

Esame 20240725

Esercizio 3

(1) Esercizio 3 v1

ESSAY marked out of 10 penalty 0 File picker

Data una struttura dati `Queue` che rappresenta una coda di interi (si veda il file `esercizio3.cpp`), e una serie di operazioni su questa struttura (e.g., `initQueue`, `isEmpty`, `enqueue`, `first`, `dequeue`, `length`, `deleteQueue`, `printQueue`), si vuole implementare una nuova funzione `calcola` che prende come argomento un puntatore a `Queue` `q`. Questa funzione **costruisci una nuova coda** in modo che gli elementi della coda `q` compaiono nello stesso ordine, e ogni elemento della coda `q` è seguito immediatamente dal numero di occorrenze dello stesso valore che lo precedono (incluso il valore stesso) nella coda `q`. Si assuma, per semplicità, che la coda `q` contenga interi compresi tra 0 e 9.

La funzione `calcola`

- **deve lasciare inalterato il contenuto** della coda `q` alla fine dell'esecuzione (*ma lo può modificare se ritenuto necessario per la realizzazione*).
- **NON deve creare strutture intermedie** (e.g., array, code, stack, liste, ...) **dove memorizzare il contenuto della coda** `q`. Valutare con attenzione le scelte implementative relative alle modalità di passaggio della coda `q` alla funzione `calcola`.
- **deve usare SOLO i metodi della coda** (i.e., `initQueue`, `isEmpty`, `enqueue`, `first`, `dequeue`, `length`, `deleteQueue`, `printQueue`) e **NON deve usare i dettagli implementativi della coda**, pena annullamento della prova.
- se ritenuto necessario è possibile definire funzioni ausiliarie (e.g., `reverse`), che operano sulla coda, ma che usino solo i metodi della coda, e che non usino strutture intermedie (vedi punti precedenti).

Il file `esercizio3.cpp` contiene l'implementazione della struttura `Queue`, di alcuni metodi di utilità, e un `main` con alcuni esempi e alcune invocazioni della funzione `calcola`. Di seguito è riportato l'output di esecuzione.

```
marco > ./a.out
Original before: 5 4 2 1
Result of calcola: 5 1 4 1 2 1 1 1
Original after: 5 4 2 1
Original before: 7 7 7 6 8 6 6 8 0 5 7
Result of calcola: 7 1 7 2 7 3 6 1 8 1 6 2 6 3 8 2 0 1 5 1 7 4
Original after: 7 7 7 6 8 6 6 8 0 5 7
Result of calcola: Queue is empty
Original after: Queue is empty
```

Note:

- Scaricare il file `esercizio3.cpp`, modificarlo per inserire la dichiarazione e la definizione della funzione `calcola`, e **caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`, `cstdlib`.

- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).

`esercizio3.cpp`

Information for graders:

(2) Esercizio 3 v2

ESSAY marked out of 10 penalty 0 File picker

Data una struttura dati `Queue` che rappresenta una coda di interi (si veda il file `esercizio3.cpp`), e una serie di operazioni su questa struttura (e.g., `initQueue`, `isEmpty`, `enqueue`, `first`, `dequeue`, `length`, `deleteQueue`, `printQueue`), si vuole implementare una nuova funzione `calcola` che prende come argomento un puntatore a `Queue` `q`. Questa funzione **costruisci una nuova coda** in modo che gli elementi della coda `q` compaiono nello stesso ordine, e ogni elemento della coda `q` è seguito immediatamente dal numero di occorrenze dello stesso valore che lo seguono (incluso il valore stesso) nella coda `q`. Si assuma, per semplicità, che la coda `q` contenga interi compresi tra 0 e 9.

La funzione `calcola`

- **deve lasciare inalterato il contenuto** della coda `q` alla fine dell'esecuzione (*ma lo può modificare se ritenuto necessario per la realizzazione*).
- **NON deve creare strutture intermedie** (e.g., array, code, stack, liste, ...) **dove memorizzare il contenuto della coda** `q`. Valutare con attenzione le scelte implementative relative alle modalità di passaggio della coda `q` alla funzione `calcola`.
- **deve usare SOLO i metodi della coda** (i.e., `initQueue`, `isEmpty`, `enqueue`, `first`, `dequeue`, `length`, `deleteQueue`, `printQueue`) e **NON deve usare i dettagli implementativi della coda**, pena annullamento della prova.
- se ritenuto necessario è possibile definire funzioni ausiliarie (e.g., `reverse`), che operano sulla coda, ma che usino solo i metodi della coda, e che non usino strutture intermedie (vedi punti precedenti).

Il file `esercizio3.cpp` contiene l'implementazione della struttura `Queue`, di alcuni metodi di utilità, e un `main` con alcuni esempi e alcune invocazioni della funzione `calcola`. Di seguito è riportato l'output di esecuzione.

```
marco > ./a.out
Original before: 5 4 2 1
Result of calcola: 5 1 4 1 2 1 1 1
Original after: 5 4 2 1
Original before: 7 7 7 6 8 6 6 8 0 5 7
Result of calcola: 7 4 7 3 7 2 6 3 8 2 6 2 6 1 8 1 0 1 5 1 7 1
Original after: 7 7 7 6 8 6 6 8 0 5 7
Result of calcola: Queue is empty
Original after: Queue is empty
```

Note:

- Scaricare il file `esercizio3.cpp`, modificarlo per inserire la dichiarazione e la definizione della funzione `calcola`, e **caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`, `cstdlib`.
- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta **NON** deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).

esercizio3.cpp

Information for graders:

Total of marks: 20