Data Structures and Algorithms

Lab 2 – Linked List

The following SingleLinkedList interface is applied to questions 1 to 4.

```cpp
struct Node {
    public:
        int data;  // value of list element
        Node *next; // pointer to next element of the list
};

class SingleLinkedList {
    public:
        Node *pHead; // pointer to the 1st node of the list

        SingleLinkedList () {
            pHead = NULL;
        }

        void prepend(int data) {
            Node *pNew = new Node();
            pNew->data = data;
            pNew->next = pHead;
            pHead = pNew;
            return;
        }

        void display() { // add your code here
        }

        void insert(int data, int idx) { // add your code here
        }

        Node *search(int target) { // add your code here
        }

        void remove(int target) { // add your code here
        }

        void extend(SingleLinkedList other) { // add your code here
        }
};
```
Question 1: Use the already implemented method prepend to construct linked list L1 as follow:
L1 = {1, 9, 6, 5, 7, 10, 13, 4, 8, 7}
Then, implement method display to check your results.

Question 2: Implement method insert to add a new node with value ‘data’ at a given index ‘idx’.
e.g.  // L2 = {1, 3, 2, 5, 6}
     L2.insert(4, 2) // data = 4, idx = 2
     // L2 = {1, 3, 4, 2, 5, 6}

Question 3: Implement method search to find a node with value ‘data’.
e.g.  // L3 = {1, 3, 2, 5, 6}
     Node *target = L3.search(5)
     // target->data is 5

Question 4: Implement method remove to delete ALL nodes with value ‘data’.
e.g.  // L4 = {1, 3, 2, 5, 6}
     L4.remove(3)
     // L4 = {1, 2, 5, 6}

Question 5: Implement method extend to join two linked list.
e.g.  // L5a = {1, 4, 7}
     // L5b = {9, 6, 5}
     L5a.extend(L5b)
     // L5a = {1, 4, 7, 9, 6, 5}
     // L5b = {9, 6, 5}

The following struct is used to form DoubleLinkedList

```cpp
struct Node {
    public:
        int data;
        Node *next;
        Node *prev;
}
```

Question 6: Create a new class DoubleLinkedList and implement the corresponding method insert and remove same as stated for SingleLinkedList.

Question 7: Implement method reverse for DoubleLinkedList.
e.g.  // L7 = {1, 3, 2, 5, 6}
     L7.reverse()
     // L7 = {6, 5, 3, 2, 1}