

Laboratorio di Calcolo per Fisici, Prova d'esame del 27/06/2025, AA 2024/2025

Nome: _____ Cognome: _____
Matricola: _____ Ritirata/o

Lo scopo della prova d'esame è scrivere un programma che simuli il moto di una navicella spaziale in un campo di asteroidi. Si tenga presente che:

1. Per svolgere il compito si hanno 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. **Tutti i file vanno salvati in una cartella chiamata EJUN25_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 EJUN25_MARCO_ROSSI contenente tutti i file specificati nel testo. **Tutto ciò che non si trova all'interno della cartella suddetta non verrà valutato.** In tutti i programmi e script inserite all'inizio un commento con il vostro nome, cognome e numero di matricola.
4. **Dovete consegnare il presente testo indicando nome, cognome e numero di matricola** (vedi sopra), barrando la casella "Ritirato/a" se ci si vuole ritirare, ovvero se non si vuole che la presente prova venga valutata.
5. **Per consegnare il compito** dovrete eseguire, all'interno della cartella creata in precedenza (come spiegato al punto 3), il seguente comando da terminale: `cp * /media/sf_esame/`

Una navicella spaziale si muove casualmente in un campo di asteroidi, rappresentato da una griglia bidimensionale quadrata di dimensione fissata. La navicella parte dalla cella in alto a sinistra (0,0) e può muoversi, a ogni passo, in una delle quattro direzioni cardinali (su, giù, sinistra, destra), restando entro i confini della griglia.

Sulla griglia sono distribuiti `nstar` asteroidi (indicati con il simbolo `*`), posti casualmente in celle diverse dalla posizione iniziale della navicella. La navicella parte con un numero `shields` di scudi, e ogni volta che la navicella entra in una cella contenente un asteroide, il valore dei suoi scudi diminuisce di uno. La simulazione termina non appena il valore degli scudi scende sotto lo zero.

► **Prima parte:** Si scriva un programma in c che simuli il moto dell'astronave, rispettando le seguenti specifiche:

1. Tramite una direttiva `define` viene inizializzata a 10 la dimensione della griglia `SIZE`.
2. La griglia iniziale viene riempita con una serie di trattini `-`, che rappresentano caselle vuote.
3. Il programma chiede all'utente di inserire un valore per `nstar` (numero di asteroidi) nell'intervallo `[10, 20]` (estremi inclusi). La richiesta all'utente viene ripetuta fino a quando l'utente non inserisce un valore valido.

-	-	-	-	-	-	-	-	-	*
-	-	*	-	-	-	-	-	-	-
-	-	-	-	-	-	-	*	-	-
-	-	-	-	*	-	-	-	-	-
-	-	-	-	-	-	*	-	-	-
-	*	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	*	-
-	-	-	*	-	-	-	-	-	-
-	-	-	-	-	*	-	-	-	-
*	-	-	-	-	-	-	-	-	-

Figura 1: Rappresentazione schematica del campo di asteroidi; -: spazio vuoto, *: asteroide.

4. Il valore degli scudi `shields` viene posto uguale a 5.
5. Una funzione `fillmatrix()`, di tipo e argomenti opportuni, posiziona casualmente gli `nstar` asteroidi sulla griglia, assicurandosi che la posizione iniziale dell'astronave (0,0) sia libera.
6. La astronave viene inizialmente posta in (0,0).
7. Una funzione `move()`, di tipo e argomenti opportuni, aggiorna la posizione dell'astronave, muovendola casualmente in una delle quattro direzioni cardinali. Se il movimento porta l'astronave fuori dai bordi della griglia, la posizione resta invariata e la mossa non viene conteggiata: in tal caso, la funzione deve ripetere la scelta finché viene effettuato uno spostamento valido.
8. Se l'astronave finisce in una casella occupata da un asteroide, il valore degli scudi diminuisce, e il programma stampa su schermo il messaggio: `Dopo 33 passi l'astronave ha colpito un asteroide in (0, 8) scudi=7.`
9. Dopo ogni mossa, alla casella della griglia relativa alla posizione attuale dell'astronave viene assegnato il valore `A`.
10. Il programma stampa su schermo la configurazione iniziale e finale della griglia. Un esempio di griglia finale che rispecchia le specifiche date (per cui le caselle visitate dall'astronave sono indicate da una `A` maiuscola, le caselle contenenti un asteroide da `*`, e le caselle mai visitate da un trattino `-`) è:

	A	A	A	A	-	-	-	A	A	A	
	-	-	A	A	A	A	-	A	A	A	
	-	A	A	A	A	A	A	A	A	-	
	-	A	-	*	A	A	-	-	A	*	
	A	A	A	A	-	-	-	-	A	*	
	-	A	A	A	A	A	A	A	A	*	
	-	-	*	A	-	-	-	-	-	-	
	-	-	-	-	*	-	-	-	-	-	
	-	*	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	*	*	-	*	

11. Dopo aver stampato la configurazione finale, il programma stampa su schermo un messaggio di riepilogo, secondo il formato:

Dopo 68 passi l'astronave e' esplosa.

► **Seconda parte:** Dopo aver verificato che il codice funzioni correttamente, modificare il programma nel modo seguente:

1. Il valore inserito dall'utente nell'intervallo $[10, 20]$ viene immagazzinato in `nstar_min`.
2. Contenga un ciclo esterno che simuli il moto dell'astronave per tutti i valori interi di `nstar` compresi tra `nstar_min` e 99 (incluso).
3. Stampi l'output richiesto dalla prima parte solo per il primo valore di `nstar`, cioè per `nstar == nstar_min`.
4. Crei un file di nome `passi.dat` che contenga due colonne, contenenti il numero di asteroidi e il numero di passi corrispondenti alla simulazione effettuata.

► **Esercizio in Python:** Creare uno script in python (`NOME_COGNOME.py`) che, a partire dal file `passi.dat` generato al punto precedente con `nstar_min` posto uguale a 10, crei il grafico del numero di passi in funzione del numero di asteroidi, e lo salvi su di un file `NOME_COGNOME.png`. Il grafico deve essere completo di titolo e di etichette per gli assi.