

Laboratorio di Calcolo, Esercitazione 3, 27-31 ottobre 2025

Canale Pet-Z, Docenti: Shahram Rahatlou, Fabio Bellini, Sibilla Di Pace

Lo scopo di questa esercitazione è di utilizzare il costrutto `if/else` ed i cicli `do-while` e `for` ed usare python per graficare i dati

Un proiettile lanciato con velocità iniziale v_0 e con un angolo θ rispetto al piano orizzontale raggiunge l'altezza

$$y = x \tan \theta - \frac{1}{2} g \frac{x^2}{v_0^2 \cos^2 \theta} \quad (1)$$

dove x è la distanza dal punto di partenza, e $g = 9.81 \text{ m/s}^2$ l'accelerazione gravitazionale. La gittata massima G del proiettile è $G = 2 \frac{\sin \theta \cos \theta}{g} v_0^2$.

Fare login sulla postazione utilizzando le vostre credenziali `lcsrNNN`, dove `NNN` indica il numero del vostro gruppo, ed esempio `lcsr098`. Creare una cartella `LCSR3` con il comando `mkdir` in cui scriverete i programmi di oggi.

► Prima parte

Creare un programma `gittata.c` nella cartella `LCSR3` utilizzando l'editor di testo `emacs`, per eseguire le seguenti operazioni:

1. chiedere di inserire la velocità iniziale v_0 in m/s ed assicurarsi che il valore sia sempre positivo;
2. chiedere di inserire l'angolo θ in gradi ed assicurarsi che sia nell'intervallo $(0, 90]$;
3. chiedere di inserire la distanza x (in metri) alla quale si vuole misurare l'altezza h del proiettile;
4. verificare che la distanza inserita sia inferiore alla gittata massima G e in caso di errore stampare sullo schermo un messaggio utile per l'utente e terminare l'esecuzione chiamando la funzione `exit(-1)`;
5. calcolare l'altezza h del proiettile alla distanza x fornita dall'utente;
6. stampare sullo schermo i valori di h e x con 2 cifre decimali;

Si ricorda che per creare l'eseguibile utilizzando la libreria matematica dovete usare la sintassi `gcc -Wall -o app.exe programma.c -lm` dalla riga di comando nella shell. Ricordate anche di includere gli header file per le librerie di sistema e matematica e di usare l'istruzione `#define` del pre-compilatore per definire le costanti. Le funzioni matematiche $\sin(x)$, $\cos(x)$, e $\tan(x)$ richiedono l'argomento in radianti.

Si consiglia di scrivere il programma in modo incrementale, verificando la corretta compilazione e l'esecuzione almeno dopo ciascuno dei passi indicati nel testo.

► Seconda parte

In questa seconda parte l'utente fornisce la distanza L di un bersaglio da colpire e la velocità iniziale v_0 del proiettile, e il programma stima l'angolo θ con cui sparare il proiettile per colpire il bersaglio. Scrivere un nuovo programma `bersaglio.c` nella cartella LCSR3 utilizzando l'editor di testo `emacs`, per eseguire le seguenti operazioni:

1. chiedere di inserire la velocità iniziale v_0 in m/s ed assicurarsi che il valore sia sempre positivo;
2. chiedere di inserire la distanza L in metri che deve essere sempre positiva ed inferiore a v_0^2/g ;
3. a partire da un valore iniziale $\theta = 0.01$ rad ripetere le seguenti operazioni
 - (a) aumentare l'angolo θ con un passo di $\delta\theta = 0.05$ rad;
 - (b) calcolare la gittata G del proiettile;
 - (c) calcolare $|G - L|$;
 - (d) stampare sullo schermo il valore di θ , con 2 cifre decimali, e di G , con 3 cifre decimali, separati da 2 spazi, senza altro testo o caratteri superflui (ad esempio "0.17 1.247");
 - (e) interrompere il ciclo se $|G - L| < \epsilon$, con $\epsilon = 0.2$ m.
4. stampare sullo schermo il valore dell'angolo per il quale è stato colpito il bersaglio.

► opzionale:

1. chiedere il valore del passo $\delta\theta$ all'utente nell'intervallo $[0.01, 0.5]$ rad;
2. chiedere il valore di ϵ all'utente nell'intervallo $[0.5, 0.001]$ m;

► Terza parte

In questa ultima parte vogliamo graficare l'andamento della gittata G in funzione dell'angolo θ utilizzando un programma in python `grafica.py`, che deve trovarsi sempre nella stessa cartella `LCSR3`. A tal fine

1. creare un file `dati.txt` utilizzando l'editor `emacs` per copiare o scrivere a mano i valori di θ ed G stampati sullo schermo nella seconda parte dal programma `bersaglio.c`;
2. utilizzando `emacs` creare il file `grafica.py` che deve contenere le seguenti istruzioni nel linguaggio python

```
1 # carica moduli pyplot e numpy di python
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 plt.title("Gittata in funzione di theta")
6
7 # carica dati dal file
8 # x e' un array con i dati nella prima colonna (theta)
9 # y e' un array con i dati della seconda colonna (gittata)
10 x, y = np.loadtxt('dati.txt', unpack=True)
11
12 # crea grafica di y in funzione di x
13 plt.plot( x, y, 'o-', label='dati caricati dal file')
14
15 # specifica limite inferiore superiore per asse x e y
16 plt.xlim(0., 1.60)
17 plt.ylim(0., 250.)
18
19 # aggiungi legenda per gli assi
20 plt.xlabel('angolo theta (rad)')
21 plt.ylabel('gittata G (metri)')
22
23 # nome del file in cui salvare il grafico
24 plt.savefig("grafico.png")
25
26 # mostra grafico
27 plt.show()
```

Listato 1: Programma `grafica.py`

questo codice contiene diverse righe di commento (che iniziano con `#`) per capire le istruzioni. Il codice permette di leggere i valori di x e y dal file `dati.txt`, grafica y in funzione di x , e salva il grafico nel file `grafico.png`.

3. eseguire il codice in python invocando l'interprete dalla riga di comando nel terminale

```
1 python3 grafica.py
```