

Università di Catania  
Dipartimento di Matematica e Informatica  
Corso di Studio in Informatica, A.A. 2022-2023  
Programmazione 1 e Laboratorio  
Esempio di Prova di Esame Svolta  
10 gennaio 2023

## Descrizione del programma

Scrivere un programma in C che:

- **A** prenda come argomenti da riga di comando il nome di un file di testo di input "input" (es. "input.txt") e il nome di un file di testo di output "output" (es. "output.txt"). Si assuma che i nomi dei file non superano i 255 caratteri.
- **B** legga una matrice A di puntatori float di dimensioni  $n \times m$  dal file "input". Si assuma che il file contenga un numero di righe di testo pari a  $n$  e che ogni riga contenga  $m$  valori separati da spazi. Il programma dovrà inferire le dimensioni  $n$  e  $m$  dal file di input e riempire opportunamente i valori della matrice;
- **C** elimini dalla matrice i valori di A che sono superiori ai valori medi delle righe corrispondenti (far puntare a NULL i puntatori relativi ai valori identificati e liberare la relativa memoria);
- **D** inserisca i valori puntati dai puntatori non nulli in A in un array di float;
- **E** scriva i valori dell'array sul file "output" il cui nome è stato passato come argomento da riga di comando (es "output.txt").

## Specifiche

Il programma potrà essere articolato in un unico file sorgente, ma dovrà contenere almeno le seguenti funzioni con opportuni parametri formali:

- **readParameters**: funzione che permetta di leggere i parametri da riga di comando, verificando che rispettino i criteri specificati nel punto A, gestendo eventuali messaggi di errore e terminando il programma ove opportuno;
- **createMatrix**: funzione che permetta di creare la matrice A a partire dal nome del file di input;
- **sparsify**: funzione che permetta di mettere a NULL gli opportuni puntatori di A (come da punto C);
- **collect**: funzione che permetta di creare e riempire l'array di float come descritto nel punto D;
- **writeToFile**: funzione che permetta di scrivere i valori della lista su file.

## Note

**Durata della prova:** 120 minuti

**Generazione di numeri pseudocasuali:**

- Si consideri la seguente funzione `get_random()` per la generazione di numeri pseudo-casuali interi positivi (qualora necessaria):

```
// Scaricabile da: https://pastebin.com/f6eAKNqy
unsigned int get_random() {
    static unsigned int m_w = 123456;
    static unsigned int m_z = 789123;
    m_z = 36969 * (m_z & 65535) + (m_z >> 16);
    m_w = 18000 * (m_w & 65535) + (m_w >> 16);
    return (m_z << 16) + m_w;
}
```

- NB: Ai fini della generazione di numeri in virgola mobile, si faccia uso della costante `UINT_MAX` (<limits.h>) unitamente alla funzione `get_random()`.

```
}
```

- NB: Ai fini della generazione di numeri in virgola mobile, si faccia uso della costante `UINT_MAX` (`<limits.h>`) unitamente alla funzione `get_random()`.

**È VIETATO usare variabili globali.**

## Output di controllo

Dato il seguente file `input.txt`:

```
1.2  5.1  3.7 8.2  2.8
9.7 -8.2 -1.4 3.5  2.65
0.0  2.2  5.0 1.8 -0.3
```

Il programma dovrà generare il seguente file `output.txt`:

```
1.20
3.70
2.80
-8.20
-1.40
0.00
-0.30
```