

Esercitazione 1

Primi programmi, input/output e flow control

8-9-10 ottobre 2018

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

Laurea triennale in Fisica, Università di Genova

Lo scopo di questa prima esercitazione è prendere dimestichezza con i principali strumenti utili allo sviluppo del software (terminale, editor di testo, browser, compilatore, etc) e familiarizzare con alcuni strumenti base della programmazione C++: input/output da terminale e elementi per il controllo di flusso di un programma

- 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

Primi passi

La prima cosa da fare in laboratorio è eseguire l'accesso con le credenziali che vi sono state fornite ed eseguire queste operazioni:

1. Aprite il web browser presente sul computer (Firefox) e raggiungete questo documento (<https://tinyurl.com/yagnaasq>), in modo da avere queste istruzioni
2. Aprite un terminale ed eseguite le seguenti operazioni:
 - a. cambiate la password:

```
bash-prompt > chgpwd [invio]  
old password [invio]  
new password [invio]  
new password [invio]
```

- b. verificate in quale directory vi trovate, e listate il suo contenuto

```
bash-prompt > pwd  
bash-prompt > ls
```

- c. create una directory per contenere le esercitazioni di Lab1A, verificate di averla creata, e al suo interno create una directory per la prima esercitazione

```
bash-prompt > mkdir Lab1A  
bash-prompt > ls  
bash-prompt > cd Lab1A  
bash-prompt > mkdir Esercitazione1
```

3. Fate logout e ripetete i punti precedenti utilizzando l'account dell'altro membro del gruppo

Materiale da consegnare: nulla

L'editor: emacs

L'editor di testo è lo strumento utilizzato per scrivere il codice sorgente.

Seguite questi passaggi per ottenere il file sorgente del vostro programma:

1. Aprite l'editor emacs utilizzando il menu grafico
2. "Visitate" un nuovo file mediante il menu "File->Visit new file"
3. Salvate il nuovo file (ancora vuoto) nella directory che avete creato in precedenza:
~/Lab1A/Esercitazione1/esercizio1.cpp
4. Copiate nel file il contenuto dell'esercizio 1 che trovate più in basso
5. Selezionare dal menu "File->Save" per salvare il file
6. Chiudere l'editor
7. Da un terminale, controllate con il comando **ls** che il file sia stato effettivamente creato

Maggiori informazioni su emacs sono disponibili ad esempio qui:

<http://www.gnu.org/software/emacs/tour/>

Materiale da consegnare: nulla

Il compilatore: g++

Il compilatore è lo strumento che converte il codice sorgente in una sequenza di istruzioni in codice binario eseguibile dalla CPU.

Da terminale potete compilare il file sorgente `esercizio1.cpp` con il seguente comando:

```
bash-prompt > g++ -o esercizio1 esercizio1.cpp
```

Se la compilazione avverrà con successo, nella directory comparirà un nuovo file eseguibile: `esercizio1`. Sempre da terminale, verificate con il comando `ls -l` che effettivamente il nuovo file abbia i permessi di esecuzione.

Approfondimento

La sintassi del comando `g++` è la seguente:

```
g++ -o <file_eseguibile> <file_sorgente>
```

- Il file sorgente deve esistere e deve avere un'estensione `.cpp`, `.cxx` o `.cc`
- Il file eseguibile può avere un nome diverso da quello del file sorgente
- Nella riga di comando, il nome del file eseguibile deve seguire l'opzione `-o`, ma l'ordine tra "`<file_sorgente>`" e "`-o <file_eseguibile>`" è irrilevante

Ad esempio il comando seguente è del tutto lecito:

```
bash-prompt > g++ esercizio1.cpp -o helloworld
```

Materiale da consegnare: nulla

Esecuzione del programma

Una volta compilato, il programma è pronto per essere eseguito. Da terminale è sufficiente lanciare il seguente comando:

```
bash-prompt > ./esercizio1
```

Notate che il nome del file eseguibile è preceduto dai caratteri “./”, che stanno ad indicare la directory in cui ci si trova (**pwd**). Lo stesso risultato può essere ottenuto anche utilizzando un path assoluto, o lanciando il programma da un'altra directory.

Esempi

Esecuzione dalla directory in cui si trova l'eseguibile

```
bash-prompt> ls  
esercizio1 esercizio1.cpp  
bash-prompt> pwd  
/home/sergio/Lab1A/Esercitazione1  
bash-prompt> ./esercizio1  
Hello world!
```

Esecuzione specificando il path assoluto

```
bash-prompt> /home/sergio/Lab1A/Esercitazione1/esercizio1  
Hello world!
```

Esecuzione specificando il path assoluto, ma tramite il carattere speciale “~”, che vuol dire “la home directory” (nel caso dell'esempio /home/sergio)

```
bash-prompt> ~/Lab1A/Esercitazione1/esercizio1  
Hello world!
```

Esecuzione da un'altra directory, usando un path relativo

```
bash-prompt> cd ..  
bash-prompt> pwd  
/home/sergio/Lab1A  
bash-prompt> ./Esercitazione1/esercizio1  
Hello world!
```

Materiale da consegnare: nulla

Esercizio 1

Questo codice sorgente rappresenta uno dei programmi più semplici che è possibile scrivere. Utilizzando le istruzioni presenti nelle parti precedenti, create un file che abbia il seguente contenuto, provate a compilarlo ed infine eseguite il programma ottenuto.

```
#include <iostream>
using namespace std;
int main(){
    cout << "Hello world!" << endl;
    return 0;
}
```

Esercizi opzionali:

- verificate che il risultato non cambia se tutte le istruzioni sono sulla stessa riga
- verificate che, se il programma contiene un errore di sintassi (ad esempio manca un punto e virgola) il compilatore non riesce a produrre il programma, e dà errore (quale? provate a mettere il relazione il messaggio di errore con l'errore che avete inserito)

Materiale da consegnare: esercizio1.cpp (entro la fine dell'esercitazione)

Esercizio 2

Questo codice calcola l'area di un quadrato dopo che l'utente ha inserito il valore del suo lato:

```
#include <iostream>
using namespace std;
int main() {
    double lato,area;
    cout << "Dammi il lato del quadrato" << endl;
    cin >> lato;
    area = lato*lato;
    cout << "L'area vale: " << area << endl;
    return 0;
}
```

Elementi di novità rispetto al programma precedente:

- In questo programma dichiariamo due variabili di tipo double, ovvero dei numeri a virgola mobile con dimensioni di 64 bit, ed eseguiamo delle operazioni tra queste due variabili
- Il valore di una di queste due variabili viene richiesto all'utente tramite l'operatore "cin" (vedremo in che in realtà è un oggetto), che è il corrispettivo di cout per la lettura di stringhe (o numeri) da terminale
- Il comando "cout" può concatenare più elementi nell'output: in questo esempio vengono stampati su terminale una stringa e poi una variabile double

Materiale da consegnare: esercizio2.cpp (entro la fine dell'esercitazione)

Esercizio 3

Calcolo dell'area del rettangolo, dati base ed altezza

```
#include <iostream>
using namespace std;
int main() {
    double base, altezza;
    cout << "Dammi la base e l'altezza del rettangolo" << endl;
    cin >> base >> altezza;
    cout << "L'area vale: " << base*altezza << endl;
    return 0;
}
```

Elementi di novità rispetto al programma precedente:

- Questo programma dimostra che “cin” ammette la concatenazione: si può “leggere” più di una singola variabile con una sola istruzione.
- L’operazione di calcolo dell’area può essere effettuata direttamente “all’interno” di cout, senza dover necessariamente dichiarare un’apposita variabile (la variabile area dell’esercizio precedente)

Materiale da consegnare: esercizio3.cpp (entro la fine dell’esercitazione)

Esercizio 4

Calcolo del lato del quadrato data l'area

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    double area;
    cout << "Dammi l'area del quadrato: " << endl;
    cin >> area;
    if (area>=0){
        double lato = sqrt(area);
        cout << "Il lato vale: " << lato << endl;
    } else {
        cout << "Area negativa ! " << endl;
    }
    return 0;
}
```

Elementi di novità rispetto al programma precedente:

- Questo programma utilizza la libreria `cmath`:
 - dobbiamo includere la direttiva `#include <cmath>`
 - usiamo la funzione di libreria `sqrt()` per il calcolo della radice quadrata
- Questo programma introduce il controllo di flusso `if/else`

Esercizi opzionali:

- Sostituire la funzione di libreria `sqrt()` con la funzione `pow()`, e verificare che elevando alla $\frac{1}{2}$ si ottiene lo stesso risultato che con la radice quadrata

Materiale da consegnare: `esercizio4.cpp` (entro la fine dell'esercitazione)

Esercizio 5

Scrivere un programma che dati da terminale i coefficienti a , b e c di un'equazione di secondo grado

$$a x^2 + b x + c = 0$$

ne trovi le soluzioni.

Si considerino tutti i possibili casi (per $a=0$ l'equazione è di primo grado).

Materiale da consegnare: esercizio5.cpp (entro la fine dell'esercitazione)

Esercizio 6 (per casa)

Si consideri il lancio di un corpo puntiforme (punto materiale) da un punto di coordinate $(0, h)$ con una velocità v_0 che forma un angolo θ rispetto all'orizzontale. Si supponga che la forza di gravità sia orientata lungo l'asse y e diretta verso le y negative.

Scrivere un programma che calcoli la distanza d percorsa dal corpo, rispetto al punto iniziale, quando questo tocca $y=0$.

- Input del programma (da terminale): h , v_0 , θ
- Output del programma: d

Materiale da consegnare: esercizio6.cpp (entro venerdì 19/10 tramite AulaWeb)