

Laboratorio di programmazione e Informatica 1

- A.A. 2020-2021 -

Prova di laboratorio: PreAppello - 3 giugno 2021

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 `EseListaBaseApp0.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, definisco una posizione (i, j) *importante* se il valore $M(i, j)$ è maggiore della media dei valori della riga i e della media dei valori della colonna j .

1. Scrivere una funzione `int Importanti(int A[ROW][COL], int *mi, int *mj, int *max)` che calcola e restituisce il numero di elementi importanti nella matrice; calcola inoltre il max (in presenza di più massimi, sceglierne uno a piacere) tra i valori degli elementi importanti e ne passa il valore e le coordinate al main tramite le variabili `mi`, `mj` e `max`.

Versione facilitata: calcolare solo il numero degli elementi importanti.

2. Scrivere una funzione che genera una matrice di interi a valori nell'intervallo $[3, 47]$ in modo che il primo elemento di ogni riga sia diverso dal primo elemento della riga precedente e della riga successiva.

Versione facilitata: non imporre la condizione sui primi elementi di ogni riga.

3. Scrivere un programma in C (funzione `main`) che genera una matrice di interi random in $[3, 47]$ di dimensione 25×15 e la stampa ben formattata sullo schermo. Poi, richiamando la funzione `Importanti`, calcola il numero degli elementi importanti della matrice e il valore e la posizione del massimo tra questi. Poi stampa le informazioni calcolate.

4. Analizzare la complessità dell'algoritmo utilizzato nella funzione `Importanti`. sia in termini di tempo che di memoria supplementare eventualmente utilizzata (al netto di quella necessaria per memorizzare input ed output).

Esercizio 2

Si scriva un programma C che richiede all'utente di inserire una frase e poi scrive sullo schermo, una sotto l'altra, tutte le singole parole che contengono la lettera *r*.

Ad esempio se l'utente ha inserito la frase:

```
Questo e' certamente un esame di laboratorio veramente semplice .
```

Il programma darà in output:

Le parole cercate sono:

```
certamente
laboratorio
veramente
```

Non è consentito utilizzare funzioni predefinite sulle stringhe. Si può supporre che le parole sono separate solo da uno spazio bianco. Può essere utile inserire prima tutte le singole parole in un vettore di stringhe

Versione Semplificata: Stampa tutte le parole della frase, una sotto l'altra.

Variante (+2 punti): Si aggiunge anche la possibilità di leggere un testo da file (e poi si scrivono sullo schermo tutte le parole del testo che contengono la *r*).

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 seconda metà sono esattamente uguali a quelli della prima metà.
Ad esempio se $L = [0, 1, 2, 3, 4]$ allora $R = [0, 1, 2, 3, 4, 0, 1, 2, 3, 4]$.
2. Si scriva una funzione `Palindroma` che prende in input una lista L ed restituisce una lista P di lunghezza doppia rispetto ad L in cui gli elementi della seconda metà sono esattamente uguali a quelli della prima metà ma nell'ordine contrario.
Ad esempio se $L = [0, 1, 2, 3, 4]$ allora $P = [0, 1, 2, 3, 4, 4, 3, 2, 1, 0]$.
3. Si discuta la complessità degli algoritmi utilizzati in funzione della lunghezza della lista L . (inserirlo alla fine del file `.C` come commento).