**DAILY REPORT**

**Student Name :SUSHMITHA B POOJARY**

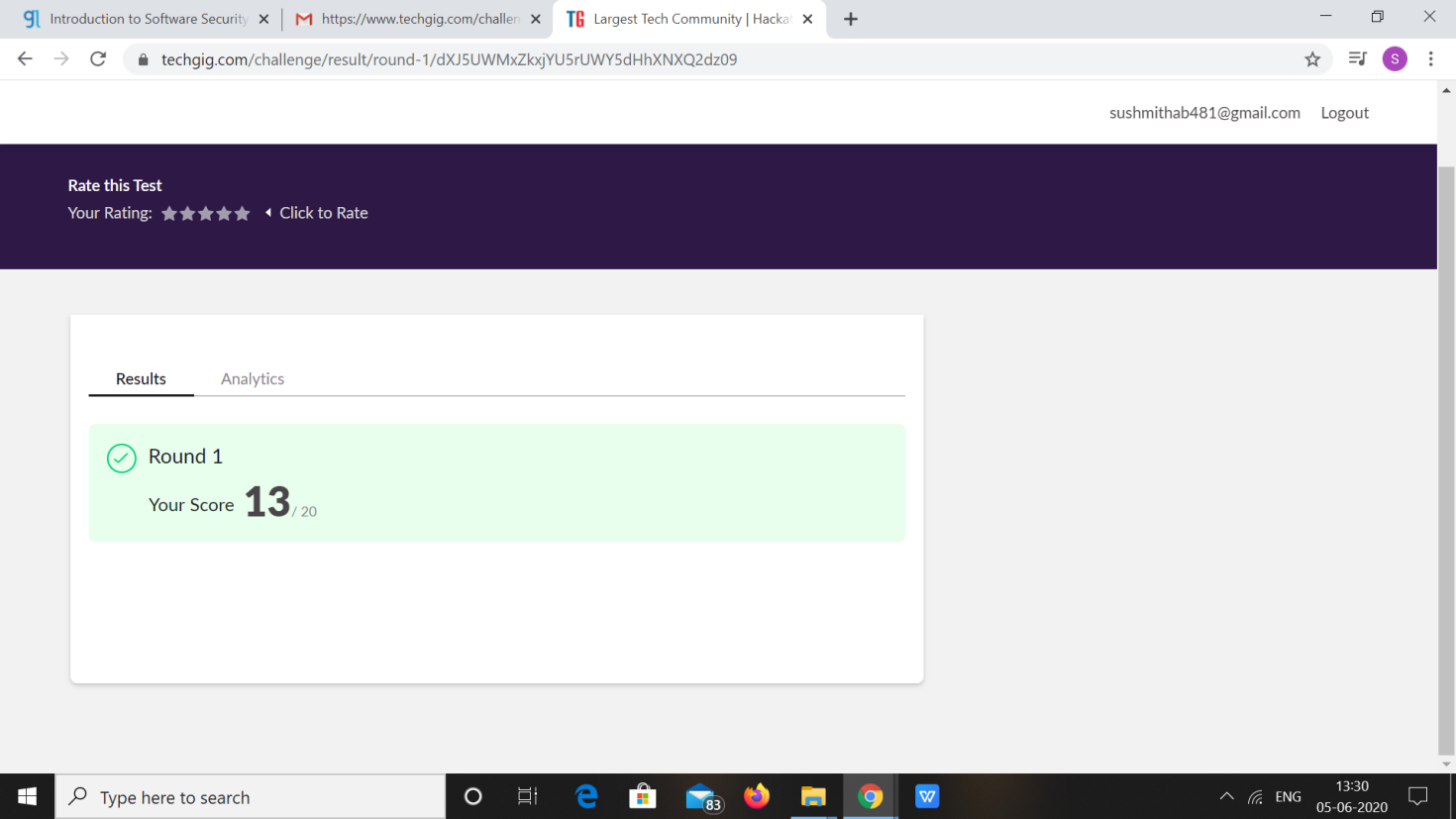
**Class and Sec : VI B**

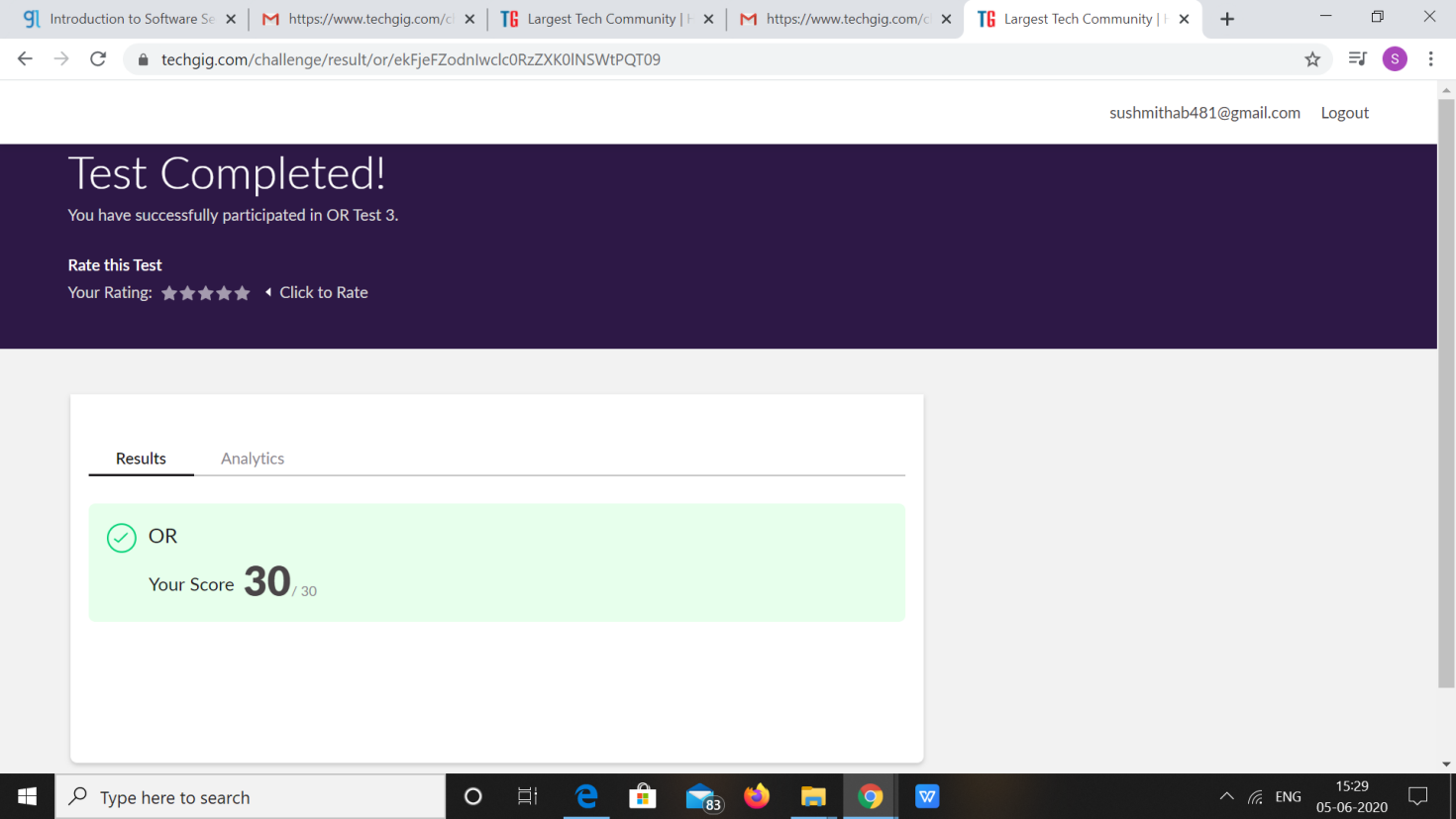
**USN :4AL17CS103**

**DATE:05-06-2020**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Online Test Details** | | | | |
| **Subject** | **Python Application Programming|OR** | | | |
| **Semester** | **VI -B** | | **Duration** | **30 Minutes-PAP**  **75Minutes OR** |
| **% of marks PAP-20**  **OR-30** | | **PAP-13**  **OR-30** | | |

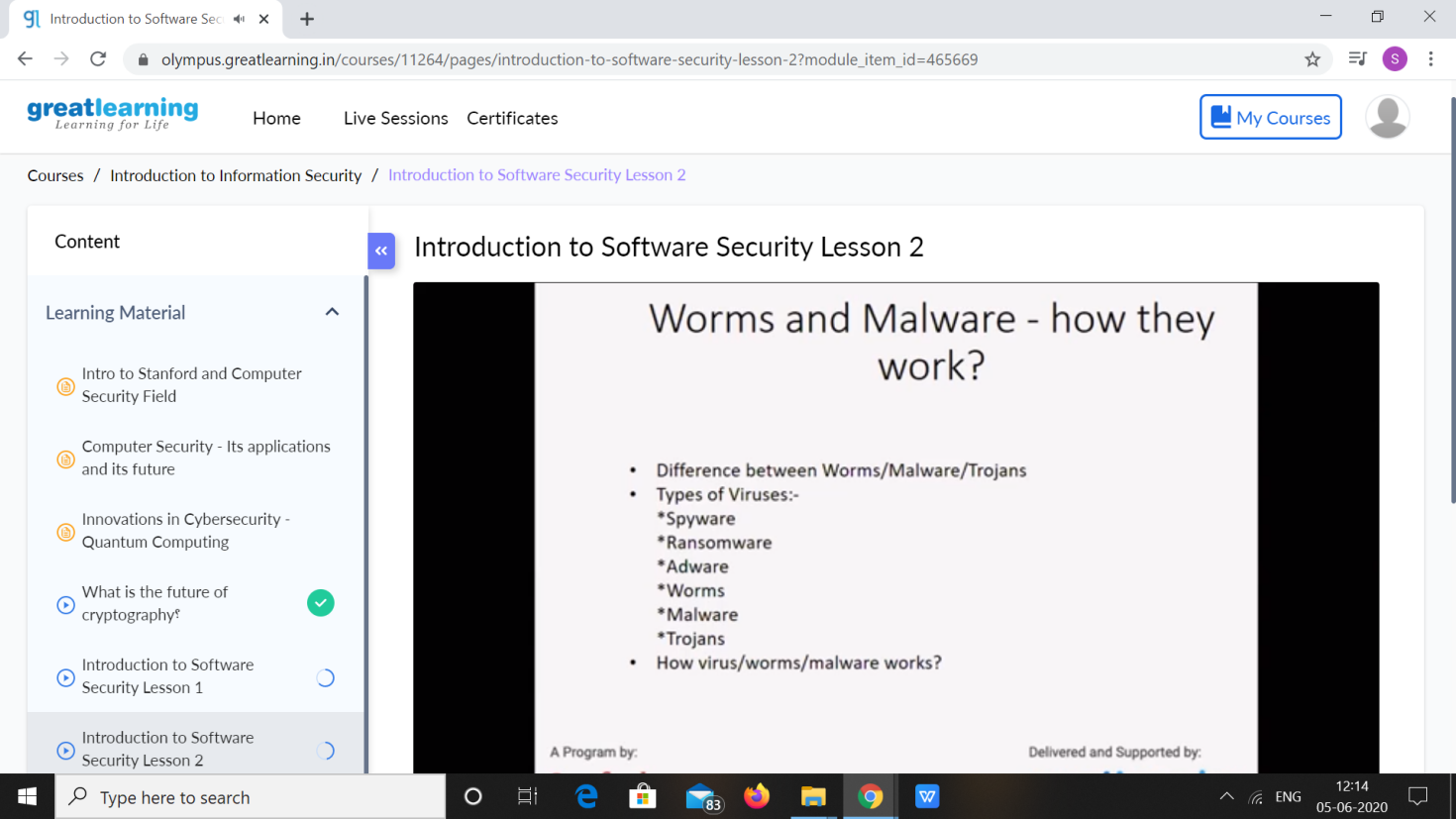
**Snapshot of the test result**

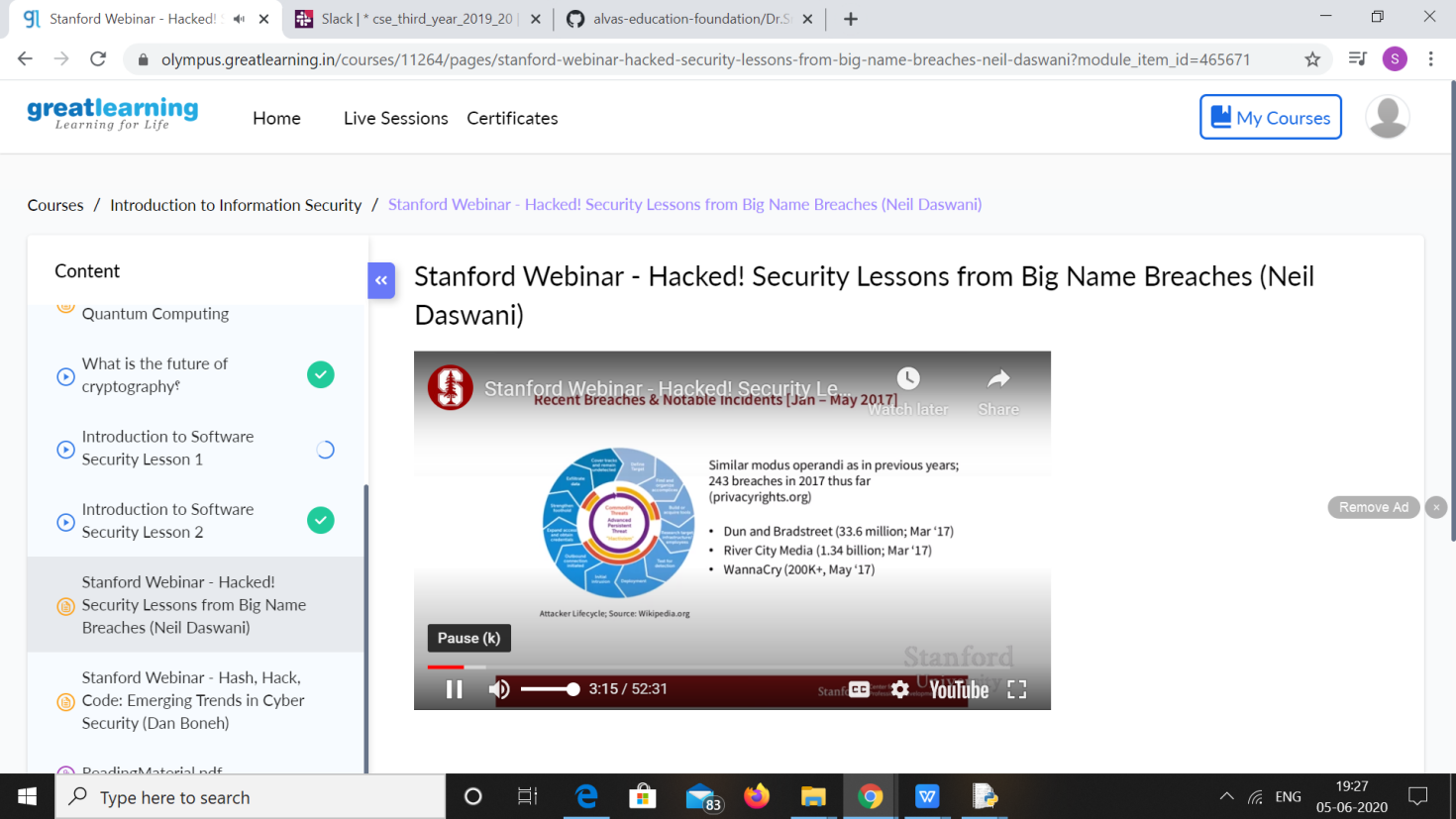
****

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Certification Course Details** | | | |
| **Course** | **Information Security** | | |
| **Certificate Provider** | **Great Learning** | **Duration** | **5.5hours** |

**Snapshots of the daily class acitivities**

****



|  |  |
| --- | --- |
| **Coding Challenges** | |
| **Problem Statement: 1.** Write a Java program to implement Circular Linked List Using Array And Class.   1. Python program to square each odd number in the list   **3.The program takes a string and removes the nth index character from the non-empty string.** | |
| **Status: Executed** | |
| **Uploaded the report both in Github & Slack** | **Yes** |

**Snapshots of your response to challenge.**

**Coding Challenges Details:**

1**.Write a Java program to implement Circular Linked List Using Array And Class**

**package** prog14;

**public class** CircularLinkedList {

**public int** size =0;

**public** Node head=**null**;

**public** Node tail=**null**;

//add a new node at the start of the linked list

**public void** addNodeAtStart(**int** data){

System.***out***.println("Adding node " + data + " at start");

Node n = **new** Node(data);

**if**(size==0){

head = n;

tail = n;

n.next = head;

}**else**{Node temp = head;

n.next = temp;

head = n;

tail.next = head;

}

size++;

}

**public void** addNodeAtEnd(**int** data){

**if**(size==0){

addNodeAtStart(data);

}**else**{

Node n = **new** Node(data);

tail.next =n;

tail=n;

tail.next = head;

size++;

}

System.***out***.println("\nNode " + data + " is added at the end of the list");

}

**public void** deleteNodeFromStart(){

**if**(size==0){

System.***out***.println("\nList is Empty");

}**else**{

System.***out***.println("\ndeleting node " + head.data + " from start");

head = head.next;

tail.next=head;

size--;

}

}

**public int** elementAt(**int** index){

**if**(index>size){

**return** -1;

}

Node n = head;

**while**(index-1!=0){

n=n.next;

index--;

}

**return** n.data;

}

//print the linked list

**public void** print(){

System.***out***.print("Circular Linked List:");

Node temp = head;

**if**(size<=0){

System.***out***.print("List is empty");

}**else**{

**do** {

System.***out***.print(" " + temp.data);

temp = temp.next;

}**while**(temp!=head);

}

System.***out***.println();

}

//get Size

**public int** getSize(){

**return** size;

}

**public static void** main(String[] args) {

CircularLinkedList c = **new** CircularLinkedList();

c.addNodeAtStart(3);

c.addNodeAtStart(2);

c.addNodeAtStart(1);

c.print();

c.deleteNodeFromStart();

c.print();

c.addNodeAtEnd(4);

c.print();

System.***out***.println("Size of linked list: "+ c.getSize());

System.***out***.println("Element at 2nd position: "+ c.elementAt(2));

}

}

**package** prog14;

**class** Node{

**int** data;

Node next;

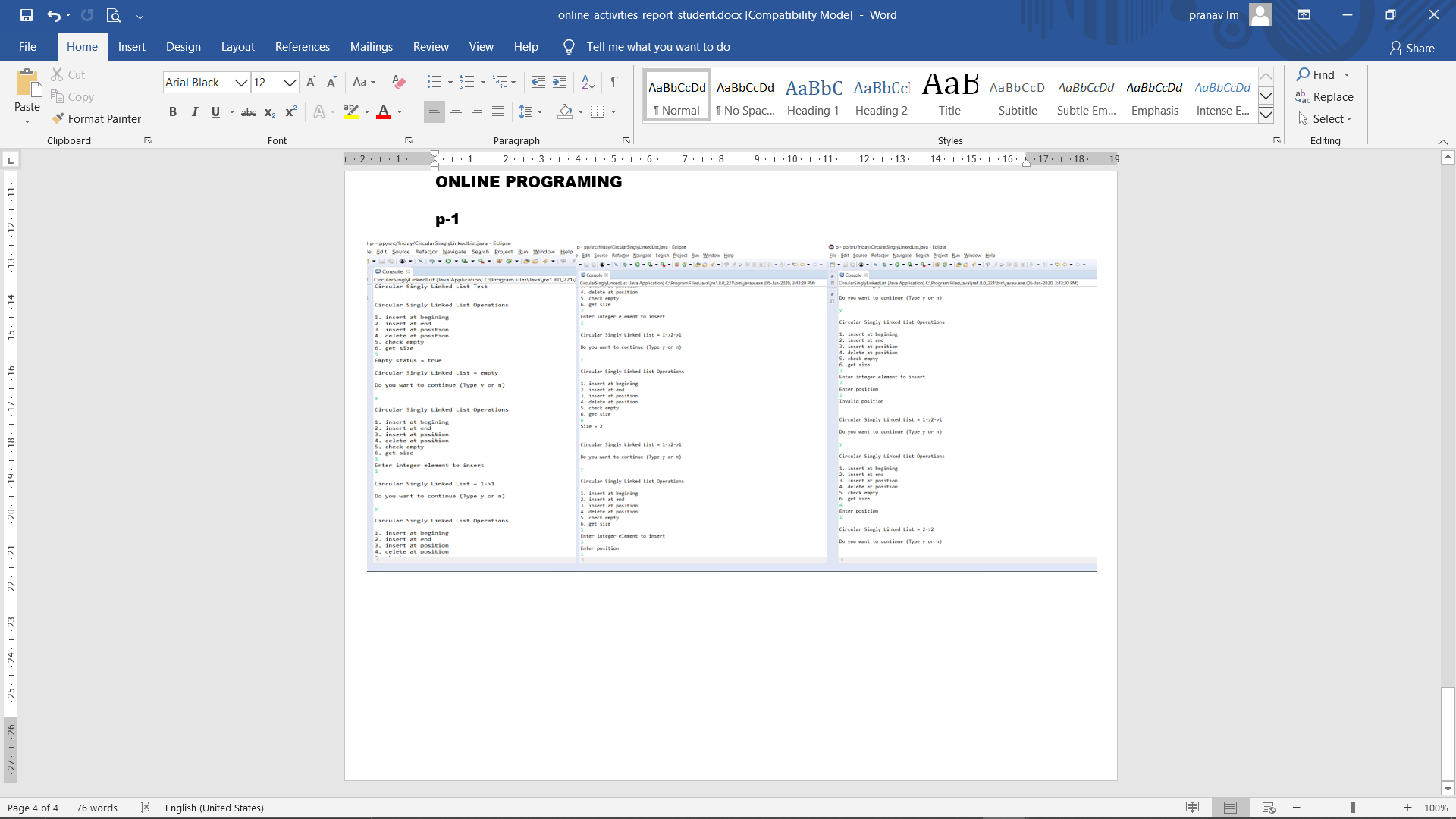
**public** Node(**int** data){

**this**.data = data;

}

}

**Output:**



**2.Python program to square each odd number in the list**

Description

Take a list of numbers and square each odd number in the list. Print output as comma

separated sequence.

eg:

input list: [2,4,5,6,7,8,9]

output: 25,49,81

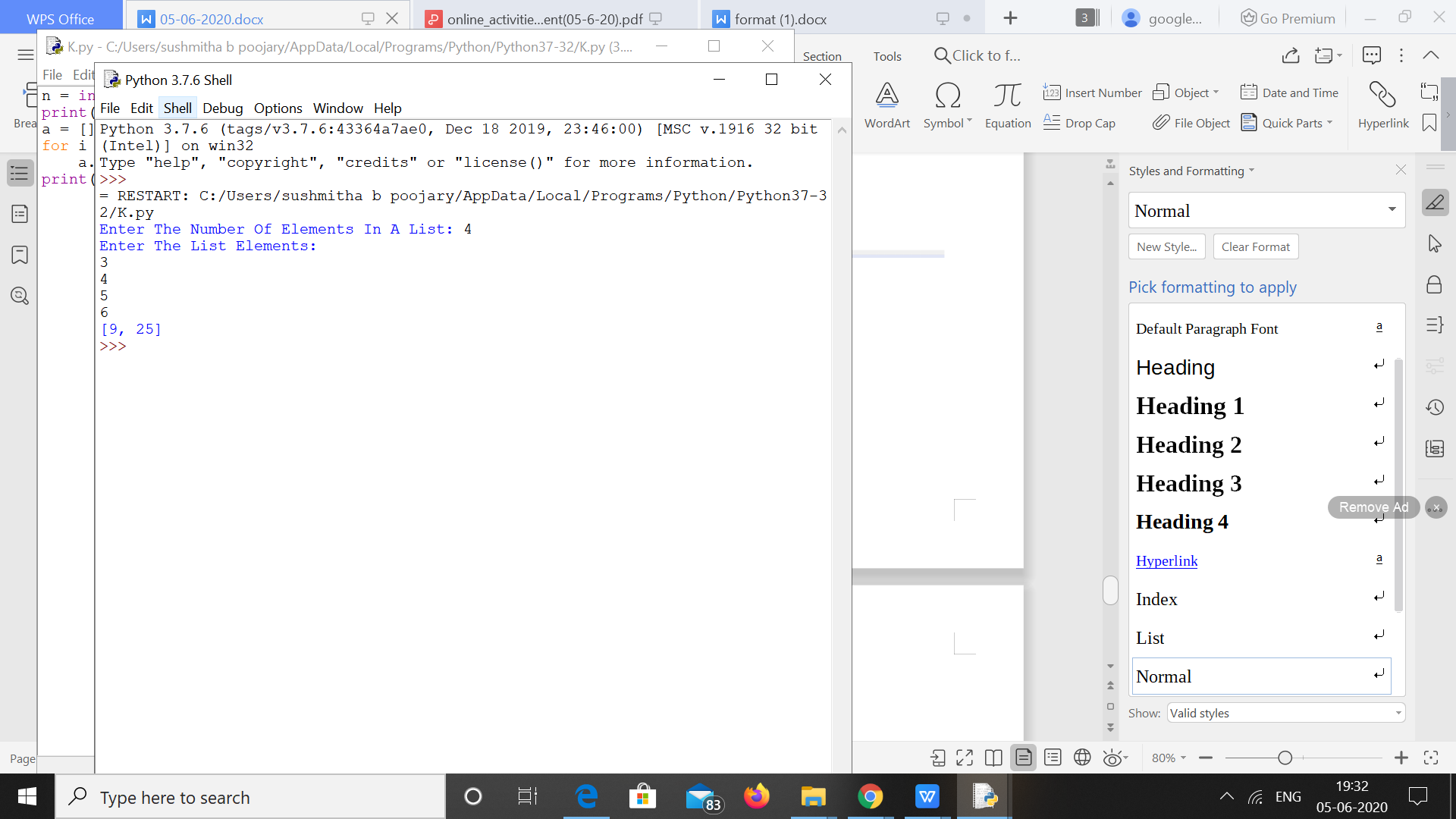
**Program:**

a=[2,4,5,6,7,8,9]

print(a)

print([i\*i for i in a if(i%2!=0)])

**OUTPUT**

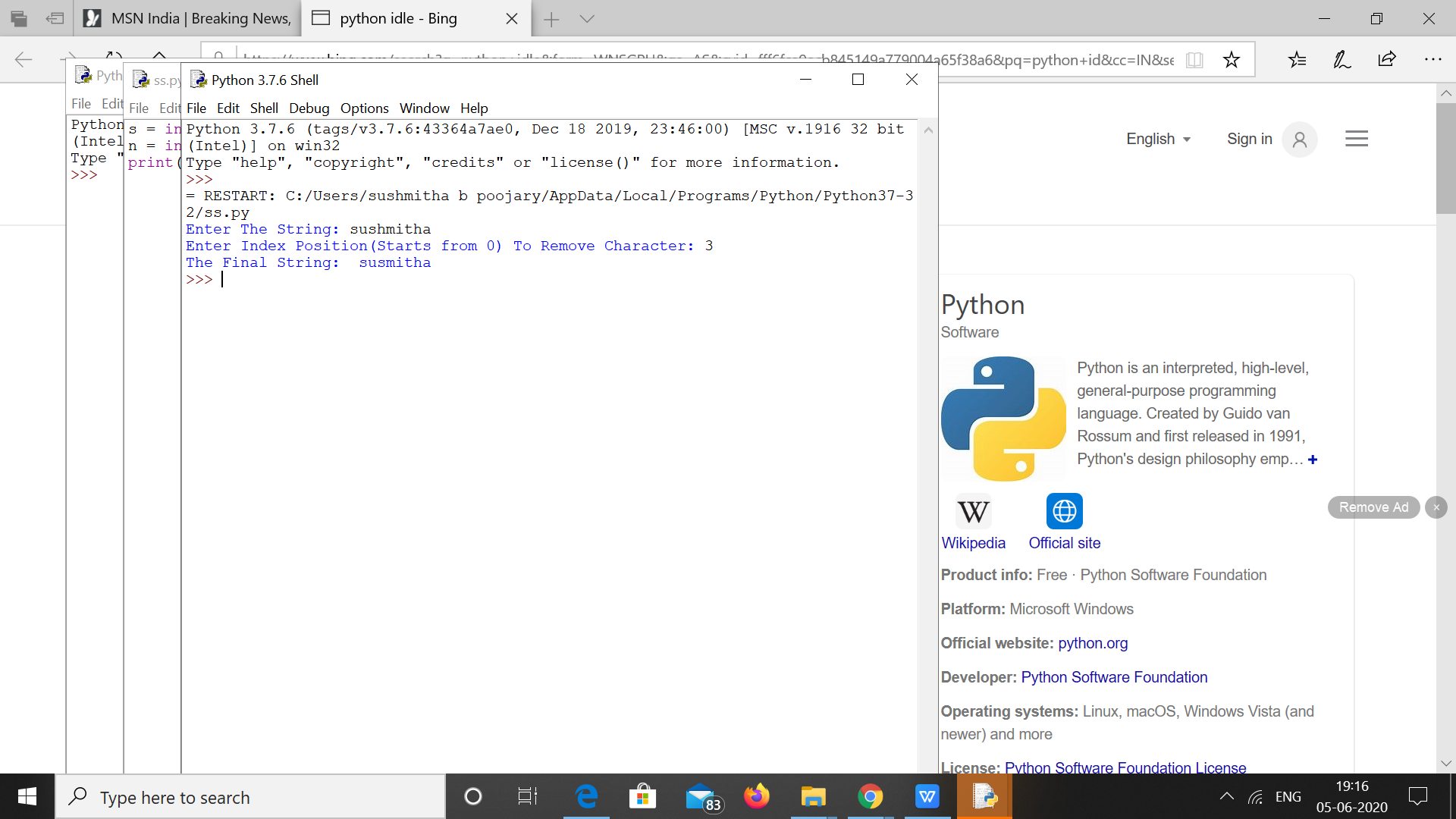
****

**3.The program takes a string and removes the nth index character from the non-empty string.**

s = input("Enter The String: ")

n = int(input("Enter Index Position(Starts from 0) To Remove Character: "))

print("The Final String: ", s[:n] + s[n+1:])

****