

Programmazione I

Prova di programmazione – 12 gennaio 2017 – 2 ore

Partendo dal frammento di codice fornito, realizzare un programma per la gestione di M code di utenti, in cui ciascuna coda è associata ad un qualche *servizio* da fornire a tali utenti. Ciascuna coda è individuata da un indice numerico da 0 ad $M-1$. Per usufruire del servizio i -esimo, un utente deve entrare nella coda i -esima. Quando entra nella coda i -esima, l'utente si vede assegnato un numero intero positivo, che chiamiamo *numero di servizio*. Al primo utente che entra in una coda è assegnato il numero 1, mentre ad ogni utente successivo che entra nella stessa coda è assegnato il numero di servizio associato al precedente utente entrato nella coda, più uno. Un utente esce dalla coda quando viene servito. Gli utenti sono serviti in ordine crescente di numero di servizio.

All'avvio del programma non c'è nessuna coda. Ciascuna coda può contenere al più N utenti, con N definito a tempo di scrittura del programma. Il programma deve fornire le seguenti funzionalità.

1. **configura_code(M)** Se M è maggiore di zero, configura il programma per gestire M code, inizializzando ciascuna coda a vuota. L'eventuale precedente numero e stato delle code è perso. Assumere che **il valore massimo possibile per M non sia noto** a tempo di scrittura del programma.
2. **accoda_utente(i)** Accoda un utente nella coda i -esima, ritornando il numero di servizio per tale utente.
3. **servi_prossimo(i)** Serve il prossimo utente per la coda i -esima. Simuliamo il servizio di un utente con l'esecuzione delle sole due operazioni seguenti: ritorno del numero di servizio dell'utente, estrazione dell'utente dalla coda.
4. **stampa_stato** Stampa lo stato delle code. Per ciascuna coda, stampa: l'indice della coda, il numero di utenti accodati, il numero di servizio dell'ultimo utente accodato ed il numero di servizio del prossimo utente da servire. Ad esempio, se vi fossero tre code, un possibile output potrebbe essere
2 1 1 1
0 3 4 2
1 0 0 1
5. **salva_stato** Salva lo stato delle code in un file di testo dal nome definito a tempo di scrittura del programma.
6. **carica_stato** Carica lo stato delle code dal file. Lo stato precedente è perso.
7. **attiva_modo_numeri_limitati** Se questa funzionalità è invocata, le code vengono svuotate, ed una nuova modalità di funzionamento delle funzionalità 2 e 3 viene attivata. In questa nuova modalità, le funzionalità 2 e 3 calcolano e gestiscono i numeri di servizio in maniera tale che tali numeri siano sempre compresi tra 1 ed N , ove N è la lunghezza massima di ciascuna coda.

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 e di stringhe troppo lunghe da *stdin*.

REGOLE

- Si può utilizzare ogni genere di manuale o di materiale didattico di altra natura
- Per superare la prova, il programma deve essere perfettamente funzionante nelle parti 1 e 2. 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