZIO 1 Si consideri la rappresentazione di interi in complemento a due usando  $N = 7$  bit. Allora:

**V - F** Il massimo valore rappresentabile è 63

**V - F** Il minimo valore rappresentabile è  $-64$

**V - F** Il valore decimale  $-12$  si codifica come  $0001100_2$

**V - F** Il numero  $1000111_2$  rappresenta un numero positivo.

ESERCIZIO 2 Nelle domande seguenti, le variabili  $A$  e  $B$  rappresentano valori di input, e  $R$  rappresenta il risultato di una espressione booleana. Indichiamo con 1 il valore booleano *true*, e con 0 il valore booleano *false*. Allora:

**V - F** Esistono valori di  $A$  e  $B$  per cui l'espressione  $R = ((A \text{ AND } B) \text{ OR } ((\text{NOT } A) \text{ AND } B))$  assume valore 1 (*vero*)

**V - F** Le espressioni  $(\text{NOT } (A \text{ OR } B))$  e  $((\text{NOT } A) \text{ AND } (\text{NOT } B))$  sono tra di loro equivalenti, nel senso che le loro tabelle di verità sono uguali

**V - F** Se  $A = 1$  e  $B = 0$ , allora l'espressione  $R = ((A \text{ OR } (\text{NOT } B)) \text{ OR } A)$  ha valore 1

**V - F** L'espressione  $R = ((A \text{ OR } (\text{NOT } B)) \text{ OR } A)$  assume sempre valore 1, qualsiasi siano i valori di  $A$  e  $B$ .