

# Esercitazione 3

puntatori, reference, allocazione dinamica e vettori

22-23-24 ottobre 2018

Corso di Laboratorio 1A, A.A. 18-19

Laurea triennale in Fisica, Università di Genova

Lo scopo di questa esercitazione e' apprendere le nozioni di base relative all'utilizzo di puntatori e referenze, allocazione dinamica della memoria e utilizzo dei vector delle STL.

I primi due esercizi svolti mostrano delle funzionalita' di base di puntatori ed allocazione dinamica della memoria.

Il terzo esercizio, da svolgere in laboratorio, chiede di calcolare delle grandezze caratteristiche di una serie di dati a partire da un file di input, e di scrivere il risultato su un file di output.

Il quarto esercizio, da svolgere a casa, richiede di calcolare delle quantita' a partire da un set di dati da leggere da un file.

- Svolgete gli esercizi secondo la traccia riportata nel resto di questo documento
- Seguite le istruzioni riportate [qui](#) per la consegna degli esercizi di laboratorio
- **IMPORTANTE:** la consegna deve essere effettuata da entrambi i componenti del gruppo

# Esercizio 1

Esercizio svolto, dimostrazione delle proprietà dei puntatori

```
#include <iostream>
using namespace std;
int main(){

    double a = 8.0;
    double b = 4.0;

    double* ptr = NULL; // puntatore inizializzato a zero (nessuna zona
di memoria)
    cout << "ptr = " << ptr << endl;

    ptr = &a; // Faccio in modo che p punti alla locazione di memoria di
a
    cout << "Indirizzo di a: " << ptr << endl;
    cout << "Valore di a: " << *ptr << endl;

    b = *ptr;      // Assegno a b il valore puntato da p

    *ptr = 5.0;    // Assegno alla casella di memoria
                  // con indirizzo p il valore 5.0

    cout << "Valore di a: " << a << endl;
    cout << "Valore di b: " << b << endl;

    double v[5]={0.0, 1.0, 2.0, 3.0, 4.0};

    ptr = &v[0];
    cout << "Indirizzo di v[0]: " << ptr << endl;

    ptr = v;
    cout << "Indirizzo di v   : " << ptr << endl;

    return 0;
}
```

Materiale da consegnare: [esercizio1.cpp](#)

## Esercizio 2

Esercizio svolto, allocazione dinamica di un vettore. Questo programma dimostra l'utilizzo degli operatori new e delete.

```
#include <iostream>
using namespace std;

int main(){

    int n;
    cout << "Inserisci la dimensione del vettore " << endl;
    cin >> n;

    double *p = new double[n];

    cout << "Inserisci le " << n << " componenti del vettore "<< endl;
    for (int i=0; i < n;i++){
        cin >> p[i];
    }
    cout << "Puntatore alla prima componente " << p << endl;
    cout << "Componenti del vettore ";

    for (int i=0;i < n;i++){
        cout << " " << p[i];
    }
    cout << endl;

    delete[] p;
    return 0;
}
```

Materiale da consegnare: [esercizio2.cpp](#)

## Esercizio 3

MOLTO utile per Laboratorio 1-B. Creare un file di dati con n numeri reali. Creare un programma C++ che:

- Legga da terminale il nome del file da leggere.
- Legga dal file tutti i numeri reali salvandoli in un array allocato dinamicamente (nel file il primo dati è il numero di valori)
- Calcoli:

- media:  $\langle x \rangle = \frac{1}{N} \sum_{i=1}^N x_i$

- media dei quadrati:  $\langle x^2 \rangle = \frac{1}{N} \sum_{i=1}^N x_i^2$

- varianza:  $\sigma^2 = \langle x^2 \rangle - \langle x \rangle^2$

- varianza empirica:  $s_N^2 = \sigma^2 \frac{N}{N-1}$

- deviazione standard:  $\sigma$

- deviazione standard adattata:  $s_N$

- errore standard:  $\frac{s_N}{\sqrt{N}}$

Verificate il vostro programma con il file di dati [esempio.dat](#) che fornisce i risultati:

Media 10.0083

Varianza 5.1059

Varianza Empirica 5.28197

Deviazione Standard 2.25962

Deviazione Standard Adattata 2.29825

Errore Standard 0.419602

Salvate, infine, su un file di output (utilizzando un oggetto di tipo ofstream) i risultati del calcolo

Materiale da consegnare: **esercizio3.cpp**

Ripetere lo stesso esercizio leggendo il nome dei file di input e output da riga di comando, e usando un vector C++ invece di un array allocato dinamicamente

Materiale da consegnare: **esercizio3bis.cpp**

## Esercizio per casa

Leggere da file dal file [temperature.dat](#) i valori di temperatura registrati ogni giorno per N giorni (5 temperature per giorno).

```
N
temp1_day1 ... temp5_day1
..
..
temp1_dayN ... temp5_dayN
```

Identificare il giorno (o i giorni) in cui si è verificata la temperatura massima.

Potete risolvere il problema:

- senza usare vettori
- usando un vettore per le temperature massime

Materiale da consegnare: **compito3.cpp** (tramite Aulaweb, entro lunedì 5/11)