

# Laboratorio di Calcolo per Fisici, Prova di Esame del 26/6/2023

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_  
Matricola: \_\_\_\_\_  Ritirata/o

Lo scopo della prova d'esame è scrivere un programma che simuli una partita con dei dadi *non transitivi*, seguendo la traccia riportata di seguito. Si tenga presente che:

1. Per svolgere il compito avrete a disposizione 3 ore.
2. Si possono usare libri di testo, prontuari e gli appunti, ma non è ammesso parlare con nessuno né utilizzare cellulari, tablet o laptop, pena l'annullamento del compito.
3. Il programma va scritto e salvato esclusivamente sul computer del laboratorio, a cui si deve accedere utilizzando come username **studente** e come password **informatica**.
4. Tutti i file vanno salvati in una cartella chiamata LBC\_NOME\_COGNOME nella home directory, dove NOME e COGNOME indicano rispettivamente il vostro nome e cognome. Ad esempio lo studente Marco Rossi deve creare una cartella chiamata LBC\_MARCO\_ROSSI contenente tutti i file specificati nel testo. All'inizio di tutti i programmi e script va inserito un commento con nome, cognome e numero di matricola. Per consegnare il compito (o dietro richiesta del docente) dovete copiare tutto il contenuto della cartella con il seguente comando: `cp * /media/sf_esame/`
5. Consegnare il presente testo indicando nome, cognome e numero di matricola (vedi sopra), barrando la casella "Ritirata/o" se ci si vuole ritirare.

## ► Parte I

Lo scopo di questa esercitazione è di realizzare un programma che simuli una partita con dei dadi *non transitivi*. I dadi non transitivi sono una serie di dadi in cui se un primo dado ha maggiori probabilità di battere un secondo dado, e quest'ultimo di battere un terzo dado, non è però vero che il primo dado batte il terzo, non valendo quindi la proprietà transitiva.

Un esempio di dadi non transitivi a 6 facce è riportato in figura: il dado A ha sulle 6 facce i seguenti valori {3, 3, 3, 3, 3, 6}; il dado B {2, 2, 2, 5, 5, 5}; il dado C {1, 4, 4, 4, 4, 4}. Il dado A vince sul dado B il 58% (21/36) delle volte; il dado B vince sul dado C il 58% (21/36) delle volte, e il dado C vince sul dado A il 69% (25/36) delle volte.

Si scriva un programma in C che simuli il lancio di tre dadi non transitivi come segue:

1. Un array `dices[] []`, di tipo e dimensioni opportune, contiene la sequenza delle facce dei tre dadi A, B, C. Potete utilizzare le sequenze dell'esempio riportato sopra.
2. Una funzione `roll()` simula l'esito di un singolo lancio di dadi. La funzione riceve cioè in input un indice che identifica i due dadi da tirare (0 == coppia AB, 1 == coppia BC, 2 == coppia CA), e restituisce in output il risultato del lancio. Per farlo, tale funzione deve anche avere come argomenti due puntatori a intero da utilizzare per memorizzare i valori usciti nel lancio (questi valori verranno poi utilizzati come argomenti della funzione `print_esito()` discussa al punto 5).
3. Il programma chiede quindi all'utente di inserire un numero `N_rolls` di lanci, compreso tra 1 e 10.000.000 (estremi inclusi); il programma reitera la richiesta fino a quando l'utente non inserisca un numero valido.
4. Il programma esegue quindi `N_rolls` lanci per ciascuna coppia di dadi, cioè lancia `N_rolls` volte il dado A e il dado B, il dado B e il dado C, il dado C e il dado A, in questo ordine.

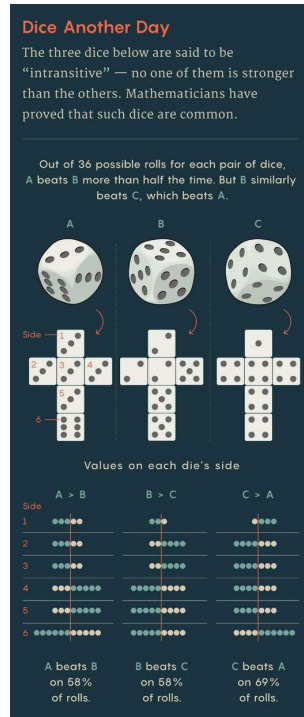


Figura 1: Dadi Intransitivi A,B,C usati nell'esercizio.

- Per i primi 5 lanci, il programma stampa su schermo l'esito del lancio, utilizzando il formato: `Lancio n. 2, dado A = 3, dado B = 5, vince il dado B`. Per tale stampa si richiede di usare una funzione `print_esito()` che riceverà in ingresso 5 argomenti: (i) il numero di lanci effettuati, (ii) il risultato del lancio restituito dalla funzione `roll()`, (iii) un indice che indichi la coppia di dadi lanciati (0 == coppia AB, 1 == coppia BC, 2 == coppia CA), e (iv)-(v) i 2 valori usciti nel lancio effettuato.
- Il programma tiene traccia, tramite tre contatori, di quante volte il dado A abbia battuto il dado B (NAB), di quante volte il dado B abbia battuto il dado C (NBC) e di quante volte il dado C abbia battuto il dado A (NCA), in quest'ordine.
- Alla fine degli `N_rolls` lanci, il programma stampa un messaggio di riepilogo, contenente il numero di lanci effettuati, e il valore dei tre contatori NAB, NBC e NCA.

### ► Parte II

Memorizzare in un file chiamato `freq.dat` su righe distinte le frequenze relative alle vittorie di A su B, B su C e C su A nel caso in cui `N_rolls` sia pari a 10.000.000. Creare quindi uno script in python (chiamato `NOME_COGNOME.py`) che grafichi l'istogramma relativo al file `freq.dat`.

### ► Parte III

Copiate il codice sviluppato nella parte I in un nuovo file chiamato `intransitive_parteIII.c` e modificalo come segue:

- Aggiungere un array `histo[]`, di tipo e dimensioni opportune, che contenga la frequenza con cui si è verificato che il dado A abbia battuto il dado B (FAB), il dado B il dado C (FBC), e il dado C il dado A (FCA)– in quest'ordine.
- Introdurre un ciclo sulla variabile `N_rolls`, in modo che il programma ripeta le operazioni 4–7 per `N_rolls` = 10, 100, 1000, 10.000 e 100.000.
- Salvare su di un file `histo.dat` le seguenti informazioni: `N_rolls`, FAB, FBC, FCA.