

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 un input da tastiera (argomenti della funzione main) costituito da un intero positivo N. Il programma deve verificare che N sia inserito nel formato corretto e restituire un messaggio di errore terminando il programma opportunamente nel caso in cui il formato di N non dovesse essere corretto;
- **B** generi una sequenza di N operazioni di inserimento (push) di caratteri pseudo-casuali in [A-Z,a-z,1-9] in una struttura dati LIFO dinamica (pila o stack) da implementare mediante lista concatenata semplice (top==testa della lista), nel seguente modo:
 - ad ogni passo, si generi innanzitutto un carattere x in [1-9];
 - se x rappresenta un numero in [1-4], allora si proceda ad x operazioni di inserimento (push) di caratteri pseudo-casuali che siano vocali, seguite dall'inserimento del carattere x;
 - se x rappresenta un numero in [5-9], allora si proceda ad x operazioni di inserimento (push) di caratteri pseudo-casuali che siano consonanti, seguite dall'inserimento del carattere x;
 - in particolare, sia c il generico carattere da inserire sullo stack:
 - se c==v, si inserisca sullo stack al posto di esso il carattere '*';
 - se c==w, si inserisca sullo stack al posto di esso il carattere '?';
- **C**
 - crei un array di stringhe (puntatori a caratteri) di lunghezza N;
 - proceda nel seguente modo, fino a svuotamento dello stack: successivamente proceda ad una sequenza di operazioni di rimozione (pop()) come segue:
 - si proceda quindi, ad ogni passo con un'operazione di rimozione (pop()) del carattere x che indica (per costruzione) il numero di caratteri da rimuovere successivamente mediante x operazioni pop();
 - si memorizzi sull'array di caratteri ogni stringa composta dal carattere x e dai successivi x caratteri;
- **D** stampi, sullo standard output, il contenuto dell'array di stringhe;

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 prende in input l'array di puntatori a carattere argv della funzione main, controlli che gli argomenti richiesti siano nei limiti specificati, e restituisca il record (struct) che contiene tali parametri; se il controllo non va a buon fine, stampa un messaggio sullo standard error e termina il programma.
- **genVowel**: funzione che produca un carattere vocale pseudo-casuale;
- **genConsonant**: funzione che produca un carattere consonante pseudo-casuale;
- **push e pop**: funzioni che consentono di inserire un elemento sullo stack o rimuovere un elemento da esso;
- **fillStack**: funzione con opportuni parametri formali che rappresenti l'implementazione della procedura descritta nel punto B;
- **emptyStack**: funzione con opportuni parametri formali che sia conforme alla procedura descritta nel punto C;
- **printArray**: funzione per la stampa dell'array di stringhe prodotto al punto C.

- `printArray`: funzione per la stampa dell'array di stringhe prodotto al punto C.

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.

Output di controllo

Il programma eseguito mediante il seguente comando: `./main 12` dovrà restituire il seguente output:

```
R*tKm
Iou
SyDZMh
o
LZfWB
KMjzCNsPR
GtVcSGGXt
iIIE
XpcQXLqQ?
qKhWXfjH
y?QsSSX
q?VJMSQ1
```