

Laboratorio di programmazione e Informatica 1

- A.A. 2024-2025 -

Quarto-appello - 15 luglio 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 $m \times n$ a valori interi, definisco *asse verticale* la colonna i tale che la somma dei valori di tutte le posizioni a sinistra della colonna i è uguale alla somma dei valori di tutte le posizioni a destra della colonna i . Ad esempio nella seguente matrice:

6	2	4	1	0
0	5	0	0	1
4	6	2	0	3
5	2	2	2	0

la seconda colonna ($i = 1$) è un asse.

- Scrivere una funzione in C `void Genera(int a[DIM][DIM])` che genera una matrice di DIM righe e DIM colonne a valori interi random tra 0 e 9 tale che in ogni riga ogni elemento sia diverso dall'ultimo.
- Scrivere una funzione:

```
int CalcolaAsse(int a[][COL1], int *somma)
```

che calcola e restituisce l'indice dell'asse se esiste (altrimenti restituisce -1) e mette nella variabile `somma` il valore della somma delle posizioni a sinistra (o a destra!) dell'asse. Se non esiste l'asse, in `somma` ci sarà un valore qualsiasi.

- In alternativa (+2 punti):
Scrivere una funzione:

```
int CalcolaAsse2(int a[][COL1], int *sommaL, int *sommaR)
```

che, calcola e restituisce l'indice della colonna in cui la differenza tra la somma delle posizioni a sinistra e delle posizioni a destra della colonna stessa sia minima (in valore assoluto) e mette tali due somme nelle variabili `sommaR` e `sommaL`. Ovviamente se esiste l'asse le due somme saranno uguali.

- 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 `CalcolaAsse` (oppure `CalcolaAsse2`) sulla matrice generata.
 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 5 parole e poi le modifica in modo che tutte le *d*, le *s* e le *t* siano raddoppiate. Ad esempio la parola "studio" diventa "sstuudio" mentre la parola "addio" diventa "adddio" . Le nuove stringhe verranno poi stampate sullo schermo e viene scritto anche qual'è la parola più lunga (tra quelle modificate). Non è consentito utilizzare funzioni predefinite sulle stringhe.

Nota: Devono essere effettivamente generate nuove stringhe prima di essere stampate.

Esercizio 3

Sia *L* una lista di interi.

1. Si scriva una funzione `Raddoppia` che prende in input una lista $L = a_1, a_2, \dots, a_k$ e restituisce una nuova lista $R = a_1, b_1, a_2, b_2, \dots, a_k, b_k$ di lunghezza doppia rispetto ad *L* tale che $b_i = 2 * a_i$ per ogni $i = 1, \dots, k$.
Ad esempio se $L = [0, 1, 7, 3, 4]$ allora $R = [0, 0, 1, 2, 7, 14, 3, 6, 4, 8]$.
La funzione dovrà essere di complessità lineare nella lunghezza della lista.
2. Si inserisca tale funzione nel codice fornito listaBase4app.c rinominando il file come `CognomeNomeEse3.c`