

Università di Catania
Dipartimento di Matematica e Informatica
Corso di Studio in Informatica, A.A. 2022-2023
Prova di Laboratorio
17 Marzo 2023

Descrizione del programma

Scrivere un programma in C che:

- **A.** Prenda un input da tastiera (argomenti della funzione main) costituito da due numeri interi positivi **a e b in [5,20]**; un parametro in virgola mobile in doppia precisione **alpha in [10,20]**, ed un parametro in virgola mobile in doppia precisione **beta in [1,5]**. Il programma deve verificare che i parametri a, b, alpha e beta rispettino le specifiche e restituire in caso contrario un messaggio di errore.
- **B.** Si allochi dinamicamente una matrice M di puntatori a numeri in virgola mobile in doppia precisione di dimensioni a x b.
- **C.** Si popoli la matrice nel seguente modo.
 - Si estragga un numero pseudo-casuale in virgola mobile **p in [0,1]**;
 - se $p > 0.2$, si popoli l'elemento della matrice con un puntatore ad un numero pseudo-casuale appartenente all'intervallo $I = [\alpha - \beta, \alpha + \beta]$;
 - altrimenti, ovvero se $p \leq 0.2$ si ponga il puntatore a NULL.
- **D.** Si allochi un'ulteriore matrice M1 di puntatori a double di dimensioni b x a, che sia la **matrice trasposta** della matrice m. NB: per ogni elemento dalla trasposta si allochi nuovo spazio in memoria per un double, che potrà ospitare la copia del dato (il numero in virgola mobile).
- **E.** Si stampi infine la matrice M e la sua trasposta M1.

Specifiche

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, controlli che gli argomenti richiesti siano nei limiti specificati, e restituisca il record (struct) che contiene tali parametri; se il controllo non va a buon fine, stampa un messaggio sullo standard error e termina il programma.
- **allocMatrix:** funzione che alloca una matrice di puntatori a double delle dimensioni specificate mediante opportuni parametri formali e che restituisca tale matrice come dato di ritorno.

- **fillMatrix:** funzione che permetta di popolare una matrice di puntatori a double sulla base delle specifiche al punto C.
- **transposeMatrix:** funzione che prenda in input la matrice m e restituisca in output una nuova matrice m1 allocata dinamicamente che sia la trasposta di m (le righe di m1 saranno quindi popolate con le colonne di m).
- **printMatrix:** funzione che consenta di stampare una matrice di puntatori a double.

È VIETATO usare variabili globali.

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):

<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 eventuale generazione di numeri in virgola mobile, si faccia uso della costante UINT_MAX (<limits.h>) unitamente alla funzione get_random().

OUTPUT DI CONTROLLO:

./a.out 6 10 10 3

```
*** Matrice m (6 x 10) ***
12.55  7.05  8.67 10.89  9.34 12.10  8.11  7.15  7.84      N
      N      N  8.51  9.48  7.78  7.83  8.28  9.06  9.56      N
      N  7.69      N 11.71      N  8.35  7.43  7.80  8.48      N
10.70  8.45  8.43  7.65 11.28  9.88  7.07  8.46 12.35 12.57
 8.60  7.14 11.61  9.65 10.91 12.03 12.18 11.01 11.31 11.02
 7.86      N  7.92  7.81  7.74 11.16  9.86 11.50      N  9.74

*** Matrice m1 (10 x 6) ***
12.55      N      N 10.70  8.60  7.86
 7.05      N  7.69  8.45  7.14      N
 8.67  8.51      N  8.43 11.61  7.92
10.89  9.48 11.71  7.65  9.65  7.81
```

9.34	7.78	N	11.28	10.91	7.74
12.10	7.83	8.35	9.88	12.03	11.16
8.11	8.28	7.43	7.07	12.18	9.86
7.15	9.06	7.80	8.46	11.01	11.50
7.84	9.56	8.48	12.35	11.31	N
N	N	N	12.57	11.02	9.74