

Laboratorio di programmazione e Informatica 1

- A.A. 2023-2024 -

Prova di laboratorio: Terzo appello - 26 giugno 2024

ISTRUZIONI: La prima riga di ogni programma C deve contenere il proprio nome e cognome. I programmi devono essere strutturati in funzioni e completi di commenti che spieghino il procedimento.

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.

Ogni esercizio vale 10 punti. Per passare all'orale bisogna avere un punteggio ≥ 18 con almeno 5 punti in ogni esercizio.

Tempo a disposizione: 3 ore.

ESERCIZI

Esercizio1

Data una matrice a valori interi in $\{0, 1\}$, definisco *segmento orizzontale* una successione di 1 consecutivi nella stessa riga. La *lunghezza* del segmento è il numero di 1 contenuti mentre l'*inizio* del segmento sono le coordinate (i, j) dell'elemento con j minimo della sequenza di 1. Vogliamo calcolare la lunghezza e l'inizio del segmento di lunghezza massima.

Ad esempio nella seguente matrice:

1	0	1	1	1	0	1	1
0	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1
1	1	1	1	1	0	0	0
1	1	1	0	0	1	1	1
1	0	0	0	0	1	1	1
1	0	1	1	1	1	1	0

un segmento di lunghezza massima si trova in posizione $(1, 3)$ (iniziando a contare da zero) e ha lunghezza 5. Ce n'è anche uno in $(3, 0)$ e un altro in $(6, 2)$.

1. Scrivere una funzione in C **MaxSegm** che prende in input una matrice e calcola e restituisce la lunghezza del massimo segmento contenuto salvando gli indici dell'inizio del segmento in opportune variabili passate per riferimento dal main. (Se nella matrice ci sono più segmenti di lunghezza massima, riportare quello di inizio in (i, j) con i minimo).

NOTA: Una versione semplificata potrebbe essere quella che non restituisce le coordinate dell'inizio del segmento massimo.

2. Scrivere un programma in C (funzione main) che genera una matrice di interi random in $\{0, 1\}$ di dimensione 35×25 e richiamando la funzione **MaxSegm** calcola l'inizio e la lunghezza del massimo segmento contenuto. Poi stampa le informazioni calcolate. (Tutte le stampe sono fatte nel main).

3. Analizzare la complessità dell'algoritmo utilizzato nella funzione **MaxSegm**. (inserirlo alla fine del file `.C` come commento).

Esercizio 2

Si scriva un programma `C` che richiede all'utente di inserire una frase e poi sostituisce l'ultima lettera di ogni parola con il carattere `*`. (Assumo per semplicità che la frase non contenga caratteri di punteggiatura).

Ad esempio, se il testo originale è: `Oggi scrivo un programma in C`

il testo modificato sarà: `Ogg* scriv* u* programm* i* *`

NOTA: Non è consentito utilizzare funzioni predefinite sulle stringhe.

Esercizio 3

Sia L una lista di interi.

Si scriva una funzione **EstraiMedia** che prende in input una lista L ed restituisce una lista P contenente tutti i nodi di L che hanno un valore $> m$ dove m è la media dei valori contenuti nei nodi di L . Tali nodi in P devono essere eliminati da L e devono trovarsi in P nello stesso ordine in cui si trovano in L . La funzione proposta deve avere complessità lineare nella lunghezza della lista.

Versione Semplificata

Si scriva una funzione **SeparaMedia** che prende in input una lista L ed restituisce una lista P contenente i valori nei nodi di L che hanno un valore $> m$ dove m è la media tra tutti i valori contenuti nei nodi di L . La lista L rimane invariata I nodi in P devono trovarsi nello stesso ordine in cui si trovano in L .

Inserire tale funzione nel file `ListaBase.c` allegato.