

modulo di Laboratorio del corso di Programmazione 1  
prof. Marco Roveri [marco.roveri@unitn.it](mailto:marco.roveri@unitn.it)

# Programmazione 1 - LAB

## 02 - Esercitazione

Martina Battisti [martina.battisti-1@unitn.it](mailto:martina.battisti-1@unitn.it)

Giovanna Varni [giovanna.varni@unitn.it](mailto:giovanna.varni@unitn.it)

Andrea E. Naimoli [andrea.naimoli@unitn.it](mailto:andrea.naimoli@unitn.it)

# Feedback Programmazione I

<https://forms.gle/wPqgTQZrDPoCAmue9>

Le risposte inviate attraverso questo form vengono registrate in modo anonimo. Nessuna vostra informazione personale verrà conservata. Il sondaggio è accessibile soltanto se in possesso di credenziali unitn.

# Moodle

<https://didatticaonline.unitn.it/dol/course/view.php?id=39241>

# Nelle puntate precedenti

- **Variabili**

```
int i; float c; double z = 3.0; char c = 'a';  
const float RADICE_2 = 1.41421;
```

- **Stream input/output**

```
#include <iostream>  
  
using namespace std;  
char a; cin >> a; cout << a << endl;
```

- **Operazioni Matematiche**

```
int x = 3 + 2; int y = x - 2;  
int w = 3 / 2; int k = 3 * 2; int x = 3 % 2;
```

# Nelle puntate precedenti

```
1.  #include <iostream>
2.  using namespace std;
3.
4.  int main()
5.  {
6.      char carattere;
7.
8.      cout << "Inserisci un carattere: ";
9.      cin >> carattere;
10.     cout << "Il carattere inserito è: " << carattere << endl;
11.
12.     return 0;
13. }
```

# 00 - Let's begin!

- **Operatori misti aritmetica/assegnazione**

```
x += y; x -= y; x *= y; x /= y; x %= y;
```

- **Operatori di (pre/post)-incremento/decremento unitario**

```
x++; ++x; x--; --x;
```

```
int valore = 6;
```

```
int i = valore++; // i = 6, valore = 7
```

```
int j = ++valore; // j = 8, valore = 8
```

- **Operazioni booleane (&&, ||, >, <, >=, <=, !, ==, !=)**

```
bool maggiore = 5 > 6;
```

```
bool and = true && false;
```

# 00 - Let's begin!

Per creare un file vuoto, modificarlo ed eseguirlo, ricordate la sequenza di comandi base da lanciare al terminale

```
touch example.cc  
code example.cc  
g++ -o esempio.out example.cc  
./esempio.out
```

# 1 - Tipi Fondamentali

**Scrivere un programma che stampi a video una lista di valori, almeno uno per ogni tipo base, ed espressioni**

- **interi:** `int`, `short`, `long`, `long long`
- **booleani:** `bool`
- **carattere:** `char`
- **reali:** `float`, `double`, `long double`

suggerimento: provare a stampare espressioni di vario genere anche mischiando i tipi (esempio: somma di `int` e `bool` o `char`)



## 2 - Maiuscole/Minuscole

**(recap) dato in input un carattere maiuscolo, ritorna lo stesso carattere minuscolo**

**VARIANTE: risolvere anche il problema inverso (da minuscolo a maiuscolo) MA con un unico codice (unico programma e non due distinti) e SOLO con quanto fino ad ora visto (quindi NIENTE costrutti condizionali)**

**EXTRA: mantenendo i vincoli aggiungere la caratteristica che se il carattere non è “valido” è mostrato un punto interrogativo (“?”) come output**

# 3 - Secondi da mezzanotte (in output)

**Scrivere un programma che, dati tre input da tastiera (ore, minuti e secondi), li memorizzi in tre variabili distinte e calcoli i secondi da mezzanotte.**

**(mezzanotte => 0; 23:59:59 => 86399)**

## 4 - Secondi da mezzanotte (in input)

**Scrivere un programma che, dati in input i secondi da mezzanotte, li salvi in una variabile e ritorni a video l'equivalente orario in ore, minuti e secondi.**

**(mezzanotte => 0; 86399 => 23:59:59)**

## 5 - Secondi da mezzanotte (in input) - II

**Riscrivere il programma dell'esercizio precedente senza usare variabili per calcoli intermedi (usando al massimo quattro variabili per input, e ore, minuti e secondi calcolati).**