

# Compito di Elementi di Programmazione

17 Luglio 2017 - Turno 1 – Traccia A

Nome \_\_\_\_\_ Cognome \_\_\_\_\_ Matricola \_\_\_\_\_

Sviluppare un programma C (C++) che legge le dimensioni e gli elementi di una matrice di numeri reali.

Il programma:

1- Il calcola la media degli elementi di ogni riga

2- Aggiunge una colonna alla matrice tale che ogni elemento è pari alla media degli elementi della riga a cui appartiene

3- Ordina gli elementi di ogni riga per valori crescenti.

3 Ordina le righe per valori crescenti dell'ultima colonna

## Matrice Originale

0	2.4	3	4	0
3	0	5	10	2
1	0	2	1	3
8	0	0	0	0

## Matrice non ordinata:

0	2.4	3	4	0	1.88
3	0	5	10	2	4
1	0	2	1	3	1.4
8	0	0	0	0	1.6

## Matrice Finale

0	1	1	1.4	2	3
0	0	1.88	2.4	3	4
0	0	0	0	1.6	8
0	2	3	4	5	10

Risolvere il problema articolando il programma nei seguenti sottoprogrammi:

- funzione per lettura dimensioni e gli elementi della matrice
- funzione per calcolo media riga
- funzione per ordinamento singola riga
- funzione per ordinamento righe matrice

### **Per gli studenti che svolgono la prova da nove crediti:**

- leggere le dimensioni e gli elementi della matrice di ingresso da un file di testo "input.txt"
- salvare la matrice in un file binario "output.bin"

# Compito di Elementi di Programmazione

17 Luglio 2017 - Turno 1 – Traccia B

Nome \_\_\_\_\_ Cognome \_\_\_\_\_ Matricola \_\_\_\_\_

Sviluppare un programma C (C++) che legge n+1 frasi fino a quando la frase letta non corrisponde alla parola “exit” oppure è stata letta la 100-esima frase.

Il programma confronta la prima frase con tutte le successive, e per ogni confronto memorizza in una struct:

numero di frase letta

esito del confronto (1 se la prima frase è contenuta in questa, altrimenti 0)

posizione della prima frase nella frase corrente, altrimenti -1

N.B. Per il confronto delle 2 frasi non utilizzare funzioni di libreria

## Esempio

*acque*

*Venere, e fea quelle isole feconde*

*col suo primo sorriso, onde non tacque*

*le tue limpide nubi e le tue fronde*

*l'inclito verso di colui che l'acque*

*exit*

→ **1,0,-1; 2,1,32;3,0,-1;4,1,30**

Risolvere il problema articolando il programma nei seguenti sottoprogrammi:

- funzione per il confronto stringhe
- funzione per controllare se una stringa è contenuta nell'altra

## **Per gli studenti che svolgono la prova da nove crediti:**

- leggere le frasi da un file di testo “input.txt”

- l'array di struct in un file binario output.bin”

# Compito di Elementi di Programmazione

17 Luglio 2017 - Turno 2 – Traccia A

Nome \_\_\_\_\_ Cognome \_\_\_\_\_ Matricola \_\_\_\_\_

Sviluppare un programma C (C++) che legge una serie di stringhe.

La prima stringa contiene numero di righe e di colonne di una matrice es: "4,5"

Ogni riga successiva contiene indici di riga, di colonna e valore dell'elemento es: "0,3,4"

Il programma continua a leggere fino a quando non viene letta la stringa "stop".

Tutti gli elementi non letti della matrice son da assumere pari a zero.

Inserire nella matrice una riga contenente i massimi di tutte le colonne e ordinare le colonne in maniera crescente secondo la riga dei massimi.

Es:

4,5

0,1,2

0,2,3

0,3,4

1,0,3

1,2,5

[...]

stop

**Matrice Originale**

0	2	3	4	0
3	0	5	10	2
1	0	2	1	3
8	0	0	0	0

**Matrice Finale non ancora Ordinata**

0	2	3	4	0
3	0	5	10	2
1	0	2	1	3
8	0	0	0	0
8	2	5	10	3

Risolvere il problema articolando il programma nei seguenti sottoprogrammi:

- funzione per lettura dimensioni e elementi matrici
- funzione per calcolo delle somme di una colonna
- funzione per ordinamento colonne matrice

**Per gli studenti che svolgono la prova da nove crediti:**

- leggere le dimensioni e gli elementi della matrice di ingresso da un file di testo "input.txt"
- salvare la matrice finale in un file binario "output.bin"

# Compito di Elementi di Programmazione

17 Luglio 2017 - Turno 2 – Traccia B

Nome \_\_\_\_\_ Cognome \_\_\_\_\_ Matricola \_\_\_\_\_

Sviluppare un programma C (C++) che legge n stringhe. Ogni stringa può contenere una lettera o una sequenza di coppie di interi. Il programma termina la lettura quando viene inserita una riga vuota.

1. Definire una struct che contiene come campi una lettera e due interi.
2. Per ogni coppia di interi salvare in una struct del tipo definito l'ultima lettera inserita e la coppia di interi letta.
3. Per ogni struct, salvare la lettera contenuta in una matrice di caratteri nella posizione indicata dalla coppia di interi.
4. Copiare tutti gli elementi della matrice per colonna in un array di caratteri e stampare la stringa così ottenuta

## Esempio:

e  
0,0,0,2,3,3  
i  
0,1,1,0,1,2,1,3  
b  
0,3  
c  
1,1,2,1  
f  
2,0  
u  
3,0  
m  
2,2,3,2,4,1  
[...]

## Matrice corrispondente

e	i	e	b
i	c	i	i
f	c	m	l
u	o	m	e
s	m	o	

## Array finale:

eifusiccomeimmobile

Risolvere il problema articolando il programma nei seguenti sottoprogrammi:

- funzione per lettura delle stringhe
- funzione per la copia del vettore di struct nella matrice di caratteri
- funzione per la copia delle colonne della matrice in un array

**N.B.: Ritenere che al massimo possono essere inseriti 10.000 caratteri**

## Per gli studenti che svolgono la prova da nove crediti:

- leggere le stringhe da un file di testo "input.txt"
- salvare l'array finale in un file binario "output.bin"