

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 in input da tastiera (argomenti della funzione main) un intero positivo N in [10,20] ed un carattere w; 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;
- **B** generi, mediante successivi inserimenti, una lista concatenata semplice (ordinata in modo crescente - ordinamento lessicografico) che contenga N stringhe con caratteri pseudo-casuali in ['a'-'z'] di lunghezza pseudo-casuale L nell'intervallo [5,15];
- **C** stampi sullo standard output l'intera lista;
- **D** stampi sullo standard output il numero totale di occorrenze del carattere w in tutte le stringhe della lista.

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 prenda in input gli argomenti della funzione main argv ed argc, controlli i requisiti dei parametri e restituisca al chiamante un record (struct) che contenga tutti i parametri.
- **genString**: funzione che restituisca una stringa di caratteri pseudo-casuali appartenenti ad un determinato intervallo specificato mediante opportuni parametri formali.
- **insertString**: funzione che inserisca una stringa nella lista; per tale funzione e' possibile usare la funzione di libreria strcmp();
- **genList**: funzione che si occupa di creare e riempire la lista di stringhe come richiesto nel punto B;
- **printList**: funzione che si occupa di stampare la lista a schermo come specificato nel punto C;
- **printOcc**: funzione che stampa sullo standard output il numero totale di occorrenze del carattere w in tutte le stringhe della lista, come specificato nel punto D.

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.

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

- **readInput**: funzione che prenda in input gli argomenti della funzione main argv ed argc, controlli i requisiti dei parametri e restituisca al chiamante un record (struct) che contenga tutti i parametri.
- **genString**: funzione che restituisca una stringa di caratteri pseudo-casuali appartenenti ad un determinato intervallo specificato mediante opportuni parametri formali.
- **insertString**: funzione che inserisca una stringa nella lista; per tale funzione e' possibile usare la funzione di libreria strcmp();
- **genList**: funzione che si occupa di creare e riempire la lista di stringhe come richiesto nel punto B;
- **printList**: funzione che si occupa di stampare la lista a schermo come specificato nel punto C;
- **printOcc**: funzione che stampa sullo standard output il numero totale di occorrenze del carattere w in tutte le stringhe della lista, come specificato nel punto D.

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/f6eAKN0y
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 14 c, esso dovrà stampare il seguente output:

```
dnichgbctrsulhg
ebvju
hterg
hxsmekjlag
nsmptvthnt
tywvyxgjwy
wexgnpgd
wwhvqfhokjcu
xnmeou
xxxkqifjeulfjte
zauhdgsf
zeivkuwpa
zljriyourjykpj
zycdxxg
Numero totale di occorrenze di c: 4
```