

Programmazione I
Prova di programmazione – 21 febbraio 2018 – 2 ore

Partendo dal frammento di codice fornito, realizzare un programma di gestione di una sequenza di parole. La lunghezza massima N della sequenza e la lunghezza massima L di ciascuna parola sono stabilite a tempo di scrittura del programma, ma la sequenza deve essere implementata in maniera tale per ciascuna parola inserita nella sequenza si occupi **solo la memoria necessaria** per memorizzare i caratteri di tale parola, più un'eventuale terminatore. Indipendentemente dal fatto che si riesca effettivamente a realizzare il punto 5, si ottiene il punteggio massimo se si implementa una **struttura dati** che permette di realizzare la modifica della sequenza nel punto 5 **nel modo più semplice possibile**, ossia con un solo ciclo non nidificato (più eventualmente altro semplice codice al contorno). Tale struttura dati può rendere il punto 1 molto più inefficiente e difficile realizzare, rispetto ad altre possibili strutture dati. E' accettato anche l'utilizzo di una doppia struttura dati, nel caso vi agevoli nello svolgimento del compito.

All'avvio del programma la sequenza è vuota. Il programma deve fornire le seguenti funzionalità.

1. **inserisci_parola(p)** Inserisce la parola **p** in fondo alla sequenza.
2. **stampa_sequenza** Stampa il contenuto della sequenza, una parola per riga. Ad esempio:
topolino
pippo
minnie
3. **salva_sequenza** Salva la sequenza in un file di testo, dal nome definito a tempo di scrittura
4. **carica_sequenza** Carica la sequenza dal file. L'eventuale precedente contenuto è perso.
5. **sposta_in_prossima(idx, n)** Solo se tutte le parole a partire dall'indice **idx** hanno lunghezza strettamente superiore ad **n**, allora effettua la seguente operazione su tutte le parole a partire da quella di indice **idx** e successive nella sequenza: elimina gli ultimi **n** caratteri della parola e, se la parola non è l'ultima nella sequenza, li aggiunge in testa alla parola successiva. Ad esempio, se questa funzionalità è invocata con **idx = 0** ed **n = 4** sulla sequenza all'esempio al punto 2, allora la sequenza diviene
topo
linop
ippomi
invece se invocata con **idx = 1** ed **n = 2**
topolino
pip
pominn

I parametri di ingresso delle funzionalità sono solo indicativi. Gestire opportunamente le situazioni di errore, tranne l'*overflow* e l'inserimento di dati in formato errato o di parole troppo lunghe da *stdin*.

REGOLE

- Si può utilizzare ogni genere di manuale e di materiale didattico
- Per superare la prova, bisogna svolgere almeno i punti 1 e 2. Se si svolgono solo tali punti, il programma deve essere perfettamente funzionante. Il voto ottenuto in questo caso è 18.
- Ciascuna funzionalità DEVE essere implementata mediante almeno una funzione.
- Il voto massimo (almeno 30) si ottiene se
 - a) il programma è perfettamente funzionante in ogni sua parte
 - b) tutti i principi di ingegneria del codice visti nel corso sono stati applicati
 - c) sono state seguite eventuali altre indicazioni presenti nella traccia in merito al voto finale