

Programmazione I
Prova di programmazione – 14 Gennaio 2021 – 2 ore

Partendo dal frammento di codice fornito, realizzare un programma per la gestione di una coda di utenti. Ogni utente è etichettato con il codice del servizio per il quale è in attesa. I codici dei servizi sono singoli lettere maiuscole. All'avvio del programma, la coda è vuota. Il programma fornisce le seguenti funzionalità.

1. **inizializza_coda(N)** Inizializza la coda a vuota ed a contenere al massimo N utenti. L'eventuale precedente contenuto della coda è perso. Il valore massimo possibile per N non è noto a tempo di scrittura del programma. Se $N = 0$, allora la coda avrà lunghezza massima nulla.
2. **[3 punti extra] accoda_utente(c)** Accoda un utente per il servizio di codice c . Ad esempio, se la coda contiene due utenti, per i codici J ed A , e si accoda un utente per il codice R , allora la coda conterrà utenti per i codici J , A ed R . Il prossimo utente da servire sarà quello di codice J , poi sarà servito A , ed infine R .
Si ottengono i 3 punti in più se si realizza questa funzionalità a costo $O(I)$ rispetto al numero di utente, ed in maniera tale che anche l'estrazione dalla testa della coda abbia costo $O(I)$.
3. **stampa_coda** Stampa la coda, scrivendo, per ciascun utente solo il codice del servizio per cui l'utente è in attesa. I codici sono separati da spazi. Ad esempio, dopo l'inserimento dell'utente con codice R nell'esempio al punto 2, stamperebbe
J A R
4. **[2] salva_coda** Salva la coda in un file di testo dal nome predefinito.
5. **[3] carica_coda** Carica la coda dal file. L'eventuale precedente contenuto è perso.
6. **[1, 3] servi_prossimo_utente** Elimina dalla coda l'utente in testa. Si ottengono tre punti se si realizza questa funzionalità a costo $O(I)$.
7. **[4] anticipa_servizio(c)** Sposta in testa alla coda tutti gli eventuali utenti del servizio c , e lasciando inalterato l'ordine tra gli altri utenti. Ad esempio, se si invocasse questa funzionalità con $c=A$ per la seguente coda
A J A R A
allora la coda diverrebbe
A A A J R

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 da *stdin*.

REGOLE

- Si può utilizzare ogni genere di manuale e di materiale didattico
- Per superare la prova, bisogna svolgere almeno i punti 1, 2 e 3. 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