

Questa pagina riporta esempi di prove di esame con relative soluzioni per il corso di Programmazione 1 e Laboratorio 2022/2023 tenuto all'interno del corso di studio in Informatica, Università di Catania, a cura dei Proff. Giovanni Maria Farinella, Antonino Furnari, Fabrizio Messina

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

Si scriva un programma C che:

- **A** prenda un input da tastiera (argomenti della funzione main()) un intero k in [10,15], un carattere w in ['a'-'z'] e due interi N ed M in [4,8]; se gli argomenti a riga di comando non rispondono ai suddetti requisiti, il programma stampa un messaggio di errore sullo standard error e termina la propria esecuzione con codice di errore "-1";
- **B** allochi dinamicamente una matrice S di stringhe (char\*) di dimensioni N x M;
- **C** riempi la matrice S con NxM stringhe di lunghezza k composte di caratteri pseudo-casuali in [a-z];
- **D** ordini ogni colonna della matrice S in modo crescente (ordinamento lessicografico) con un algoritmo di ordinamento a scelta tra Insertion Sort e Bubble Sort;
- **E** stampi la matrice sullo standard output;
- **F** stampi sullo standard output la stringa (e gli indici all'interno della matrice) che contiene il maggior numero di occorrenze del simbolo w. Queste ultime vanno sostituite, sullo standard output, con il carattere '\*'.

## Specifiche

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

- **readInput**: funzione che prende in input il numero argc e il vettore argv della funzione main(), che controlli la presenza ed i requisiti degli argomenti k, w, N ed M, e che li inserisca in un record (struct) da restituire allo user code (funzione main). La funzione deve gestire correttamente gli errori relativi a input non corretti;

- **allocateS**: funzione per allocazione dinamica della matrice di dimensioni NxM, tale matrice va restituita come dato di ritorno al chiamante (funzione main());
- **genString**: funzione che restituisce una stringa della lunghezza specificata con caratteri pseudo-casuali in un ben determinato insieme specificato mediante opportuni parametri formali;
- **fills**: funzione di riempimento della matrice S come specificato nel punto C;
- **sortS**: funzione di ordinamento della matrice come specificato nel punto D; NB: si faccia uso, al suo interno, della funzione di libreria strcmp();
- **printMatrix**: funzione per la stampa della matrice S;
- **printMax**: funzione per la stampa della stringa contenente il maggior numero di occorrenze del simbolo w come specificato nel punto F.

## 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().

**È VIETATO usare variabili globali.**

## Output di controllo

Eseguendo il programma con il comando “./main 10 x 5 4”, esso dovrà produrre il seguente output:

```
nsmpvthnts bfdtfmjqqy eivkuwpadm jcughtergo
nwexgnpgdc ctrsulhgcx jlhxsmeokj lagozauhdg
sfgdnichgb riyourjykp lfjtetzycd lewtavgvyg
```

uvbayzwszq tywvyxgjwy quxsqbjzri tvqxaciajv  
xnmeouozlj xxxkqifjeu uwwhvqfhok xsgnebvjuy

Stringa con 3 occorrenze di 'x' trovata agli indici 4, 1:  
\*\*\*kqifjeu