

Laboratorio di programmazione e Informatica 1

- A.A. 2021-2022 -

Prova di laboratorio: PreAppello - 26 maggio 2022

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. Gli esercizi vanno consegnati come file singoli denominati rispettivamente `CognomeNomeEse1.c`, `CognomeNomeEse2.c`, `CognomeNomeEse3.c`.

L'analisi, ove richiesta, dell'algoritmo utilizzato va scritta alla fine del file `.C` come commento.

L'esercizio 3 consiste nello sviluppo di una funzione da inserire poi nel file `ListeBase0App.c` fornito a parte.

Ogni esercizio vale un max di 10 punti. Si è ammessi all'orale se si raggiunge un punteggio totale ≥ 18 con un punteggio ≥ 5 sui singoli esercizi.

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. I file eseguibili non vanno consegnati.

Tempo a disposizione: 3 ore.

ESERCIZI

Esercizio1

Data una matrice M a valori interi, una posizione (i, j) si definisce *DoppioRowCol* se il valore $M(i, j)$ compare anche in un'altra posizione h della riga i e in un'altra posizione k della colonna j , ($h \neq j$ e $k \neq i$).

1. Scrivere una funzione `int DoppioRowCol(int A[ROW][COL], int *pmax1, int *pmax2)` che calcola e restituisce il numero delle posizioni *DoppioRowCol* della matrice; calcola inoltre i due valori distinti più grandi che si trovano in queste posizioni *DoppioRowCol* e ne passa il valore al main tramite i puntatori `pmax1` e `pmax2`.

Versione facilitata: calcolare solo il numero delle posizioni *DoppioRowCol*.

2. Scrivere una funzione che genera una matrice di interi a valori nell'intervallo $[33, 66]$ in modo che in ogni riga non ci siano elementi consecutivi uguali.

Versione facilitata: non imporre la condizione su elementi consecutivi di ogni riga.

3. Scrivere un programma in C (funzione main) che genera una matrice di interi random in $[33, 66]$ di dimensione 25×15 e la stampa ben formattata sullo schermo. Poi, richiamando la funzione `DoppioRowCol`, calcola il numero delle posizioni *DoppioRowCol* e i due valori più grandi in tali posizioni. Poi stampa le informazioni calcolate.

4. Analizzare la complessità dell'algoritmo utilizzato nella funzione `DoppioRowCol`.

Esercizio 2

Data una stringa s di lunghezza n e un parametro $k < n$, posso considerare le k trasformazioni di s in cui prendo un simbolo tra i primi k di s e lo sposto alla fine della stringa. Ad esempio, data la stringa `programmazione` e dato il valore $k = 5$, le 5 trasformazioni possibili sono:

`rogrammazionep`

`pogrammazione`

`prprogrammazioneo`

`prorammazioneg`

`progammazione`

Si scriva un programma C che richiede all'utente di inserire una stringa e un valore k e poi scrive sullo schermo tutte le k trasformazioni e alla fine dice quale è la prima in ordine lexicografico (l'ordine del vocabolario). Nell'esempio la prima è `pogrammazione`).

Non è consentito utilizzare funzioni predefinite sulle stringhe. **Versione Semplificata:** Stampa soltanto le k trasformazioni.

Esercizio 3

Sia L una lista di interi (positivi e negativi)

1. Si scriva una funzione `SuccNulla` che prende in input una lista L e verifica se esiste una successione di nodi consecutivi la cui somma è nulla. Ad esempio nella lista $L = 2 \rightarrow 3 \rightarrow 4 \rightarrow -7 \rightarrow 5 \rightarrow -6 \rightarrow 6$ i nodi consecutivi $3 \rightarrow 4 \rightarrow -7$ danno somma 0.
2. Si scriva una funzione `CancellaSuccNulla` che prende in input una lista L e cancella gli elementi di una successione nulla in L . Nell'esempio precedente la nuova lista diventa $L = 2 \rightarrow 5 \rightarrow -6 \rightarrow 6$
3. Si discuta la complessità degli algoritmi utilizzati in funzione della lunghezza della lista L . (inserirlo alla fine del file `.C` come commento).