

Laboratorio di programmazione e Informatica 1

- A.A. 2024-2025 -

Primo-appello - 4 febbraio 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

- 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 in $\{-1, 1\}$.
- Scrivere una funzione `int Somma0(int a[][DIM], int n)` che prende in input la matrice e restituisce se esiste un indice *i* tale che la somma degli elementi dalla riga 0 alla riga *i* sia nulla. Se tale indice non esiste la funzione restituisce -1 .
- Scrivere una funzione `int MinDiff(int a[][DIM], int n, int *sum1, int *sum2)` che prende in input la matrice e restituisce la riga *i* tale che la differenza tra la somma dei valori dalla riga 0 alla riga *i* e la somma delle rimanenti righe da *i* + 1 ad *n*, sia minima e mette nelle variabili `sum1` e `sum2` rispettivamente le due somme "sopra" e "sotto" la riga *i*.
- Scrivere un programma in C che:
 1. Usando `Genera`, inizializza una matrice di dimensione 25x25 e stampa tale matrice sullo schermo.

2. Utilizza le funzioni `Somma0` e `MinDiff` e scrivi i risultati sullo schermo.
 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 la riscrive abbreviata sostituendo:

`per` con `x`

`sei` con `6`

Ad esempio, se il testo originale è: `sei perfetto`

il testo compresso sarà: `6 xfetto`

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

NOTA: Non è consentito utilizzare funzioni predefinite sulle stringhe.

Esercizio 3

Sia data una lista di interi.

1. Si implementi una funzione in C `SdoppiaLista`, che prende in input una lista A (data come un puntatore al suo primo nodo) di valori interi $a_1, a_2, \dots, a_n$, chiede all'utente un valore x e crea una nuova lista B estraendo da A tutti gli elementi che sono strettamente maggiori di x . Alla fine quindi la lista A conterrà gli elementi che sono minori o uguali ad x . Gli elementi in B devono trovarsi nello stesso ordine in cui erano in A . La funzione deve essere di complessità lineare rispetto alla lunghezza della lista in input.
2. Si inserisca tale funzione nel codice fornito `listaBase.c` rinominando il file come `CognomeNomeEse3.c`