North South University

CSE 225: Data Structures & Algorithms

Assignment

The following header file (*dblist.h*) describes the functionality of a doubly linked list. Your job is to implement the functions in this header file in *dblist.cpp* and then test these functions in *main.cpp* (by creating a dblist object and then calling different functions on this object).

```
//dblist.h file:
#ifndef DLIST_H
#define DLIST_H

#include<iostream>
using namespace std;

class ListEmpty{};
class PositionOutOfBound {};

template<class T>
struct Dnode
{
    T data;
    Dnode<T> *prev,*next;
};

template<class T>
class DoublyLinkedList
{
protected:
    Dnode<T> *head, *tail, *curptr;
    int length;

public:
    DoublyLinkedList();
    ~DoublyLinkedList();

    bool isEmpty();
    int listSize();

    //print all items from leftmost node to rightmost node in this doubly linked list
    void print();
    //print all items from rightmost node to leftmost node in this doubly linked list
    void printReverse();

    //return position (starts from 1) of first occurrence of key in the list if it is
     //found, otherwise return 0
    int searchFirst(T key);

    //return position (starts from 1) of last occurrence of key in the list if it is
     //found, otherwise return 0
    int searchLast(T key);

    void insertAtStart(T value);
    void insertAtEnd(T value);

    void deleteStart();
    void deleteEnd();
    //if value is not found in the list,then this function does nothing
    //otherwise, it deletes the node that contains value in its data field.
    //The deleted node's previous node (say, p) and next node (say, n)
    //are updated as well so that after deletion, they become contiguous
    void deleteValue(T value);

    int getValue(int pos);
    void setValue(int pos, T val);

    bool hasNext();
    bool hasPrev();
    T nextItem();
    T prevItem();
    void resetAtHead();//move curptr to head
    void resetAtTail();//move curptr to tail
};
```