

Università di Catania
Dipartimento di Matematica e Informatica
Corso di Studio in Informatica, A.A. 2025-2025
Compito di Programmazione 1 e Laboratorio F-N
19 febbraio 2025

Descrizione del programma

Si scriva un programma C che:

- A. **[5 punti]** Prenda in input da riga di comando il nome di un file di output (es. *output.txt*). Il programma controlli che tale nome di file abbia estensione “.txt”. Il programma legga dunque da standard input una stringa *alfabeto* di lunghezza compresa tra 8 e 12 caratteri (inclusi) e un intero n 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.
Si stampino a schermo i valori letti in input.
- B. **[5 punti]** Costruisca un array A di lunghezza n e inserisca al suo interno il resto della divisione per 101 del fattoriale dell'indice corrispondente: $A[i]=i! \% 101$.
Si stampi a schermo il contenuto dell'array.
- C. **[7 punti]** Costruisca un array di stringhe B di lunghezza n in cui l' i -esimo elemento sia una stringa di lunghezza $A[i]$ di caratteri casuali estratti dalla stringa *alfabeto* fornita in input. Si calcoli dunque la lunghezza media delle stringhe in B .
Si stampi a schermo il contenuto dell'array B e la lunghezza medi calcolata.
- D. **[10 punti]** Inizializzi una pila vuota, scorra gli elementi di B nell'ordine in cui essi appaiono e li inserisca nella pila. Prima di ciascuna operazione di inserimento (ad eccezione del primo inserimento), il programma controlla se la stringa da inserire ha un numero di caratteri superiore o inferiore alla lunghezza media delle stringhe calcolata nel punto precedente. Se la stringa ha una dimensione superiore alla media, viene estratto un elemento dalla pila e concatenato con la stringa corrente, quindi la stringa risultante viene inserita nella pila. Se la stringa ha una dimensione inferiore o uguale alla media (o se si tratta del primo inserimento), la stringa viene inserita regolarmente nella pila.
Si stampi a schermo il contenuto della pila.
- E. **[3 punti]** Svuoti la pila conservandone il contenuto sul file di output il cui nome è stato indicato dall'utente.

Nota: Gestire opportunamente i casi in cui il file non possa essere correttamente aperto in 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:

- **decodeInput:** funzione che prende in input la stringa *alfabeto* e l'intero n , controlla la validità dei parametri e li restituisce in una struct. La funzione deve gestire correttamente gli errori relativi a input non corretti.

- **factorialModArray**: funzione che restituisce l'array calcolato come richiesto nel punto B.
- **sampleString**: funzione che prende in input la stringa alfabeto e un intero e restituisce una stringa di caratteri casuali campionati dalla stringa alfabeto.
- **getStringArray**: funzione che prende in input l'array e la stringa alfabeto e permette di ottenere l'array di stringhe come specificato nel punto C.
- **computeAverageLength**: funzione che calcola la lunghezza media delle stringhe contenute in .
- **getStack**: funzione che prende in input l'array e permette di ottenere la pila 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
- **Implementare le strutture dati dinamiche mediante puntatori (non mediante array)**
- **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 (lo trovate nella home):

```
abcdefghil 4
```

Eseguendo il programma con il comando `./soluzione output.txt < input`, l'output di controllo è il seguente:

```
=====A=====
File di output: output.txt
Stringa alfabeto: abcdefghil, Intero n: 4
=====B=====
A[0] = 1
A[1] = 1
A[2] = 2
A[3] = 6
=====C=====
B[0] = i
B[1] = d
```

```
B[2] = bc
B[3] = llddfa
Lunghezza media: 2.50
=====D=====
bc11ddfa
d
i
=====E=====
File scritto con successo.
```

Il file `output.txt` conterrà il seguente contenuto:

```
bc11ddfa
d
i
```