## CHANDIGARH UNIVERSITY

## UNIVERSITY INSTITUTE OF NGINEERING

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



|  |  |
| --- | --- |
| **Submitted By: Submitted To:**  Vivek Kumar(21BCS8129) Mamta Punia(E12337) | |
| **Subject Name** | Competitive Coding - I |
| **Subject Code** | 20CSP-314 |
| **Branch** | Computer Science and Engineering |
| **Semester** | 5th |

**Experiment No. - 3**

**Student Name: Vivek Kumar UID: 21BCS8129**

**Branch: BE-CSE(LEET) Section/Group: WM-20BCS-616/A**

**Semester: 5th Date of Performance: 02/09/2022**

**Subject Name: Competitive coding - I Subject Code: 20CSP-314**

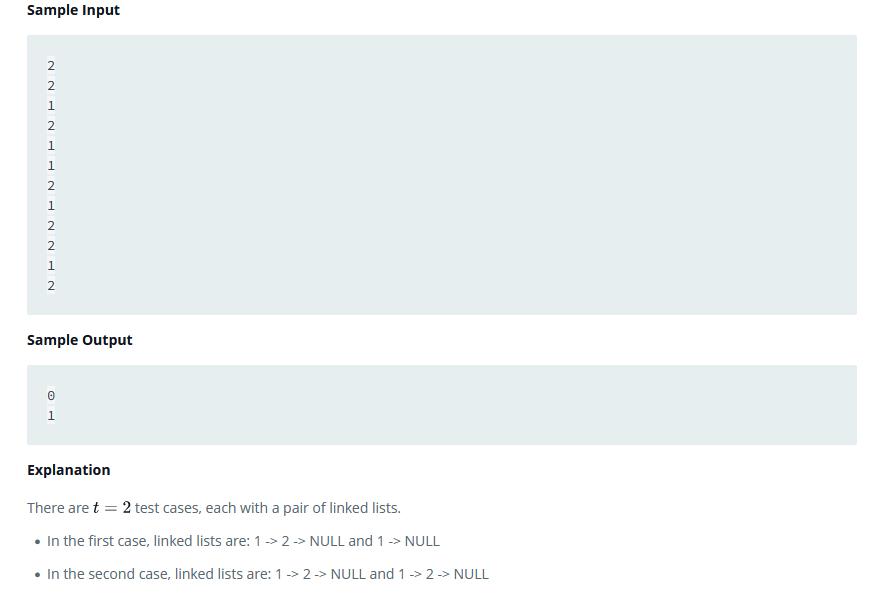
**Compare two Linked list:**

**1. Aim/Overview of the practical:**

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 0. Otherwise, return 1.

**2. Task to be done/ Which logistics used:**





**3. Hardware and Software Requirements (For programming-based labs):**

* Laptop or Desktop
* Hacker-Rank Account

**4. Steps for experiment/practical/Code:**

static boolean compareLists(SinglyLinkedListNode head1, SinglyLinkedListNode head2) {

while ((head1 != null && head2 != null) && head1.data == head2.data) {

head1 = head1.next;

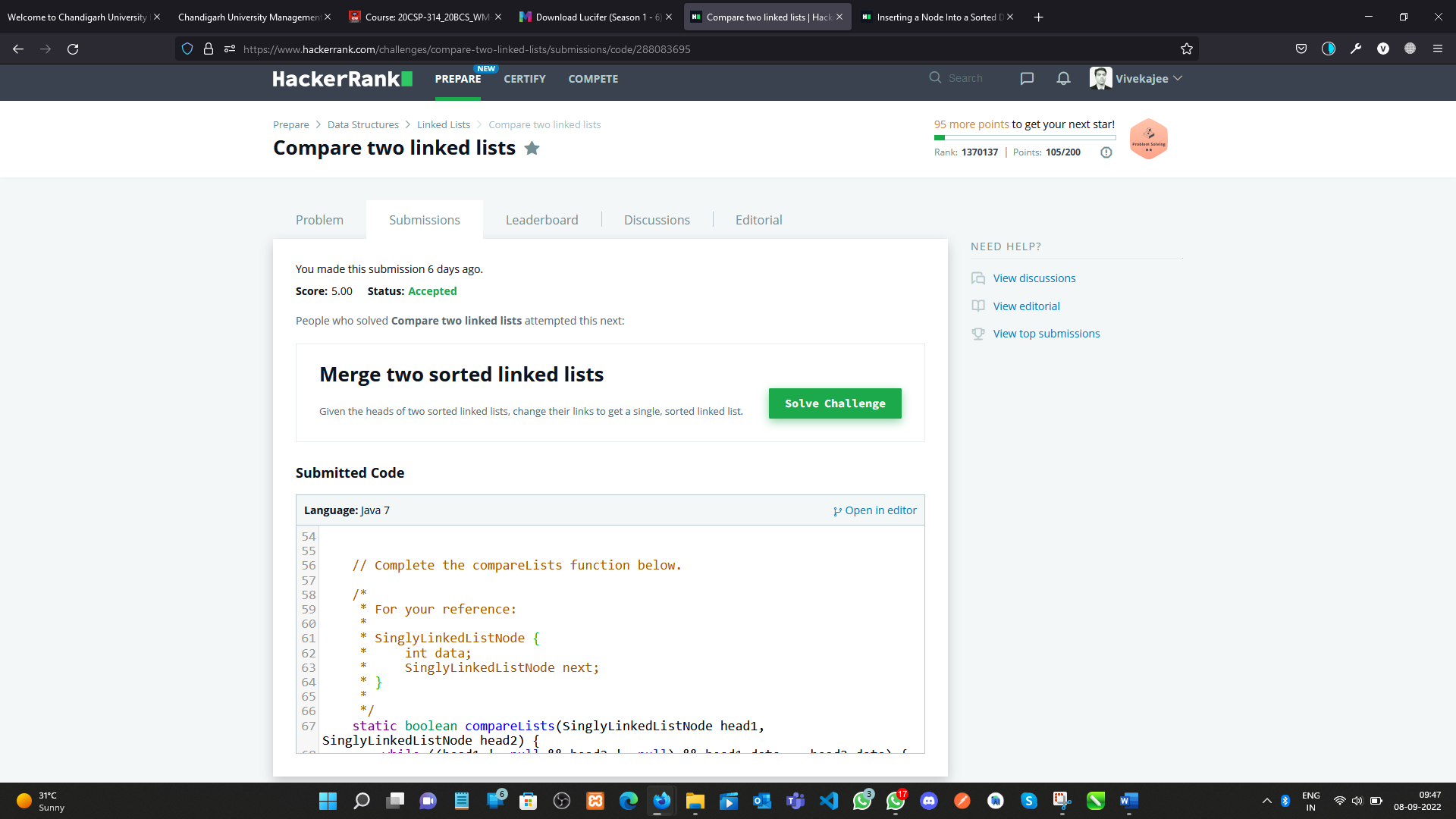
head2 = head2.next;

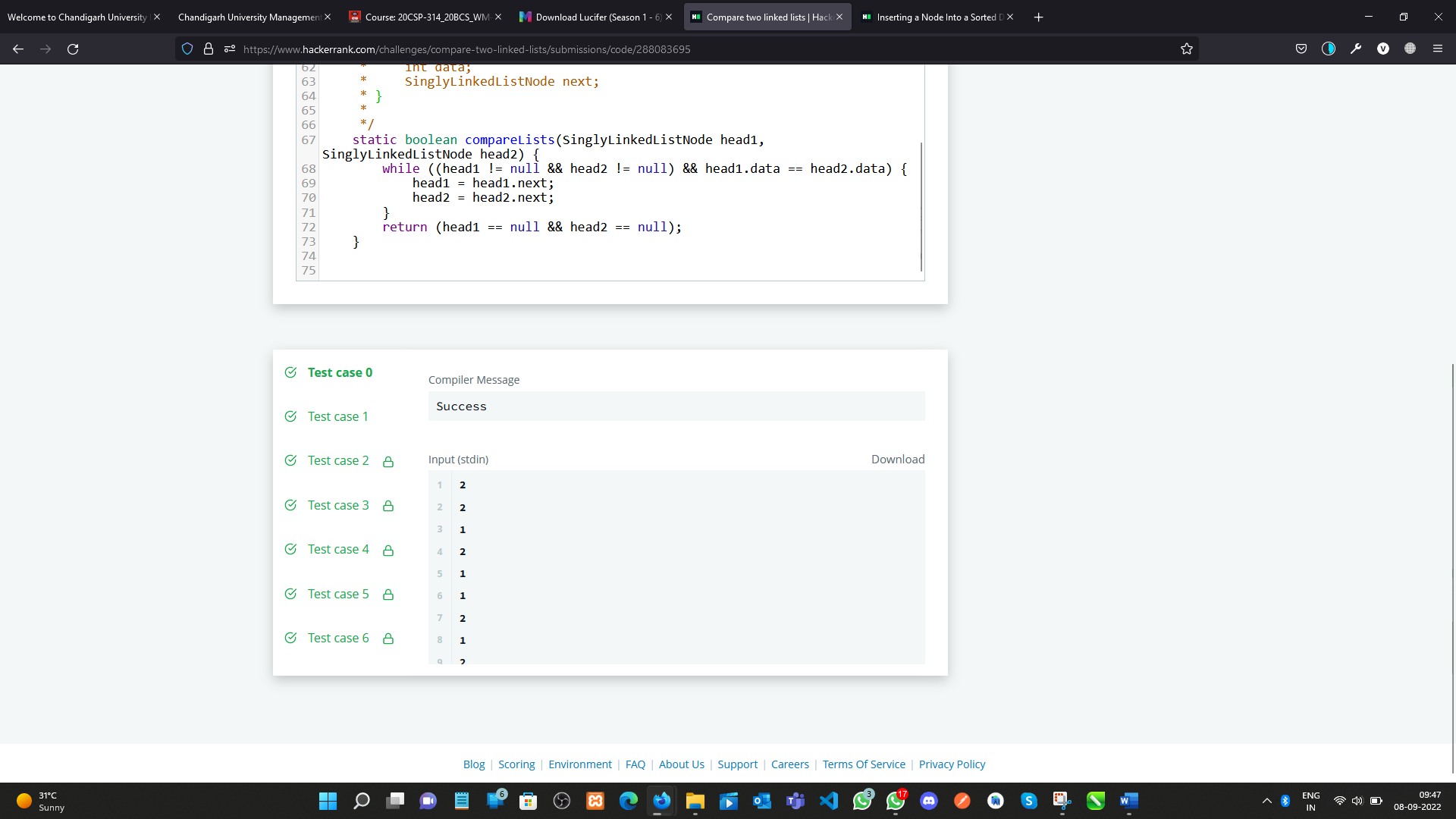
}

return (head1 == null && head2 == null);

}

**5. Result/Output/Writing Summary:**





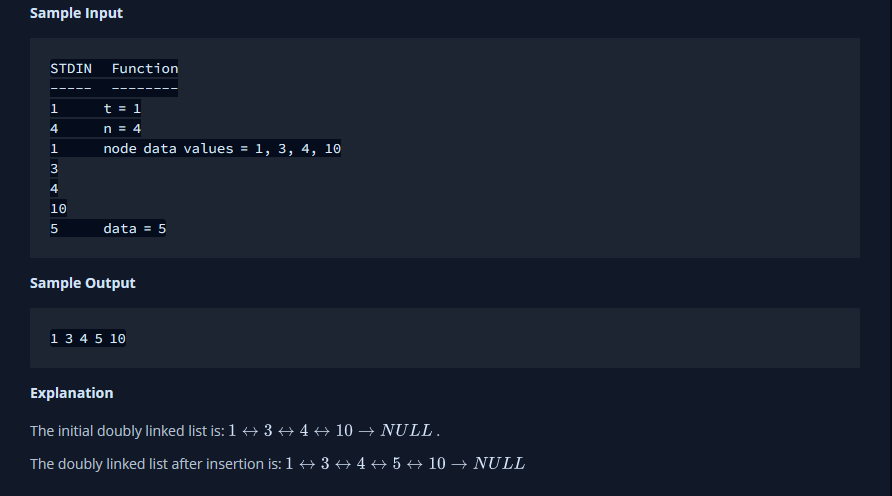
**Inserting a Node Into a Sorted Doubly Linked List:**

**1. Aim/Overview of the practical:**

Given a reference to the head of a doubly-linked list and an integer, data, create a new DoublyLinkedListNode object having data value data and insert it at the proper location to maintain the sort.

**2. Task to be done/ Which logistics used:**





**3. Hardware and Software Requirements (For programming-based labs):**

* Laptop or Desktop
* Hacker-Rank Account

**4. Steps for experiment/practical/Code:**

public static DoublyLinkedListNode sortedInsert(DoublyLinkedListNode llist, int data) {

// Write your code here

DoublyLinkedListNode node = new DoublyLinkedListNode(data);

DoublyLinkedListNode temp = llist;

if(llist.data>=node.data){

node.next = llist;

llist.prev = node;

llist = node;

return llist;

}

while(temp != null ){

if(node.data < temp.data){

DoublyLinkedListNode early = temp.prev;

node.next = temp;

node.prev = temp.prev;

temp.prev = node;

early.next = node;

return llist;

}

if(temp.next == null){

temp.next = node;

node.prev = temp;

return llist;

}

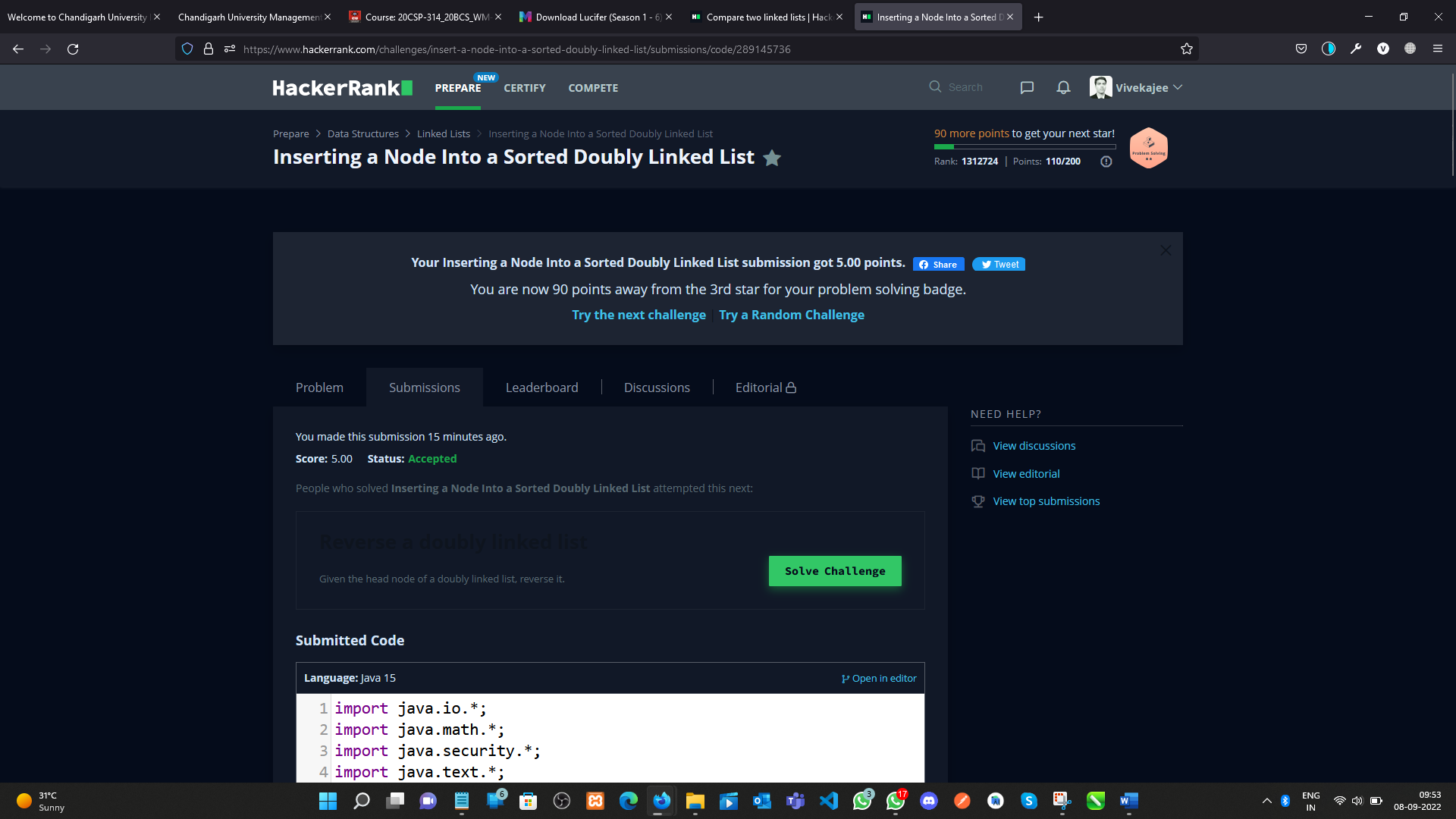
temp = temp.next;

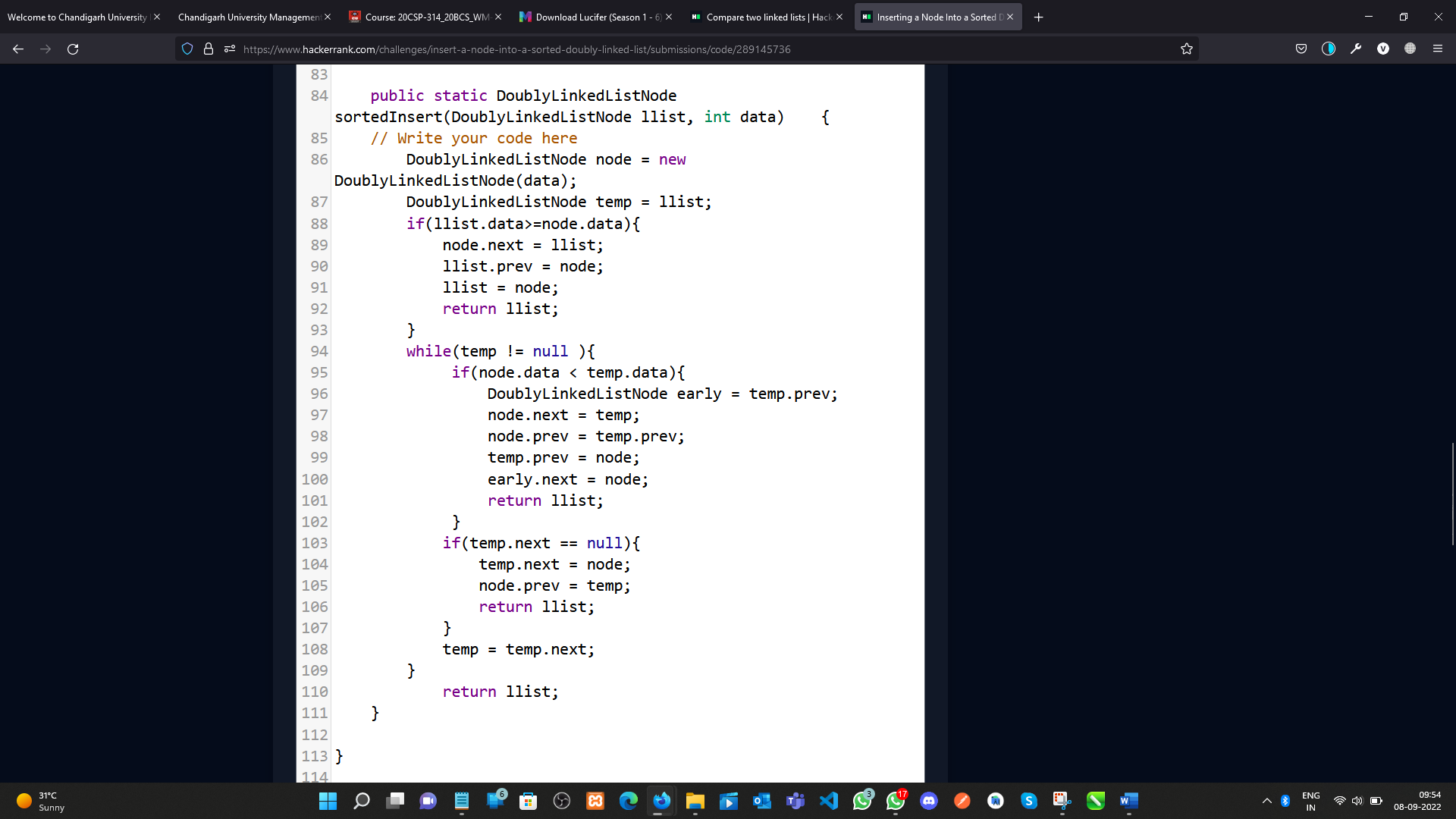
}

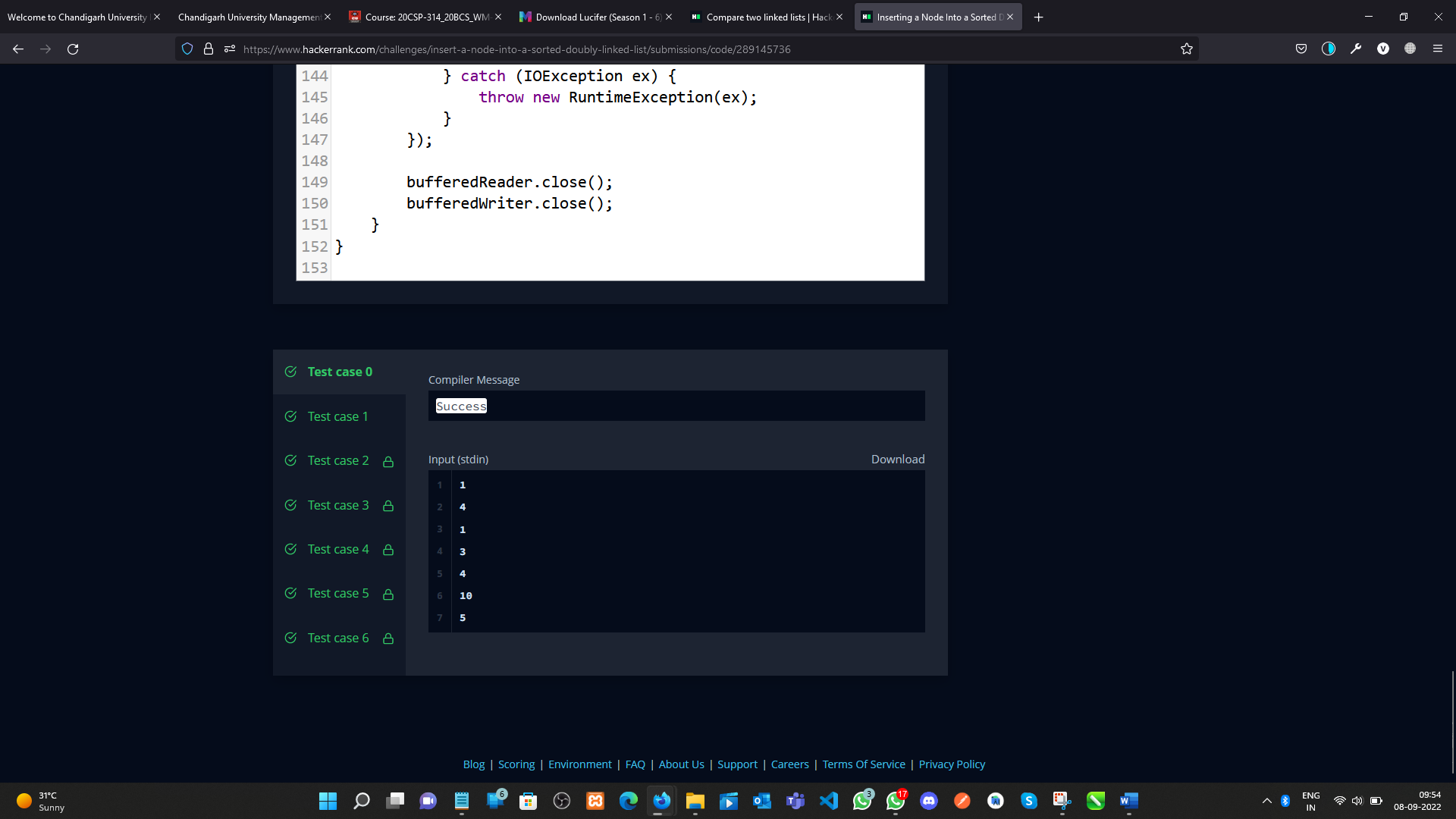
return llist;

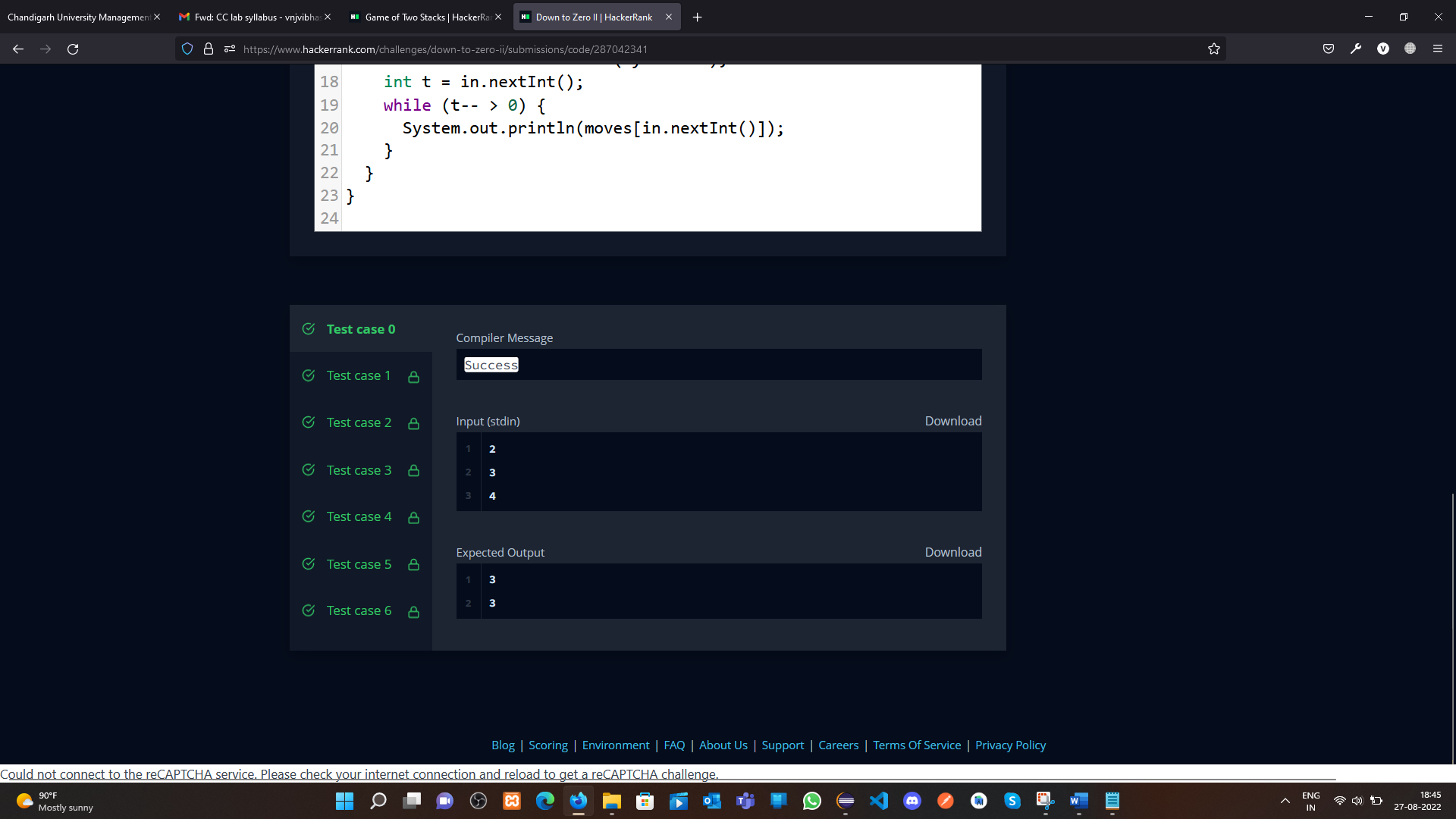
}

**6. Result/Output/Writing Summary:**









**Learning outcomes (What I have learnt):**

1. Concept of LinkedList & Doubly LinkedList**.**
2. Completed my two questions.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |