

Università di Catania  
Dipartimento di Matematica e Informatica  
Corso di Studio in Informatica, A.A. 2024-2025  
Prova di Laboratorio  
31 Ottobre 2025

**Descrizione del programma**

Scrivere un programma in C che:

- **A.** Prenda un input da tastiera (argomenti della funzione main) costituito dal nome di un file di testo (ES: input.txt) e due numeri interi N ed M; il file contiene una matrice di double di dimensione N x M, dove ogni riga del file di testo contiene una riga della matrice stessa. I numeri N ed M dovranno essere contenuti nell'intervallo [5,10];
- **B.** il programma crei in memoria una matrice A di dimensioni N x M di puntatori a double. La matrice sia popolata da puntatori a celle double che contengono i corrispondenti numeri presenti nel file di input.
- **C.** Si crei una nuova matrice B di dimensioni N x M di puntatori a double, in cui ogni singolo valore della matrice corrisponderà al valore corrispondente della matrice A, normalizzato rispetto al massimo valore della colonna stessa di A, quindi:
  - $B[i][j] = A[i][j] / \text{MAX\_A\_j}$ , dove MAX\_A\_j rappresenta il valore massimo della colonna j-esima della matrice A, (i in [0,N-1] e j in [0,M-1]).
- **D.** Si stampi, sullo standard output, la matrice A e la matrice B;
- **E.** Mediante un qualunque algoritmo di ordinamento (insertion sort / bubble sort / selection sort), si ordini in modo crescente ogni colonna della matrice B, e la si stampi sullo standard output;

Il programma potrà essere strutturato in un unico file sorgente, ma dovrà contenere almeno le seguenti funzioni:

- **readInput:** funzione che prende in input l'array di puntatori a carattere argv della funzione main, controlla che gli argomenti richiesti siano nel numero e nei limiti specificati, e restituisce un record (struct) che contiene tali parametri; se il controllo non va a buon fine, stampa un messaggio sullo standard error e termina il programma;
- **allocateMatrix:** funzione che crea una matrice di puntatori a double delle dimensioni specificate tra i parametri formali;
- **fillMatrix:** funzione che legge il file di input e popola la corrispondente matrice di puntatori a double;
- **printMatrix:** funzione che stampa una matrice sullo standard output;
- **normalize:** funzione che, presa una matrice sorgente ed una matrice destinazione, popola la matrice destinazione sulla base di quanto esposto al punto C
- **sortMatrixCols:** funzione che ordina le colonne di una matrice in input in modo crescente.

**E' VIETATO usare variabili globali.**

**Durata della prova:** 120 minuti

**INPUT di esempio:**

(contenuto del file input.txt, che si trova nella home directory)

```
12.3 1.2 6.5 7.8 9.8 3.4
8.54 0.34 9.2 8.6 1.2 45.3
1.2 3.4 5.6 7.8 9.1234 6.123
21.345 7.234 8.231 9.321 5.123 9.123
3.54 1.34 3.2 5.6 0.2 12.3
```

**OUTPUT DI CONTROLLO** (file output.txt, che si trova nella home directory):

```
$ ./a.out input.txt 5 6
```

**Output:**

```
** A **
12.300000 1.200000 6.500000 7.800000 9.800000 3.400000
8.540000 0.340000 9.200000 8.600000 1.200000 45.300000
1.200000 3.400000 5.600000 7.800000 9.123400 6.123000
21.345000 7.234000 8.231000 9.321000 5.123000 9.123000
3.540000 1.340000 3.200000 5.600000 0.200000 12.300000
** B **
0.576247 0.165883 0.706522 0.836820 1.000000 0.075055
0.400094 0.047000 1.000000 0.922648 0.122449 1.000000
0.056219 0.470003 0.608696 0.836820 0.930959 0.135166
1.000000 1.000000 0.894674 1.000000 0.522755 0.201391
0.165847 0.185236 0.347826 0.600794 0.020408 0.271523
** B (sorted columns) **
0.056219 0.047000 0.347826 0.600794 0.020408 0.075055
0.165847 0.165883 0.608696 0.836820 0.122449 0.135166
0.400094 0.185236 0.706522 0.836820 0.522755 0.201391
0.576247 0.470003 0.894674 0.922648 0.930959 0.271523
1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
```