Experiment Title 3

Student Name: Anjali Singh **UID: 20BCS9239**
Branch: CSE **Section/Group: 607/A**
Semester: 5th **Date of Performance: 04/09/2022**
Subject Name: Competitive Coding **Subject Code: 20CSP-314**

1. Aim/Overview of the practical: To implement the concept of Linked list.

2. Task to be done/ Which logistics used: In this practical we are going understand various problems and find out better approach to solve particular problem related to Linked list.

3. Algorithm/Flowchart (For programming-based labs):

Compare two linked lists.

1. Start
2. We create a function to check identical
3. We compare the elements of Linked lists till last element which are at same position.
4. If both heads are NULL return true.
5. Compare each element data and move the pointers forward in both linked lists till it reaches the end.
6. End.

Inserting a node into a Sorted Doubly Link

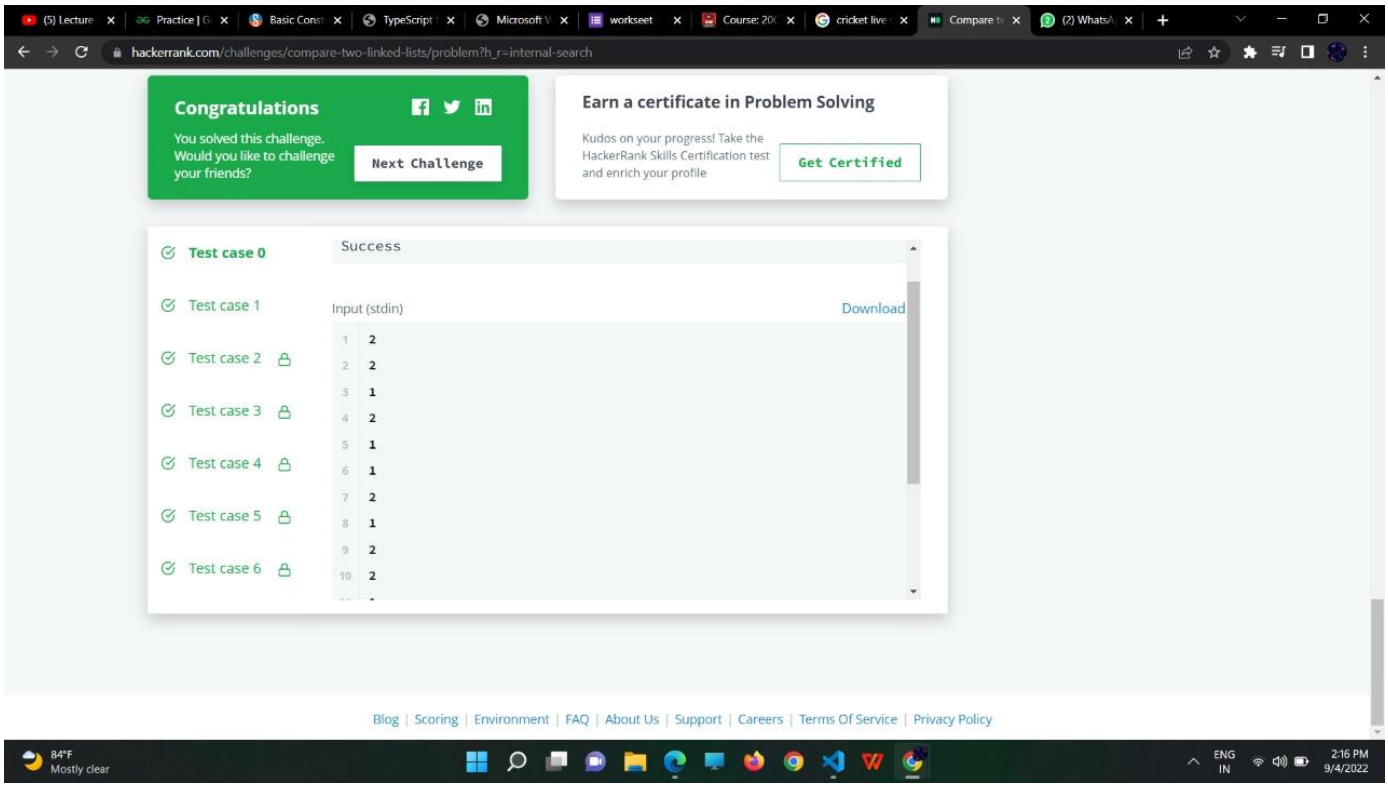
1. Start
2. Create a new node and allocate memory for new node and assign the data to new node.
3. Assign the value if next from previous node to the next of new node.
4. Assign the address of new node to the next of previous node.
5. Assign the value of prev of next node to the prev of new node.

6. Assign the address to new node to the prev of next node.
7. End.

4. Steps for experiment/practical/Code:

A. Compare two linked lists.

```
bool compare_lists(SinglyLinkedListNode *head1, SinglyLinkedListNode *head2)
{
    while (head1 != nullptr && head2 != nullptr && head1->data == head2->data)
    {
        head1 = head1->next;
        head2 = head2->next;
    }
    if (head1 == head2)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
```

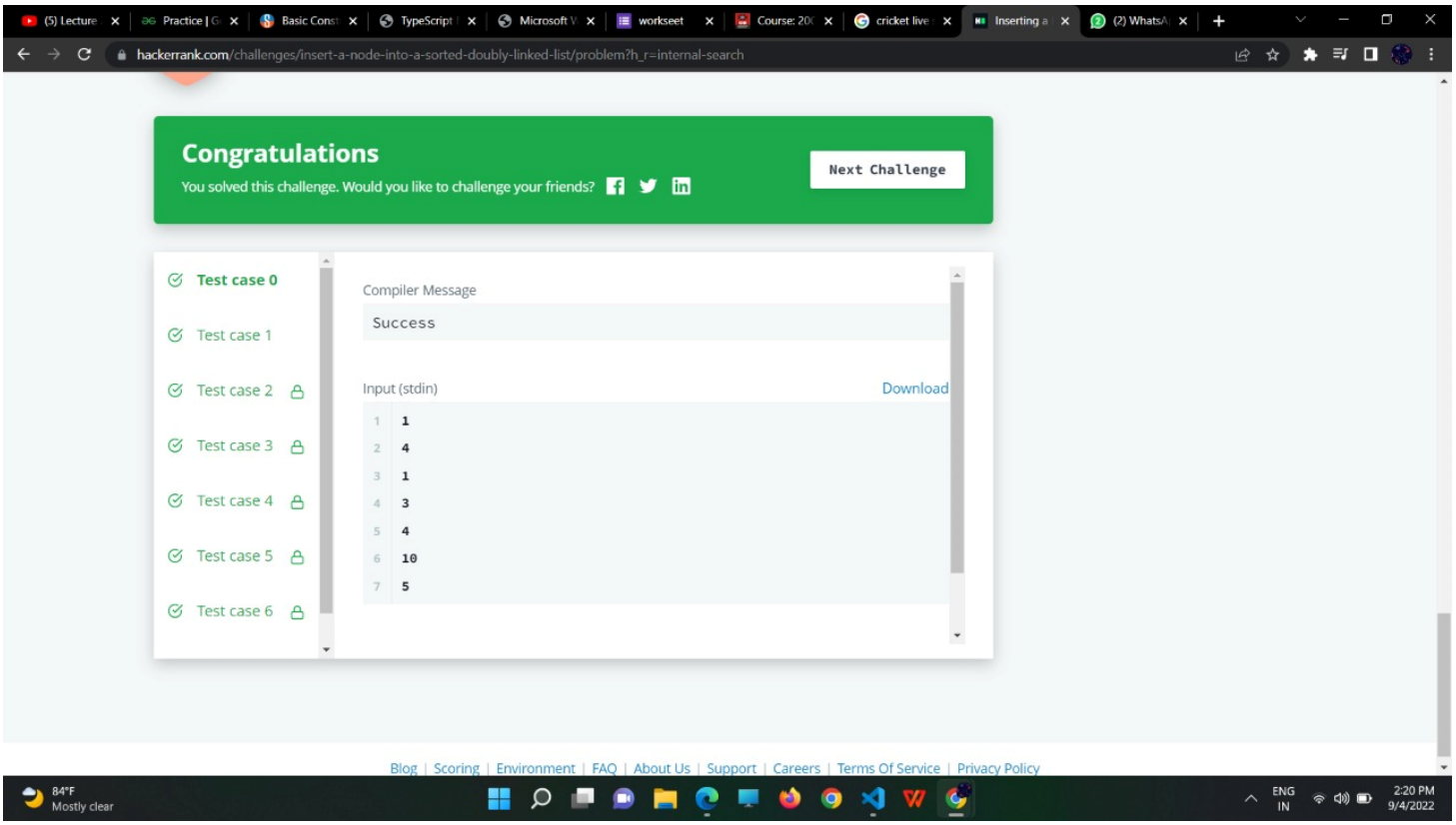


B. Inserting a node into a Sorted Doubly Link

```
DoublyLinkedListNode *sortedInsert(DoublyLinkedListNode *llist, int data)
{
    DoublyLinkedListNode *Node = new DoublyLinkedListNode(data);
    DoublyLinkedListNode *temp =
        llist;
    DoublyLinkedListNode *prev, *next;

    while (temp != NULL)
    {
        if (data < temp->data && temp->prev == NULL)
        {
            llist->prev = Node;
            Node->next = llist;
            llist = Node;
            return llist;
```

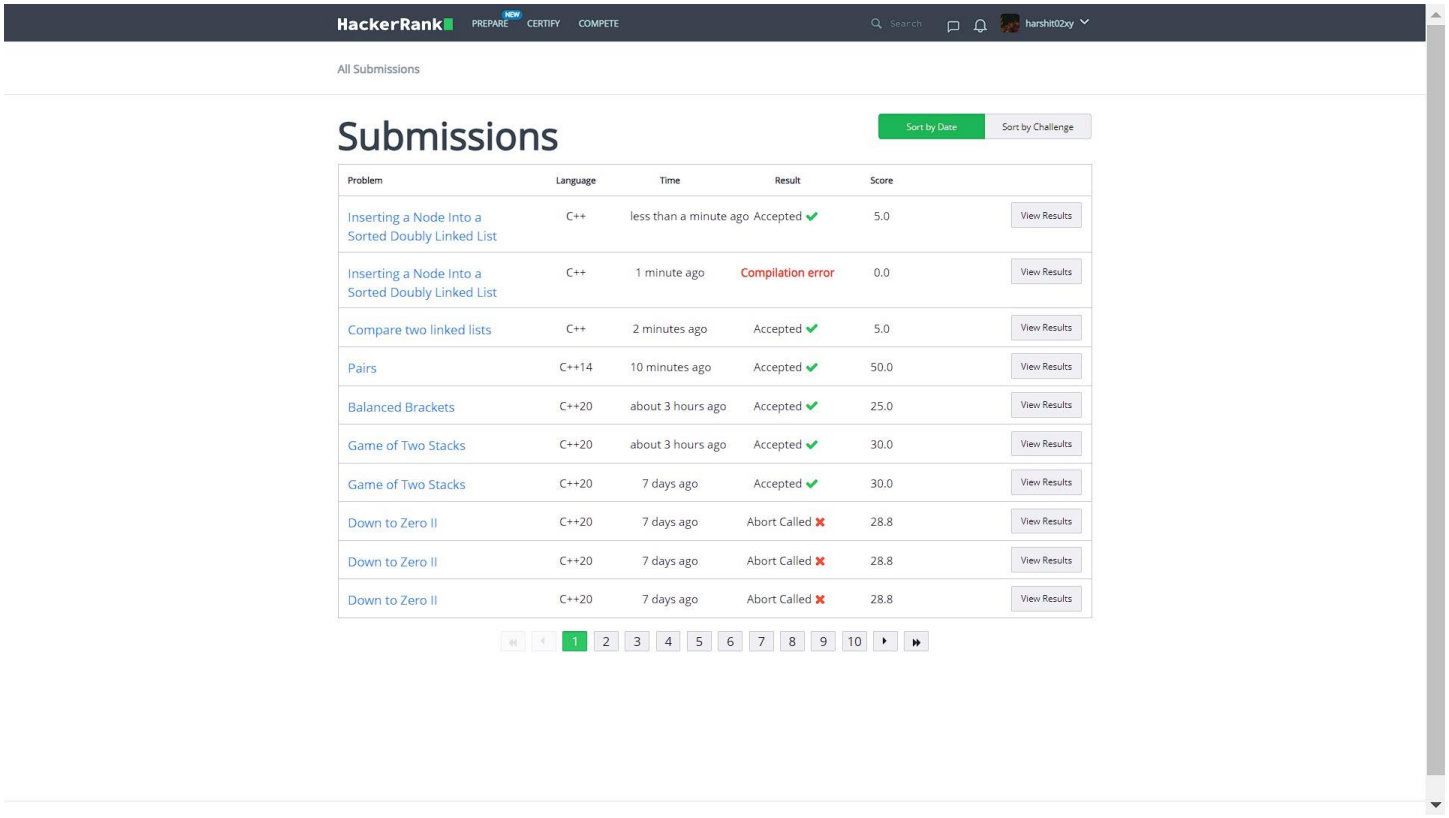
```
        }
        else if (data > temp->data && temp->next == NULL)
        {
            temp->next = Node;
            Node->prev = temp;
            return llist;
        }
        else if (data <= temp->data && temp->prev != NULL)
        {
            next = temp;
            temp = temp->prev;
            prev = temp;
            Node->next = next;
            Node->prev = prev;
            temp->next = Node;
            temp = next;
            temp->prev = Node;
            return llist;
        }
        temp = temp->next;
    }
    return llist;
}
```



5. Observations/Discussions/ Complexity Analysis:

Two Linked Lists are identical when they have the same data and the arrangement of data is also the same. For example, Linked lists a (1->2->3) and b (1->2->3) are identical. To identify if two lists are identical, we need to traverse both lists simultaneously, and while traversing we need to compare data. In order to insert a node with value X in a sorted list in sorted order, we need to traverse the list till the data of the next node of current is smaller than the given value X. After reaching the correct position, we will simply insert a new node with value X over there. This method is the same for both singly and doubly list. The only difference is that in Doubly Linked List, we have to take care of the prev pointer along with the next pointer.

6. Result/Output/Writing Summary:



HackerRank PREPARE CERTIFY COMPETE Search harshit02xy

All Submissions

Submissions

Sort by Date | Sort by Challenge

Problem	Language	Time	Result	Score	
Inserting a Node Into a Sorted Doubly Linked List	C++	less than a minute ago	Accepted ✔	5.0	View Results
Inserting a Node Into a Sorted Doubly Linked List	C++	1 minute ago	Compilation error	0.0	View Results
Compare two linked lists	C++	2 minutes ago	Accepted ✔	5.0	View Results
Pairs	C++14	10 minutes ago	Accepted ✔	50.0	View Results
Balanced Brackets	C++20	about 3 hours ago	Accepted ✔	25.0	View Results
Game of Two Stacks	C++20	about 3 hours ago	Accepted ✔	30.0	View Results
Game of Two Stacks	C++20	7 days ago	Accepted ✔	30.0	View Results
Down to Zero II	C++20	7 days ago	Abort Called ✖	28.8	View Results
Down to Zero II	C++20	7 days ago	Abort Called ✖	28.8	View Results
Down to Zero II	C++20	7 days ago	Abort Called ✖	28.8	View Results

1 2 3 4 5 6 7 8 9 10

Learning outcomes (What I have learnt):

1. How to create a linked list.
2. How to compare two linked list if they are equal or not.
3. How to create a doubly linked list.
4. How to add element in a doubly linked list.