

# Laboratorio di Calcolo per Fisici, Terza esercitazione

Canale Lp-P, Docente: Cristiano De Michele

Con questa terza esercitazione potrete continuare a familiarizzare con il terminale di linux e con alcuni aspetti del linguaggio C, quali la dichiarazione di variabili, l'uso di funzioni matematiche e di comandi per l'input/output. Lo scopo è scrivere un programma in C che risolva un problema relativo al moto dei gravi.

## ► Prima parte:

### 1. Se siete in laboratorio in via Tiburtina:

- (a) effettuare il *login* sulla propria macchina *Unix* utilizzando come *userid* e *password* `lccdmXXX`, dove `XXX` è il numero che vi è stato assegnato per il laboratorio (Nota Bene: la *password* è identica allo *userid*)
- (b) aprire una finestra del *terminale*

### Se siete in laboratorio al nuovo edificio “Fermi” di Fisica:

- (a) entrate usando come *userid* `studente` e come *password* `informatica`
  - (b) aprire il terminale
  - (c) Verificate, utilizzando il comando `ls`, se esiste già una cartella chiamata `LCCDMXXX` (lettere maiuscole), dove `XXX` è il numero che vi è stato assegnato per il laboratorio. Se non esiste tale cartella, createla con il comando `mkdir`
  - (d) entrate nella cartella suddetta con il comando `cd`
2. Creare con il comando `mkdir` una cartella chiamata `EX3`, che conterrà il materiale di questa terza esercitazione di laboratorio.
  3. Entrare nella cartella appena creata con il comando `cd EX2`
  4. Aprire con l'editor di testo il file `motogravi.c` (dal terminale con il comando `emacs motogravi.c`), digitare il listato sottostante e salvare il contenuto del file.

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void)
5 {
6
7 }
```

Listato 1: Programma `motogravi.c`

Notare che nel precedente listato `#include <stdio.h>` è una direttiva al preprocessore necessaria per utilizzare le funzioni di libreria per l'input/output (cioè `printf` e `scanf`, che verrà spiegata di seguito), mentre `#include <math.h>` serve per poter utilizzare le funzioni di libreria matematiche che abbiamo visto nella precedente esercitazione.

*Suggerimento:* se volete evitare di scrivere il precedente scheletro di programma C potreste anche copiare il programma sviluppato nella precedente esercitazione e cancellare le istruzioni `printf`. Per fare questo, assumendo che vi troviate nella cartella dell'esercitazione attuale e che il materiale della precedente esercitazione si trovi nella cartella EX2, potete usare dal terminale il seguente comando:

```
1 > cp ../EX2/matematica.c motogravi.c
```

In questo modo vi ritroverete nella cartella attuale un nuovo file chiamato `motogravi.c` il cui contenuto sarà identico a quello del file `matematica.c` che si trova nella cartella EX2 e che potrete opportunamente modificare. Per visionare il contenuto di un file (ad es. proprio `motogravi.c`) da terminale senza aprirlo con `emacs` potete usare il comando:

```
1 > cat motogravi.c
```

Considerate ora un proiettile sparato da terra con velocità  $v_0$  con un angolo  $\alpha$  rispetto all'asse orizzontale, a distanza  $x$  dal punto di lancio raggiunge la quota (da terra):

$$y = x \times \tan(\alpha) - \frac{g \times x^2}{2 \times v_0^2 \times [\cos(\alpha)]^2} \quad (1)$$

dove  $g = 9.82 \text{ m/s}^2$ . Il proiettile ricade sul terreno a una distanza dal punto di lancio  $G$  chiamata gittata con:

$$G = \frac{2 \times v_0^2 \times \sin(\alpha) \times \cos(\alpha)}{g} \quad (2)$$

Tra le distanze  $x_1$  e  $x_2$  dal punto di lancio si trova un muro di altezza  $h$ .

Modificando il codice `motogravi.c` che avete precedentemente creato, scrivere dunque un programma che:

1. Acquisisca da tastiera, utilizzando l'istruzione `scanf()` (vedi sotto per maggiori ragguagli sull'utilizzo di tale istruzione C), i valori di  $v_0$  (in m/s),  $\alpha$  (in radianti) e li stampi su schermo. Notare che il comando `scanf`, richiede degli *argomenti* che vanno racchiusi tra parentesi tonde e separati da virgole come nel caso dell'istruzione `printf`, ovvero vanno forniti nel seguente modo:

```
1 (argomento1, argomento2)
```

Il primo *argomento* (`argomento1`) deve essere una stringa ovvero (una serie di caratteri compresi tra due doppi apici) che indicherà il tipo di dato che deve essere richiesto in input all'utente. Ad esempio, utilizzeremo `%d` per indicare un intero (ovvero un `int`), `%c` per un carattere (ovvero un `char`), `%f` per un numero in singola precisione (ovvero un `float`) e `%lf` per numero in virgola mobile in doppia precisione (ovvero un `double`). *Avvertimento:* notare che, mentre per scrivere un numero in doppia precisione tramite l'istruzione `printf` si usa `%f`, per richiedere l'immissione di un numero in doppia precisione tramite l'istruzione `scanf` si deve usare `%lf` (ci vuole la `l`

prima della  $f!$ ). Il secondo *argomento* dell'istruzione `scanf` (*argomento2*) consisterà nel nome della variabile in cui andrà messo quanto inserito in input, preceduta dal simbolo `&`. Ad esempio per richiedere in input un numero in virgola mobile in doppia precision (`double`) da memorizzare nella variabile `a`, si userà il seguente comando:

```
1 scanf("%lf", &a);
```

E' buona norma fornire anche un messaggio all'utente per richiedere cosa deve inserire. Ad es. per richiedere in input la variabile `double` chiamata `a` si potrebbero usare le seguenti istruzioni:

```
1 printf("Immetti la variabile in virgola mobile a: ");  
2 scanf("%lf", &a);
```

2. Acquisisca da tastiera un valore  $L$  (in m) (sempre utilizzando la `scanf` come nel punto precedente), calcoli la quota  $y$  (in m), utilizzando la formula riportata nell'Eq. 1, raggiunta nel punto  $x = L$  e calcoli la gittata  $G$  (in m) del proiettile, usando Eq. 2. Si ricordi che in C gli operatori di moltiplicazione e divisione sono `*` e `/` rispettivamente, inoltre  $g^2$  in C si può convenientemente scrivere come `g*g`.
3. Stampi i risultati del calcolo, cioè il valore di  $G$  (in m) e il valore di  $y$  (in m) per  $x = L$ .  
*Suggerimento:* si ricordi che per stampare una variabile in virgola mobile si deve usare l'istruzione `printf`, ad es. per stampare una variabile di tipo `double` chiamata `G`, potete usare l'istruzione

```
1 printf("Il valore di G è %f\n", G);
```

*Suggerimento:* Appena dopo la parentesi graffa che segue il `main` si ricordi che vanno dichiarate tutte le variabili che si utilizzeranno poi nel programma. Ad esempio per dichiarare le variabili `L`, `alpha` e `v0` di tipo `double` si deve usare la seguente istruzione:

```
1 double L, alpha, v0;
```

Inoltre si ricorda che per la compilazione del programma `motogravi.c`, dovendo utilizzare funzioni matematiche, va usato il flag `-lm`, ovvero il codice va compilato con il comando:

```
1 > gcc motogravi.c -lm -o motogravi.exe
```

Assumendo  $x_1 = 5$  m,  $x_2 = 6$  m,  $h = 2.05$  m,  $\alpha = 0.7$  rad e  $v_0 = 10$  m/s, utilizzate il programma appena compilato per rispondere alla seguente domanda: il proiettile ha urtato il muro? Scrivete la risposta in un file di testo chiamato `risposte-ex3.txt` utilizzando `emacs` per aprirlo (con il comando `emacs risposte-ex3.txt`) e aggiungete in tale file anche eventuali motivazioni.

### ► Seconda parte:

1. Per  $v_0 = 10$  m/s, calcolare la gittata al variare di  $\alpha$  per almeno 10 diversi valori, stampando ogni volta il valore di  $\alpha$  e la gittata  $G$  separati da una spazio.

2. Copiare i valori ottenuti al punto precedente in un file di testo chiamato `gittata.dat` in modo da avere due colonne, dove nella prima colonna ci saranno i valori di  $\alpha$  e nella seconda i corrispondenti valori della gittata  $G$ .
3. Graficare il file `gittata.dat` utilizzando python ed unendo i punti con delle linee  
Si ricordi che per unire i punti con delle linee potete usare il seguente comando python:

```
plt.plot(x, y, 'x-',label='Gittata')
```

*Suggerimento:* Aniché riscrivere daccapo lo script python, potreste copiare nella cartella relativa all'esercitazione attuale lo script python scritto per la prima esercitazione. Allo scopo potete utilizzare dal terminale il seguente comando (assumendo che vi troviate nella cartella `EX3` e che il materiale della seconda esercitazione sia nella cartella `EX1`):

```
1 > cp ../EX1/ex1_2.py ex3.py
```

In questo modo vi ritroverete nella cartella attuale il file `ex3.py` che potete modificare in modo da graficare i dati contenuti nel file `gittata.dat` (invece del file `temp.dat` della prima esercitazione) utilizzando dei simboli uniti da linee come spiegato pocanzi.

4. Per quale valore di  $\alpha$  risulta massima la gittata? Vi sembra ragionevole quanto avete ottenuto? Scrivete la risposta ed eventuali osservazioni nel file di testo `riposte-ex3.txt`