

Università di Catania  
Dipartimento di Matematica e Informatica  
Corso di Studio in Informatica, A.A. 2023-2024  
Compito di Programmazione 1 e Laboratorio F-N  
26 giugno 2024

### Descrizione del programma

Si scriva un programma C che:

- A. **[5 punti]** Prenda in input da riga di comando un nome di file *input* (ad esempio “file\_di\_input.txt”), un intero  $n$  e un nome di file *output* (ad esempio “file\_di\_output.txt”). Il programma controlla che il file di input contenga una stringa di lunghezza compresa tra 8 e 12 caratteri (inclusi) e che l'intero  $n$  sia compreso tra 3 e 12 (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.  
Nota: se il file di input contiene più di una stringa, si consideri solo la prima, scartando eventuali simboli di separazione tra le stringhe come spazi o a capo.  
*Si stampino a schermo i valori dei parametri presi in input e la stringa contenuta nel file di input.*
- B. **[5 punti]** Costruisca un array  $A$  di lunghezza  $n$  e inserisca al suo interno i primi  $n$  numeri della sequenza di Fibonacci (si consideri  $F_0 = 0, F_1 = 1, F_2 = 1, \dots$ ).  
*Si stampi a schermo il contenuto dell'array  $A$ .*
- C. **[7 punti]** Costruisca un array di stringhe  $B$  di lunghezza  $n$  il cui  $i$ -esimo elemento  $B[i]$  sia una stringa di lunghezza  $A[i]$  di caratteri casuali estratti dalla stringa *alfabeto* contenuta all'interno del file *input*.  
*Si stampi a schermo il contenuto dell'array  $B$ .*
- D. **[10 punti]** Inizializzi una coda vuota, scorra gli elementi di  $B$  nell'ordine in cui essi appaiono e li inserisca nella coda. Prima di ciascuna operazione di inserimento (ad eccezione del primo inserimento), il programma controlla se la stringa  $B[i]$  da inserire ha lunghezza dispari. In tal caso, il programma estrae una stringa dalla coda, la concatena con la stringa  $B[i]$  e inserisce nella coda la stringa risultato della concatenazione. Nel caso in cui la stringa  $B[i]$  da inserire ha lunghezza pari, essa viene inserita regolarmente nella coda.  
*Si stampi a schermo il contenuto della coda.*
- E. **[3 punti]** Svuoti la coda conservandone il contenuto 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;

- **fibonacci:** funzione che restituisce la funzione di Fibonacci come richiesto nel punto B;
- **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 *A* e la stringa *alfabeto* e permette di ottenere l'array di stringhe *B* come specificato nel punto C.
- **getQueue:** funzione che prende in input l'array *B* e permette di ottenere la coda *Q* come specificato nel punto D.
- **writeToFile:** funzione per la scrittura del contenuto della pila 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 di seguito**

```
unsigned int get_random() {
    static unsigned int m_w = 424242;
    static unsigned int m_z = 242424;
    m_z = 36969 * (m_z & 65535) + (m_z >> 16);
    m_w = 18000 * (m_w & 65535) + (m_w >> 16);
    return (m_z << 16) + m_w;
}
```

## Output di controllo

Si consideri il file “input” con il seguente contenuto (crearlo nella home directory):

abcdefghijkl

Eseguendo il programma con il comando:

```
./soluzione input 9 output
```

il programma scriverà il seguente file output :

```
ialidgdf
egiicbicehfbhllddi
hfifel fhbbbebgeaclilfgdfaifbc
```

E produrrà il seguente output di controllo:

=====A=====

Parameters: inputFile=input, n=9, outputFile=output

Input string: abcdefghil

=====B=====

Fibonacci array: 0 1 1 2 3 5 8 13 21

=====C=====

String array B:

i

d

bc

lld

dfaif

ialidgdf

egiicbicehfbh

hfifelafhbbbebgeaclilfg

=====D=====

Queue:

ialidgdf

egiicbicehfbhllddi

hfifelafhbbbebgeaclilfgdfaifbc