

Laboratorio di programmazione e Informatica 1

- A.A. 2024-2025 -

Terzo-appello - 24 giugno 2025

ISTRUZIONI:

- Gli esercizi vanno chiamati come `CognomeNomeEse1.c`, `CognomeNomeEse2.c`, `CognomeNomeEse3.c` e consegnati tramite la chiavetta Usb del docente. (Consegnare soltanto i file .c)
- La prima riga di ogni programma C deve contenere il proprio nome e cognome come commento.
- I programmi devono essere strutturati in funzioni e completi di commenti che spieghino il procedimento.
- Quando è richiesto lo studio della complessità dell'algoritmo implementato, questa va descritta alla fine del programma, sempre sotto commento.
- Ogni esercizio vale 10 punti. Bisogna fare ≥ 18 punti con almeno 5 punti su ogni esercizio.
- **Attenzione!** Non saranno valutati programmi che non passano la fase di compilazione. Si consiglia pertanto di “mettere sotto commento” le parti di programma che danno errore in compilazione.

ATTENZIONE! Non saranno valutati programmi che non passano la fase di compilazione. Si consiglia pertanto di “mettere sotto commento” le parti di programma che danno errore in compilazione.

Tempo a disposizione: 3 ore.

ESERCIZI

Esercizio 1

Data una matrice quadrata di dimensione $m \times m$ a valori in un insieme $\{0, 1, \dots, n\}$, il *peso* di una posizione (i, j) , indicato con $p(i, j)$, è dato dalla differenza tra la somma degli elementi della riga i e la somma degli elementi della colonna j . Ad esempio nella seguente matrice:

6	2	4	4
0	0	0	6
4	6	2	0
2	0	8	0

$$p(0, 0) = 4, p(1, 0) = -6 \text{ e } p(3, 3) = 0.$$

- Scrivere una funzione in C `void Genera(int a[DIM][DIM])` che genera una matrice di DIM righe e DIM colonne a valori interi pari e random tra 0 e 22.
- Scrivere una funzione:

`int CalcolaPeso(int a[][COL1], int *rmax, int *cmax)`

che calcola e restituisce il numero di posizioni della matrice con peso negativo. Inoltre calcola il valore massimo tra i pesi delle posizioni della matrice e, scegliendo una delle posizioni con peso massimo, ne salva i relativi indici di riga e colonna nelle variabili `*rmax` e `*cmax`.

- Scrivere un programma in C che:
 1. Usando la funzione **Genera**, genera una matrice di dimensione 15x15 e stampa tale matrice ben formattata sullo schermo.
 2. Utilizza la funzione **CalcolaPeso** per calcolare quanti sono gli elementi di peso negativo e per individuare una posizione di un elemento con peso massimo.
 3. Stampa sullo schermo le informazioni calcolate.
- Discutere la complessità degli algoritmi implementati in funzione della dimensione della matrice.

Esercizio 2

Si scriva un programma C che richiede all'utente di inserire una frase e poi modifica la stringa-frase in modo che tra una parola e l'altra ci siano 4 spazi bianchi. Alla fine la nuova stringa viene stampata sullo schermo e viene scritto di quante parole è composta. Non è consentito utilizzare funzioni predefinite sulle stringhe.

Nota: Deve essere effettivamente generata una nuova stringa prima di essere stampata. Alla fine deve essere scritto di quanti caratteri la nuova stringa è più lunga di quella originale.

Esercizio 3

Sia L una lista di interi.

1. Si scriva una funzione **Raddoppia** che prende in input una lista L e restituisce una lista R di lunghezza doppia rispetto ad L in cui gli elementi della prima metà sono quelli di L e nella seconda metà sono rispettivamente e ordinatamente il doppio di quelli della prima metà.
Ad esempio se $L = [0, 1, 7, 3, 4]$ allora $R = [0, 1, 7, 3, 4, 0, 2, 14, 6, 8]$.
La funzione dovrà essere di complessità lineare nella lunghezza della lista.
2. Si inserisca tale funzione nel codice fornito listaBase3app.c rinominando il file come **CognomeNomeEse3.c**