# Lab Info

A.A. 2024/25

POLITECNICO
MILANO 1863

# Obiettivi

Comprendere e risolvere problemi con gli strumenti dell'informatica

→Implementare programmi in linguaggio C

Lo scopo del laboratorio è mettere in pratica quello che avete imparato.

# Introduzione al linguaggio C

Struttura di un programma C

- Librerie: Includono funzionalità comuni a molti programmi e si possono dunque importare in maniera standard.

- main(): è la funzione di ingresso per ogni programma C.
- Deve esserci obbligatoriamente. Il codice al suo interno è il punto di partenza per il programma.

- Commenti: le linee precedute da "//" non sono eseguite. È molto importante usare commenti per non perdere traccia di quello che si sta facendo.

```
#include <stdio.h>
#include <stdlib.h>

void main()
{
    //DO SOMETHING
}
```

# Compilazione

Compilare

In assenza di errori, viene prodotto un file.exe nella stessa cartella in cui è stato salvato il codice sorgente.

In presenza di errori, il codice non viene compilato e vengono mostrati dei messaggi relativi agli errori rilevati.

Correggere gli errori

I messaggi di errore sono importanti!

Correggere sempre gli errori in ordine, poiché alcuni errori potrebbero dipendere da errori precedenti.

# Errori comuni

Errori di compilazione

Controllare di aver salvato sempre i file in formato `.c` e non `.c++`.

Controllare di aver salvato nella cartella remota e non nel disco locale.

Controllare di aver chiuso altre istanze del vostro programma.

# Variabili

## Salvare un valore

Le variabili sono contenitori per i nostri dati. Al loro interno possiamo salvare informazione di vario tipo.

## Tipi di variabile

In C, le variabili sono *tipizzate*. Ciò significa che il tipo di informazione contenuto nella variabile deve essere specificato durante la definizione della variabile.

```
int n;
float x;
char c;
```

Variabile intera

`-1, 7, 12`

Variabile a virgola mobile

`-3.2, 7.0`

Carattere alfanumerico

`'c', 'F', '2'`

# Visibilità delle variabili

## Scope

Il nome delle variabili le identifica all'interno del *blocco di codice* in cui sono state definite.

Ciò significa che possiamo avere più variabili diverse con lo stesso nome, se ridefinite in diversi blocchi.

Lo *scope* più ampio è quello delle variabili globali.

```
int i 3;

void main()
{
    int i=2;

    //IL CODICE QUI VEDE LA i-2
}
```

```
int VARIABILE_GLOBALE_INTERA;

void main()
{
    char VARIABILE_LOCALE_CARATTERE;

}
```

# Commenti

Altri modi per commentare codice

---

```
printf("hi"); //This is an in-line comment
/*
This is a multi-line comment.
All this text won't be executed.
This goes on until you write:
*/
```

# printf

La funzione **`printf`** ci permette di stampare a schermo e comunicare con l'utente.

```c
printf("hello world!");
```

## Chiamare una funzione

Per chiamare una funzione, la sintassi è:

```c
nome_funzione(parametro_1, parametro_2, parametro_n);
```

⚠️

In C, le istruzioni terminano con un punto e virgola. Dimenticarselo è un errore molto comune, quindi se qualcosa non va, è tra le prime cose da guardare.

# Stampare variabili con printf (1)

La stringa di formato

La funzione **printf** ci permette anche di stampare i valori presenti nelle variabili.

Per farlo, si scrive una *format string*, che contiene delle sequenze speciali che indicano quali valori prendere e dove mostrarli:

`%d` numero intero
`%c` carattere alfanumerico
`%f` numero decimale

```
int i=3;

void main()
{
    printf("The value is: %d", i);
}
```

La sequenza speciale `%d` indica che qui dovrà essere stampato un intero

Il nome della variabile che vogliamo stampare è poi inserito come parametro nella chiamata a `printf`.

# Stampare variabili con printf (2)

## La stringa di formato

È anche possibile stampare più di un valore nella stessa chiamata a `printf`. In questo caso, aggiungiamo un parametro per ogni valore che vogliamo stampare.

⚠️ Le sequenze speciali indicano alla printf come interpretare i dati in memoria, è quindi importante che siano coerenti con il tipo effettivo delle variabili da stampare.

*Cosa succede se chiedo alla printf di stampare un char utilizzando %d?*

PROVATE VOI!!

```c
int i = 3;
float f = 2.1;

printf("The integer is: %d, the float is: %f", i, f);
```

# Operatori aritmetici

## Espressioni aritmetiche nel codice

Le normali operazioni aritmetiche usano gli operatori classici: `+ - * /`

```
int a = 1;
int b = 2;
int c = a + b + 3;
```

## Modulo

L'operatore `%` ritorna il resto divisione intera tra gli operandi.

```
12%5 = 2
```

## Operatori di assegnamento

Operatori speciali che eseguono una operazione aritmetica e assegnano il risultato al primo dei due operandi.

```
a = 2;
b = 5;

a += b;

// a = 7
```

`+=` `-=`
`*=` `/=`

**Syntactic sugar**

```
a += 1;   ↔   a++;
a -= 1;       a--;
```

# Operazioni su interi

### Implicit casting

Se il tipo degli operandi e del risultato non sono compatibili, questi vengono modificati, se possibile, per permettere l'operazione.

```
float a = 2.7;
int b = 1;

int c = a + b;

// c = 3
```

⚠ La conversione da **`float`** a **`int`** viene sempre effettuata troncando la parte decimale, e non per arrotondamento all'intero più vicino.

# Ricevere input dall'utente

## Ricevere input dall'utente - **scanf**

La funzione **scanf** ci permette di ricevere dati dall'utente e salvarli nelle variabili.

La stringa di formato

Come per la **printf**, una stringa di formato viene usata per determinare come ricevere i dati.

Il puntatore alla variabile

L'operatore unario **&** ritorna l'indirizzo di memoria della variabile operando.

Per indicare alla funzione dove salvare l'input dell'utente, durante la chiamata alla **scanf**,

```c
int a;

scanf("%d", &a);
```

**&a** indica che vogliamo salvare il valore nella cella di memoria dove è situata la variabile **a**.

La scanf può salvare anche più di un valore alla volta, ma tipicamente questo non viene fatto, perché richiede che l'utente rispetti il formato specificato nella stringa di formato.

```c
int a;
float b;

scanf("%d-%f", &a, &b);
```

Nell'esempio, l'utente dovrebbe inserire una stringa del tipo:

```
5-2.3
```

# Esercizio 0

Scrivere un programma che stampa a schermo le seguenti informazioni:

```
data di oggi
nome, cognome,
matricola
```

```
#include <stdio.h>
#include <stdlib.h>

void main()
{

      ?

}
```

ℹ️ Per andare a capo, si scrive: **`\n`**

⚠️ Il backslash **`\`** è un carattere di *escape*, ed è usato per indicare i caratteri speciali come **`\n`** (new line).

Per poter scrivere **`\`**, si può scrivere **`\\`**.

Un altro esempio di carattere speciale è il tab, che si scrive con **`\t`**.

# Esercizio 1

Scrivere un programma che, date due variabili *char,* stampa una parola composta dalle due lettere, ripetute due volte.

```
char_1 = ‘a’ char_2 = ‘b’
→      stampa: ‘abab’
```

```
void main()
{
    char char_1 = 'a'
    char char_2 = 'b'

    ?

}
```

# Esercizio 2

Scrivere un programma che chiede all’utente due numeri interi e che ne stampi la somma.

# Esercizio 3

Scrivere un programma che chiede in input all'utente  la prima lettera del proprio nome e  la  prima  lettera del proprio cognome, e che le restituisce nell'ordine inverso (prima cognome poi nome).

```
fflush(stdin);
```

# Esercizio 4

Scrivere un programma che chiede in input all'utente due numeri interi e che stampa risultato e resto della divisione tra i due numeri.

# Esercizio 5

Scrivere un programma che dato un numero rappresentante un prezzo p ed un valore s tra 0 e 100, ritorni il prezzo scontato del s%.

# Esercizio 6

Scrivere un programma che, acquisiti tre voti universitari e relativi valori di CFU, restituisca la media pesata.

# Esercizio 7

Scrivere un programma che chiede in input all'utente un numero r e che restituisca:

1. Il perimetro della circonferenza di raggio r
2. L'area del cerchio di raggio r
3. La superficie della sfera di raggio r
4. Il volume della sfera di raggio r

```
#include <math.h>

pow(a, b); //a^b
```

Wikipedia formule:

https://it.wikipedia.org/wiki/Sfera

# Esercizio 8

Scrivere un programma che acquisiti tre valori rappresentanti i coefficienti di una equazione di secondo grado, ritorni le radici reali dell'equazione.

$$x = \frac{-b \pm \sqrt{b^2 - 4\,a\,c}}{2\,a}$$

# Varianti

Rifare esercizi 4,7,8 con funzioni

# Soluzioni