



UNIVERSITÀ  
DI TORINO

---

# Laboratorio di Programmazione I LAB-C-T1-T2

---

## Lezione n. 2: Operatori e Tipi

Alessandro Mazzei

Slides Credit: Elvio Amparone

# Comunicazioni

---

## Problemi di accesso ai PC del laboratorio Turing:

- Gli/Le studenti/studentesse che hanno riscontrato dei problemi con il login di Ateneo per accedere alle macchine del Laboratorio Turing possono:
  1. Provare a cambiare la password e riprovare
  2. Se continua a NON funzionare, fare richiesta per avere un login temporaneo che consente l'accesso.
- Se qualcuna/o non è riuscito a installare il compilatore C e un Editor sul proprio PC/notebook, è invitato a farlo ORA.

# GCC su Tablet



## iSH

Why iSH? Because it emulates Alpine Linux and can tap into the massive Alpine package repository...

```
apk add build-base clang
```

<https://jsmp.me/2020/05/05/c-development-on-ios>



## Termux

Termux is an Android terminal emulator and Linux environment app that works directly with no rooting or setup required.

```
apt update  
apt install clang
```

<https://termux.dev/en/>

# Outline

---

- Esercizio live
- Presentazione Esercizi

# Esercizi

---

1. Operatori aritmetici e Cast
2. Operatori aritmetici
3. Ordine di valutazione degli operatori
4. Circonferenza
5. Media di 3 numeri
6. Errori in C
7. Il mercato
8. Uno scambio più “complesso”
9. Dimensione dei tipi
10. Documentazione

# Ex. 1 - Operatori aritmetici e Cast

Provare il programma `aritmetica.c` leggendo attentamente il codice prima di eseguirlo.

**NOTA:** per compilare usiamo il comando `gcc`:

- Unix: `gcc aritmetica.c -o aritmetica`
- Windows: `gcc aritmetica.c -o aritmetica.exe`

## Ex. 2 - Operazioni su interi

Scrivere un semplice programma **operazioni\_interi.c** che:

1. chieda all'utente di inserire due numeri interi, che chiamiamo **n1** ed **n2**;
2. calcoli la somma nella variabile **n12**
3. stampi a video il valore della variabile **n12**
4. stampi a video:
  - a. il prodotto di **n1** con **n2**
  - b. la divisione intera di **n1 / n2**
  - c. la divisione reale di **n1 / n2**
  - d. il resto della divisione intera tra **n1** ed **n2**

**NOTA:** Per chiedere all'utente i due interi **n1** ed **n2**, usare la funzione **scanf** come visto a lezione.

## Ex. 3 - Ordine di valutazione

Quali risultati otteniamo in C da queste espressioni:

1. `int x = 7 + 3 * 6 / 2 - 1;`
2. `int x = 2 % 2 + 2 * 2 - 2 / 2;`
3. `int x = (3 * 9 * (3 + (9 * 3 / (3)))));`

Quali di queste espressioni corrisponde ad:  $y = ax^3 + 7$  ?

1. `int y = a * x * x * x + 7;`
2. `int y = a * x * x * (x + 7);`
3. `int y = (a * x) * x * (x + 7);`
4. `int y = (a * x) * x * x + 7;`
5. `int y = a * (x * x * x) + 7;`
6. `int y = a * x * (x * x + 7);`



## Ex. 4 - Circonferenza

---

Aprire il programma **circonferenza.c** e completare la parte per calcolare la circonferenza.

Si ricorda che:

- la circonferenza di un cerchio di raggio  $r$  è pari a  $2\pi r$
- l'area di un cerchio di raggio  $r$  è pari a  $2\pi r^2$

## Ex. 5 - Media di tre numeri

---




Leggere il codice della classe **media.c**, e sperimentare input differenti seguendo gli esercizi proposti.

Rispondere alle domande indicate nei commenti del codice.

# Ex. 6 - Errori in C

Quali errori vedete in questo codice?

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

```
int main() {  
    int x, y;  
    printf("Inserisci due interi : ");  
    scanf("%d %d", &x, y)   
     z = x + y;  
    printf("La somma %d + %d vale ", x, y);  
    printf("%d\n",  &z);  
}
```

Leggere il codice **errori.c** e (senza compilare) individuare quali errori sono presenti. **Riuscite a trovarli tutti e 8?**

## Ex. 7 - Mercato

Scrivere un file `mercato.c` per calcolare il prezzo di quanto acquistiamo al supermercato:

- 3 pacchi di orecchiette (prezzo di un pacco di orecchiette euro 1.50)
- 1 kg e 1/2 di pomodori (prezzo al kg dei pomodori: euro 2.50)

Scrivere la funzione `main` in modo che calcoli e visualizzi a video la spesa totale. Suggestimenti:

- Modellare il problema utilizzando delle variabili opportunamente dichiarate con tipi adeguati.
- Utilizzare una variabile anche per rappresentare il totale.
- Indentare e commentare opportunamente il programma.

Infine, modificare il programma per far inserire le quantità degli articoli alimentari dall'utente da tastiera (anziché fissare nel codice del programma i valori delle variabili).

## Ex. 8 - Scambio di variabili

Leggere, compilare e provare il programma **scambio.c** (simile a quello che avete visto a lezione)

Scrivere un programma **scambio4.c** che prende 4 variabili intere **v1**, **v2**, **v3** e **v4** (opportunamente inizializzate con dei valori iniziali a scelta), e le scambia in modo tale che:

- **v1** prende il valore di **v2**
- **v2** prende il valore di **v3**
- **v3** prende il valore di **v4**
- **v4** prende il valore di **v1**

Stampare i valori delle quattro variabili prima e dopo lo scambio.

# Tabella dei tipi e specificatori di conversione

Dimensioni e specificatori dei tipi elementari con segno:

Tipo	Uso	CPU 32 bit		CPU 64 bit		specificatori di conversione
		Dimensione [Byte / bit]	Intervallo	Dimensione [Byte / bit]	Intervallo	
char	ASCII	1 / 8	-128, 127	1 / 8	-128, 127	%c
short short int	numeri interi	2 / 16	$-2^{15}$ , $+2^{15}-1$	2 / 16	$-2^{15}$ , $+2^{15}-1$	%hd
int signed int		4 / 32	$-2^{31}$ , $+2^{31}-1$	4 / 32	$-2^{31}$ , $+2^{31}-1$	%d
long long int		4 / 32	$-2^{31}$ , $+2^{31}-1$	8 / 64	$-2^{63}$ , $+2^{63}-1$	%ld
long long long long int		8 / 64	$-2^{63}$ , $+2^{63}-1$	8 / 64	$-2^{63}$ , $+2^{63}-1$	%lld
float	numeri reali (con virgola)	4 / 32	$10^{-38}$ , $10^{+38}$	4 / 32	$10^{-38}$ , $10^{+38}$	%f
double		8 / 64	$10^{-308}$ , $10^{+308}$	8 / 64	$10^{-308}$ , $10^{+308}$	%lf

**NOTA:** non serve imparare a memoria tutta la tabella!!

# Tabella dei tipi e specificatori di conversione

Dimensioni e specificatori dei tipi elementari **senza** segno:

Tipo	Uso	CPU 32 bit		CPU 64 bit		specificatori di conversione
		Dimensione [Byte / bit]	Intervallo	Dimensione [Byte / bit]	Intervallo	
unsigned char	bytes	1 / 8	<b>0, 255</b>	1 / 8	<b>0, 255</b>	%hhu
unsigned short unsigned short int	numeri naturali	2 / 16	<b>0, +2<sup>16</sup>-1</b>	2 / 16	<b>0, +2<sup>16</sup>-1</b>	%hu
unsigned int unsigned signed int		4 / 32	<b>0, +2<sup>32</sup>-1</b>	4 / 32	<b>0, +2<sup>32</sup>-1</b>	%u
unsigned long unsigned long int		4 / 32	<b>0, +2<sup>32</sup>-1</b>	8 / 64	<b>0, +2<sup>64</sup>-1</b>	%lu
unsigned long long unsigned long long int		8 / 64	<b>0, +2<sup>64</sup>-1</b>	8 / 64	<b>0, +2<sup>64</sup>-1</b>	%llu
float	numeri reali (con virgola)	4 / 32	10 <sup>-38</sup> , 10 <sup>+38</sup>	4 / 32	10 <sup>-38</sup> , 10 <sup>+38</sup>	%f
double		8 / 64	10 <sup>-308</sup> , 10 <sup>+308</sup>	8 / 64	10 <sup>-308</sup> , 10 <sup>+308</sup>	%lf

**NOTA:** non serve imparare a memoria tutta la tabella!!

## Ex. 9 - Dimensioni dei tipi

---

Leggere, compilare e provare il programma **dim\_tipi.c**

Quale caratteristica hanno le dimensioni dei tipi puntatore?  
Perché?



# Ex. 10 - Documentazione

Potete accedere ad una versione offline della documentazione del *linguaggio* e della *libreria* del C.

1. Accedere a <https://en.cppreference.com/w/>
2. Andare su “Offline version” (in basso), poi scaricare la versione più recente (*Unofficial Release*)
3. Aprire la pagina [html-book/reference/en/c.html](https://en.cppreference.com/html-book/reference/en/c.html)
4. Cercare nella sezione degli **Headers** la documentazione di `<stdio.h>`, e poi leggere la documentazione completa di `printf`

**NOTA:** la documentazione di riferimento è molto più elaborata di quello visto finora, ma vi serve per capire come cercare le informazioni che vi servono.