

Esame di Programmazione II

Appello di giorno 17 Dicembre 2014
Università degli Studi di Catania - Corso di Laurea in Informatica

Testo della Prova

Definizione Iniziale.

Un *MultiStack* è una struttura dati costituita da un insieme ordinato di Pile (Stack) non vuoti. Ogni stack è rappresentato dall'elemento che si trova in cima allo stack. Poiché tutti gli stack dell'insieme sono non vuoti tale elemento esiste sempre. L'insieme di stack è mantenuto ordinato in ordine non crescente rispetto all'elemento rappresentante di ogni stack.

Specifiche.

Si fornisca una classe C++, denominata `MyMultiStack<H>`, che implementi la seguente interfaccia `MultiStack<H>`, che rappresenta un `MultiStack` e contenente i seguenti metodi virtuali. L'insieme di stack doveà essere implementato attraverso l'utilizzo di un lista concatenata.

La corretta implementazione di ciascuno dei seguenti punti permette l'acquisizione di 7 punti. La corretta implementazione della classe come template è facoltativa e permette l'acquisizione di ulteriori 3 punti.

1. `H* pop()` restituisce l'elemento in cima al primo stack della struttura dati (il primo stack è quello che ha l'elemento in cima con valore più grande).
2. `void push(H x)` inserisce l'elemento x all'interno del primo stack della struttura dati;
3. `void newPush(H x)` aggiunge uno stack vuoto all'interno della struttura dati ed inserisce l'elemento x all'interno di questo stack;
4. `void print()` stampa la lista degli elementi della struttura dati. La procedura dovrà stampare gli elementi del primo stack, poi quelli del secondo, e così via, procedendo in ordine non crescente. Gli elementi di ogni stack dovranno essere racchiusi tra parentesi quadre.

Si crei quindi un'istanza di `MyMultiStack<int>` e si inseriscano al suo interno i valori 3, 7, 1 e 8 utilizzando la procedura `NEWPUSH`. Si inseriscano in seguito i valori 5,2,6,1,8,5,6,1,4 utilizzando la procedura `PUSH`. Infine si invochi 5 volte la procedura `POP`. Si esegua in seguito la stampa dei valori inseriti nella struttura dati. L'output del programma sarà quindi:

```
[5, 8, 3], [1,6,5,8], [1]
```

```
template <class H> class MultiStack {
public:
    virtual H pop() = 0;
    virtual void push(H x) = 0;
    virtual void newPush(H x) = 0;
    virtual void print() = 0;
}
```

