

Università di Catania
Dipartimento di Matematica e Informatica
Corso di Studio in Informatica, A.A. 2022-2023
Compito di Programmazione 1 e Laboratorio F-N
23 febbraio 2023

Descrizione del programma

Si scriva un programma C che:

- A. Prenda in input da riga di comando un parametro stringa *alfabeto*, un intero n e un nome di file *output* (ad esempio “file_di_output.txt”). Il programma controlla che la stringa *alfabeto* abbia una lunghezza compresa tra 5 e 15 caratteri (inclusi) e che l'intero n sia compreso tra 5 e 20 (inclusi). Se i parametri passati non rispettano i requisiti richiesti, il programma stampa un messaggio di errore sullo standard error e termina la sua esecuzione con un appropriato codice di terminazione.
- B. Chieda all'utente di inserire n interi x (separati da caratteri di invio) da standard input. Il programma aggiorna il valore di x usando la formula:
$$x = \min(\max(1, x), L)$$
Il programma inserisce dunque ciascun valore x (aggiornato con la formula sopra) all'interno di un array W di lunghezza n (gli elementi vanno inseriti nello stesso ordine in cui vengono letti da standard input).
- C. Costruisca un array di stringhe Q di lunghezza n il cui i -esimo elemento $Q[i]$ sia una stringa di lunghezza $W[i]$ di caratteri casuali estratti dalla stringa *alfabeto*.
- D. Inizializzi una pila vuota, scorra gli elementi di Q nell'ordine in cui essi appaiono e il inserisca nella pila. Prima di ciascuna operazione di inserimento (ad eccezione del primo inserimento), il programma controlla se la stringa $Q[i]$ da inserire ha lunghezza dispari. In tal caso, il programma estrae la stringa che si trova in cima alla pila, la concatena con la stringa $Q[i]$ e inserisce nella pila la stringa risultato della concatenazione.
- E. Salvi il contenuto della pila sul file il cui nome è indicato dal parametro “output”.

Nota: gestire opportunamente i casi in cui i file non possono essere correttamente aperti in lettura o scrittura stampando un errore sullo standard error e terminando l'esecuzione del programma.

Specifiche

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

- **decodeParameters**: funzione che prende in input il numero argc e il vettore argv della funzione main(), controlli la presenza e i requisiti degli argomenti e 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;
- **readInput**: funzione che legge l'input da tastiera e restituisce l'array W come definito nel punto B del testo;
- **sampleString**: funzione che prende in input la stringa *alfabeto* e un intero h e restituisca una stringa di h caratteri casuali campionati dalla stringa *alfabeto*.

- **getStringArray:** funzione che prende in input l'array W e la stringa *alfabeto* e permette di ottenere l'array di stringhe Q come specificato nel punto C.
- **getStack:** funzione che prende in input l'array Q e permette di ottenere lo stack come specificato nel punto D.
- **writeStackToFile:** funzione per la scrittura del contenuto dello stack su file come specificato nel punto E.

Note

- **Durata della prova:** 120 minuti
- **È VIETATO** usare variabili globali.
- **Si inseriscano i file sorgenti** direttamente nella propria home directory .
- **Accesso alla documentazione** disponibile tramite il browser al link: <https://devdocs.io/c/>.
- **Per la generazione di numeri casuali si utilizzi la funzione `get_random` fornita.**

Output di controllo

Si consideri il seguente file “input” (troverete il file nella vostra home directory):

```
6
4
-1
3
10
6
4
5
10
2
```

Eseguendo il programma con il comando:

```
./soluzione xywldbdkm 10 out.txt < input
```

il programma scriverà il seguente file `out.txt`:

```
mk
bdyydysdxw
byddbkbdkm
wykwdd
mklsldbdkk
llbxkdkx
klywmm
```