Experiment : 1.3

Student Name: SANJIV GUPTA **UID:** 21BCS-*3478*
Branch: CSE **Section/Group:** 21BCS-IOT-602B
Semester: 5th **Date:** 01/09/23
Subject Name: Advanced Programming LAB **Subject Code:** 21CSP-314

AIM:

1. *WAP to compare two linkedlist*
2. *WAP to insert a node in sorted linketlist*

OBJECTIVE:

1) *You're given the pointer to the head nodes of two linked lists. Compare the data in the nodes of the linked lists to check if they are equal. If all data attributes are equal and the lists are the same length, return . Otherwise, return.*

2) *You're given the pointer to the head nodes of two linked lists. Compare the data in the nodes of the linked lists to check if they are equal. If all data attributes are equal and the lists are the same length, return . Otherwise, return.*

CODE:

Code 1:

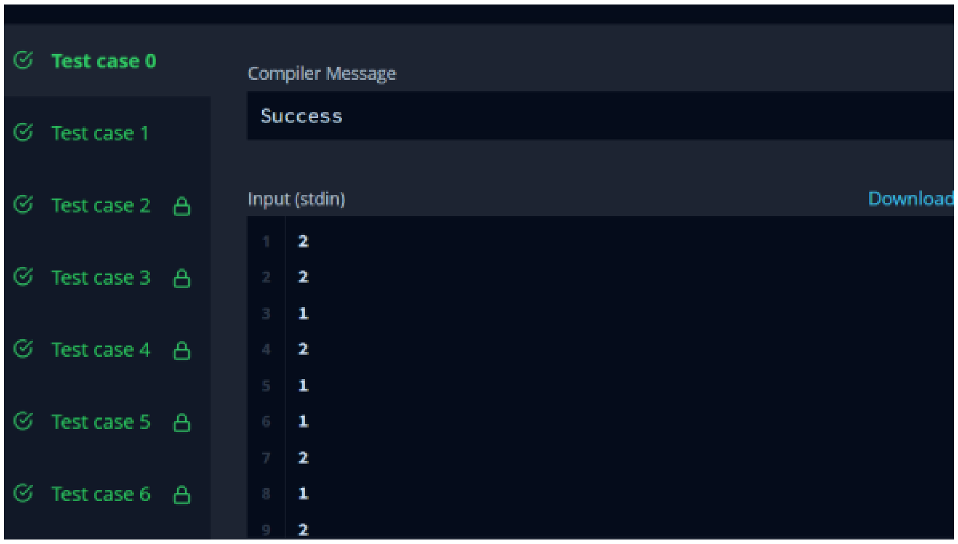
```
static boolean compareLists(SinglyLinkedListNode head1, SinglyLinkedListNode head2) {
    SinglyLinkedListNode temp1 = head1;
    SinglyLinkedListNode temp2 = head2;

    while (temp1 != null && temp2 != null) {
        if (temp1.data != temp2.data) {
            return false;
        }

        temp1 = temp1.next;
        temp2 = temp2.next;
    }

    return temp1 == null && temp2 == null;
}
```

Code 2:

```
public static DoublyLinkedListNode sortedInsert(DoublyLinkedListNode llist, int data) {
    DoublyLinkedListNode newNode = new DoublyLinkedListNode(data);

    if (llist == null) {
        return newNode;
    }

    DoublyLinkedListNode current = llist;
    DoublyLinkedListNode previous = null;

    while (current != null && current.data < data) {
        previous = current;
        current = current.next;
    }

    if (previous == null) {
        newNode.next = llist;
        llist.prev = newNode;
        llist = newNode;
    } else if (current == null) {
        previous.next = newNode;
        newNode.prev = previous;
    } else {
        previous.next = newNode;
        newNode.prev = previous;
        newNode.next = current;
        current.prev = newNode;
    }

    return llist;
}
```

OUTPUT:

OUTPUT 1



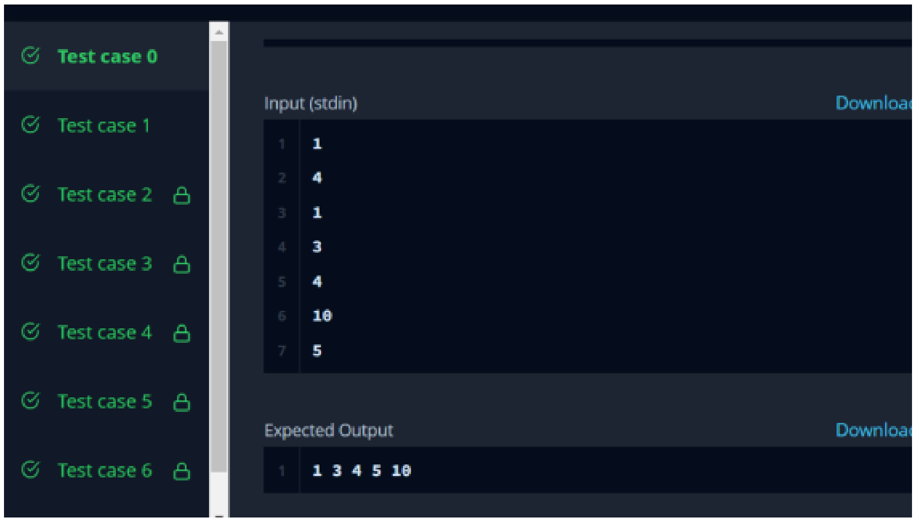
Compiler Message

```
Success
```

Input (stdin)

```
2
2
1
2
1
1
2
1
2
```

OUTPUT 2



Input (stdin)

```
1
4
1
3
4
10
5
```

Expected Output

```
1 3 4 5 10
```

LEARNING OUTCOMES:

1. *Learn about Linkedlist manipulation technique.*
2. *Learn about Linked List conditional logic.*
3. *Learn about algorithm thinking*
4. *Learn about mathematical logic*