

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
05 dicembre 2023

Descrizione del programma

Si scriva un programma C che:

- A. Prenda in input da riga di comando un parametro stringa *simboli*, un intero n e un nome di file *output* (ad esempio “file_di_output.txt”). Il programma controlla che la stringa *simboli* abbia una lunghezza L compresa tra 10 e 20 caratteri (inclusi) e che l'intero n sia compreso tra 8 e 18 (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.

Si stampino i parametri presi in input.

- B. Chieda all'utente di inserire n interi x (separati da caratteri di invio) da standard input. Il programma calcola il valore:

$$y = x! = x(x - 1)(x - 2) \dots 1$$

Il programma inserisce dunque ciascun valore y all'interno di un array W di lunghezza n (gli elementi vanno inseriti nello stesso ordine in cui vengono letti da standard input).

Si stampi il contenuto di W .

- C. Costruisca un array di stringhe Q di lunghezza n il cui i -esimo elemento $Q[i]$ sia una stringa di lunghezza $W[i] \bmod L$ di caratteri casuali estratti dalla stringa *simboli*, dove L è la lunghezza della stringa *simboli* e “mod” rappresenta l'operazione modulo.

Si stampi il contenuto di Q .

- D. Inizializzi una pila vuota, scorra gli elementi di $Q[i]$ dell'array Q nell'ordine in cui essi appaiono. Per ciascuna stringa $Q[i]$, il programma controlla se la sua lunghezza h è pari. Se h è pari, o se si tratta del primo inserimento in pila, il programma inserisce regolarmente la stringa $Q[i]$ in cima alla pila. Se h è dispari, il programma estrae la stringa che si trova in cima alla pila con una operazione di pop (sia tale stringa a) e inserisce nella pila la stringa b contenente i caratteri di $Q[i]$ che appaiono in a , nell'ordine in cui appaiono in $Q[i]$.

Si stampi il contenuto della pila.

- 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 *simboli* e un intero h e restituisca una stringa di h caratteri casuali campionati dalla stringa *simboli*.
- **getStringArray**: funzione che prende in input l'array W e la stringa *simboli* 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):

```
2
5
3
5
2
3
4
2
3
3
```

Eseguendo il programma con il comando:

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

il programma genererà il seguente output:

Punto A - Parametri: `simboli = jshdywjdufhfghdywb`,
`output=out.txt`, `n = 10`

Punto B - Contenuto di W :

```
2 120    6 120    2    6 24    2    6    6
```

Punto C - Contenuto di Q:

gh
hufwssfddsjsjg
dhdhfw
gujwyydggyff
ww
dhsywf
dfdgsh
hd
wdhbdu
fjjhfu

Punto D - Contenuto della pila:

fjjhfu
wdhbdu
hd
dfdgsh
dhsywf
ww
gujwyydggyff
dhdhfw
hufwssfddsjsjg
gh

e scriverà il seguente file out.txt:

fjjhfu
wdhbdu
hd
dfdgsh
dhsywf
ww
gujwyydggyff
dhdhfw
hufwssfddsjsjg
gh