DAILY ONLINE ACTIVITIES SUMMARY

Date:	**10-06-2020**	**Name:**	**ASHIKA**
Sem & Sec	**6 A**	**USN:**	**4AL17CS016**

Online Test Summary

Subject	**SSCD**		
Max. Marks	**30**	**Score**	**26**

Certification Course Summary

Course	**Python for data science**		
Certificate Provider	**Cognitive class**	**Duration**	**5 hour**

Coding Challenges

Problem Statement:

1.Write a C Program to print the sum of boundary elements of a matrix

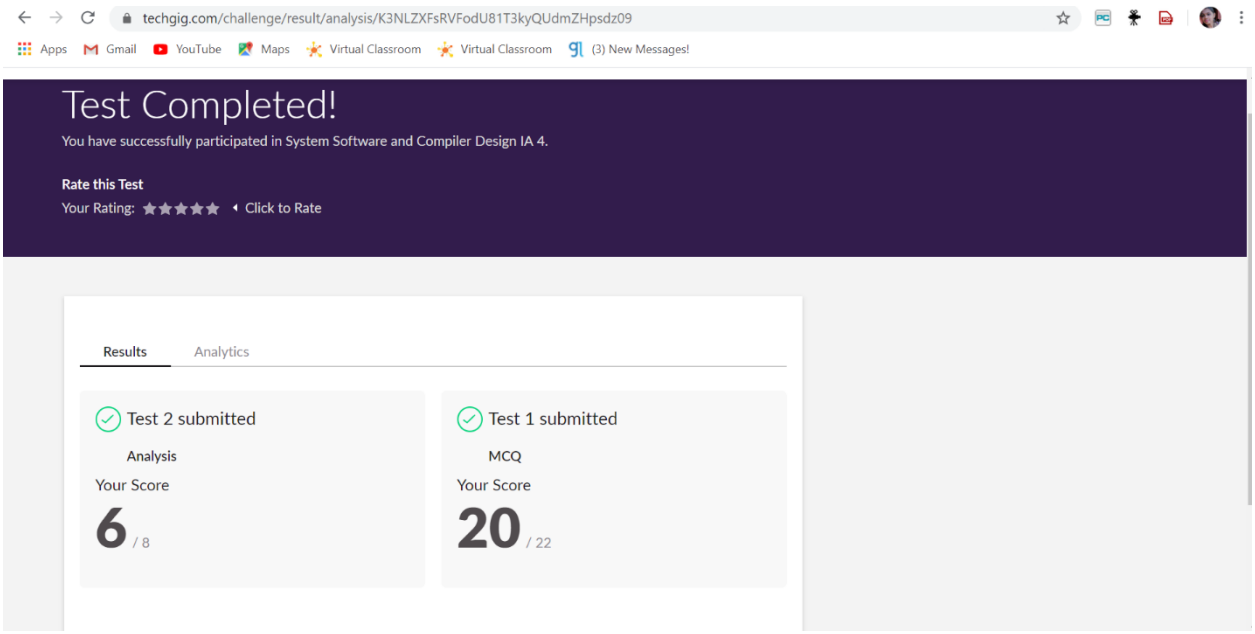
2. Write a Java program to find the maximum and minimum value node from a circular linked list

3. Python Program to check whether a given number is a fibonacci number or not

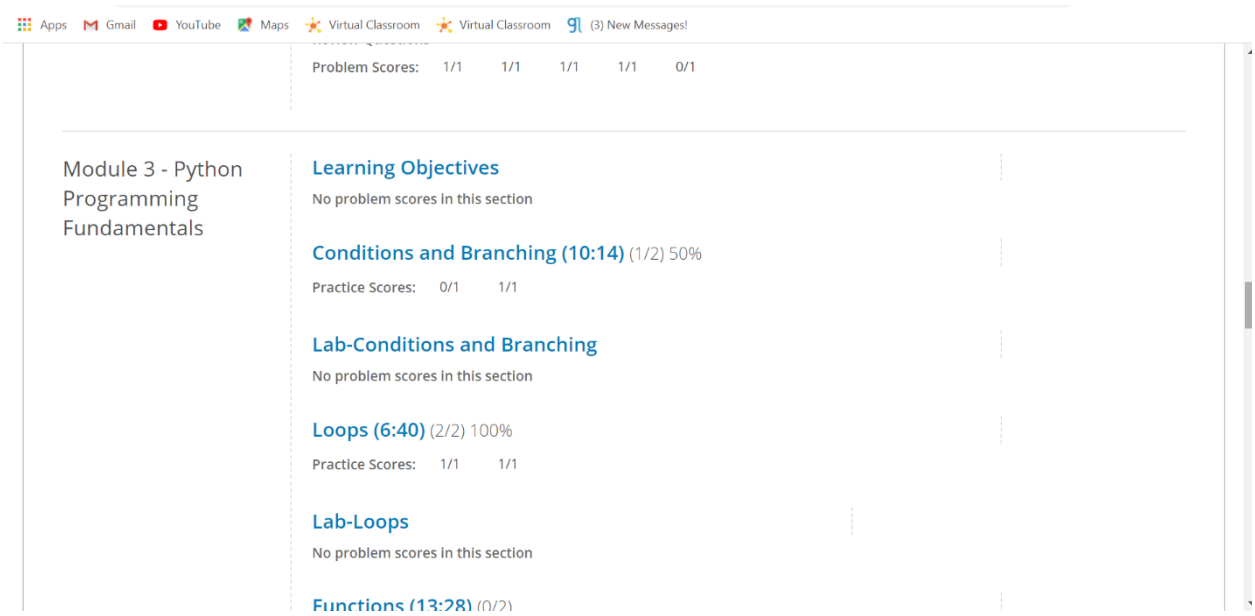
Status: done(executed)	
Uploaded the report in Github	**yes**
If yes Repository name	https://github.com/ASHIKA-05/DAILY-REPORT

Uploaded the report in slack	**yes**

SUBJECT : SSCD



CERTIFICATION COURSE



ONLINE CODING

1.Write a C Program to print the sum of boundary elements of a matrix

Given a matrix, the task is to print the boundary elements of the matrix and display their sum.
Sample Output 1:
Enter M (Rows) and N (Columns): 3, 3
Enter the Elements: 1 2 3 4 5 6 7 8 9
OUTPUT:
The Input Matrix is:
1 2 3
4 5 6
7 8 9
The Boundary Elements are: 1 2 3 4 6 7 8 9
The Sum of Boundary elements of the Matrix is: 40

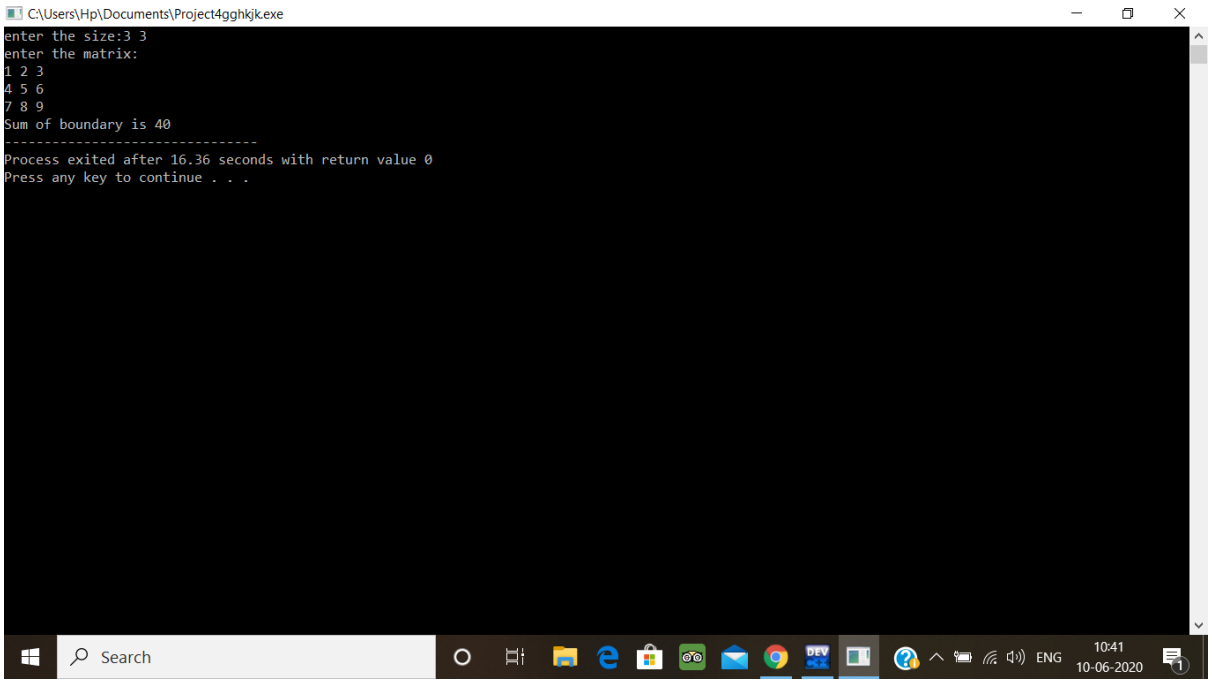
Sample Output 2:
Enter M (Rows) and N (Columns): 4, 5
Enter the Elements: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
OUTPUT:
The Input Matrix is:
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

17 18 19 20
The Boundary Elements are: 1 2 3 4 5 8 9 12 13 16 17 18 19 20
The Sum of Boundary elements of the Matrix is: 147

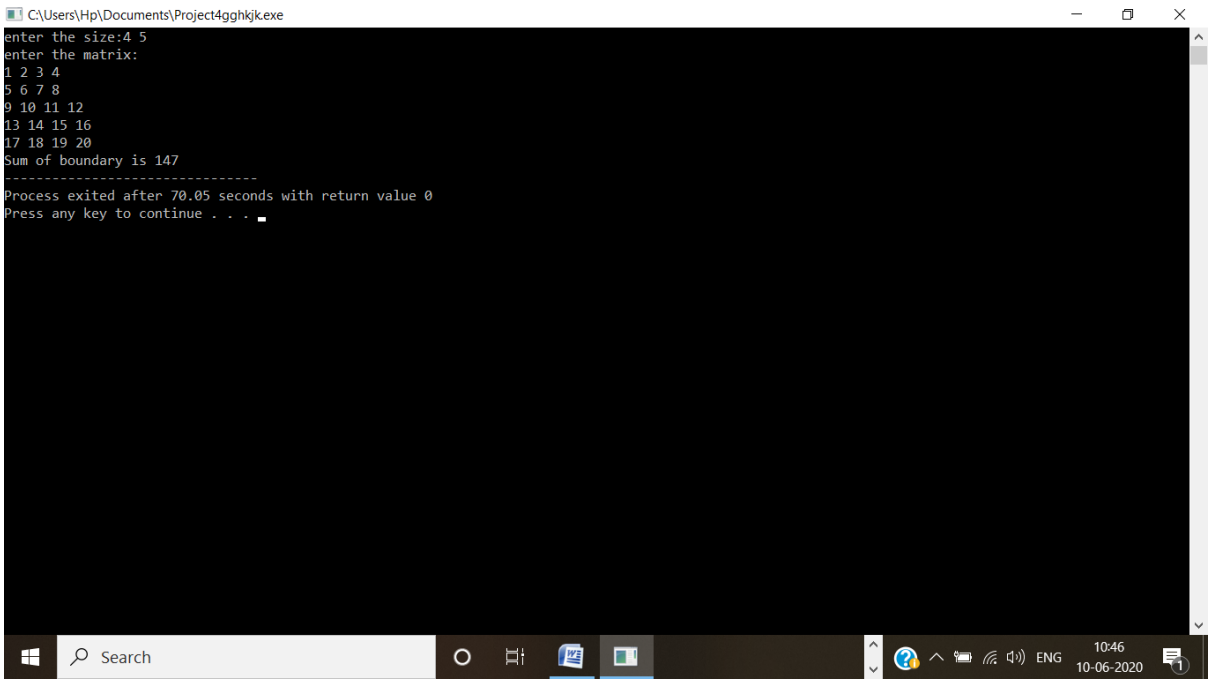
```
#include<stdio.h>

#include<stdlib.h>

int main()

{

   int **a,r,c,i,j;

   printf("enter the size:");

   scanf("%d",&r);

   scanf("%d",&c);



   a=(int**)malloc(r*sizeof(int*));

   for(i=0;i<r;i++)

      *(a+i)=(int*)malloc(c*sizeof(int));

      printf("enter the matrix:\n");

   for(i=0;i<r;i++)

   {

      for(j=0;j<c;j++)



      {



         scanf("%d",*(a+i)+j);

      }

   }
```

```
    i=0;int sum1=0;
    for(j=0;j<c;j++)
        sum1=sum1+*(*(a+i)+j);

    i=r-1;int sum2=0;
    if(i!=0)
    {
        for(j=0;j<c;j++)
            sum2=sum2+*(*(a+i)+j);
    }

    j=0; int sum3=0;
    for(i=1;i<r-1;i++)
        sum3=sum3+*(*(a+i)+j);

    j=c-1; int sum4=0;
    for(i=1;i<r-1;i++)
        sum4=sum4+*(*(a+i)+j);

    printf("Sum of boundary is %d",sum1+sum2+sum3+sum4);
    return 0;
}
```

Output:


C:\Users\Hp\Documents\Project4gghkjk.exe
enter the size:3 3
enter the matrix:
1 2 3
4 5 6
7 8 9
Sum of boundary is 40

Process exited after 16.36 seconds with return value 0
Press any key to continue . . .
Search
ENG
10:41
10-06-2020


C:\Users\Hp\Documents\Project4gghkjk.exe
enter the size:4 5
enter the matrix:
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20
Sum of boundary is 147

Process exited after 70.05 seconds with return value 0
Press any key to continue . . .
Search
ENG
10:46
10-06-2020

2. Write a Java program to find the maximum and minimum value node from a circular linked list

Algorithm
Define a Node class which represents a node in the list. It has two properties data and next which will point to the next node.
Define another class for creating the circular linked list and it has two nodes: head and tail.
minNode() will print out minimum value node:
Define variable min and initialize with head's data.
Current will point to head.
Iterate through the list by comparing each node's data with min.
If min > current's data then min will hold current's data.
At the end of the list, variable min will hold the minimum value node.
Print the min value.

maxNode() will prints out maximum value node:
Define variable max and initialize with head's data.
Current will point to head.
Iterate through the list by comparing each node's data with max.
If max > current's data then max will hold current's data.
At the end of the list, variable max will hold the maximum value node.
Print the max value.

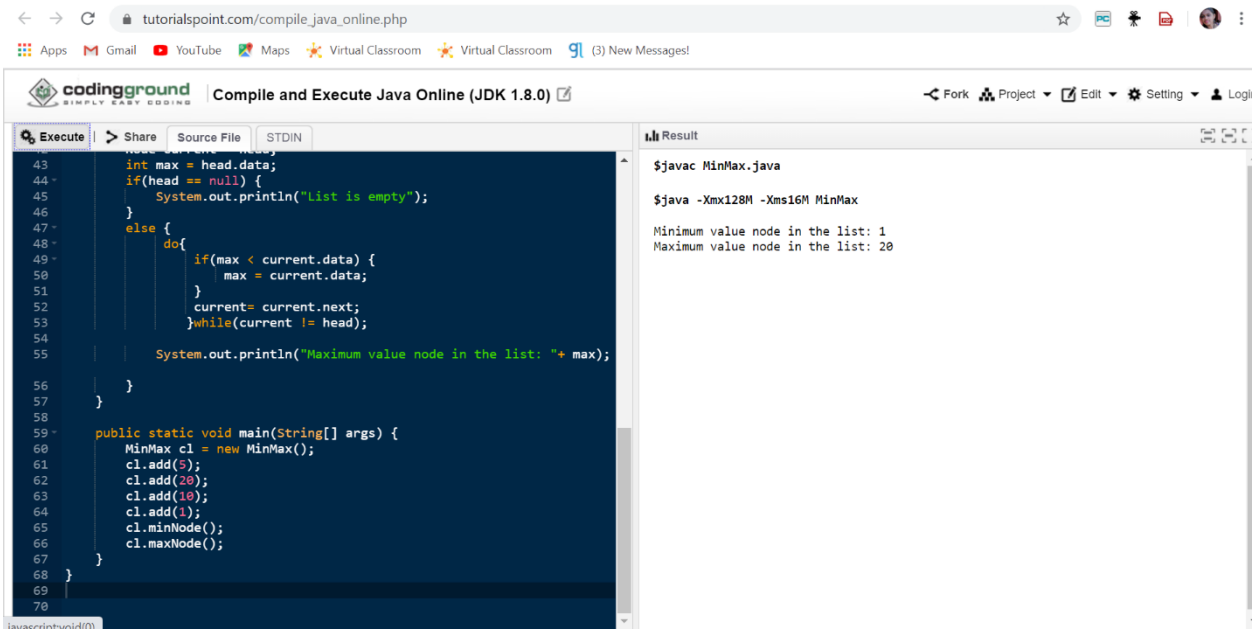
```
public class MinMax {

   public class Node{

      int data;

      Node next;

      public Node(int data) {

         this.data = data;

      }

   }

   public Node head = null;

   public Node tail = null;

   public void add(int data){

      Node newNode = new Node(data);
```

```
    if(head == null) {
        head = newNode;
        tail = newNode;
        newNode.next = head;
    }
    else {
        tail.next = newNode;
        tail = newNode;
        tail.next = head;
    }
}
  public void minNode() {
    Node current = head;
    int min = head.data;
    if(head == null) {
        System.out.println("List is empty");
    }
    else {
         do{
            if(min > current.data) {
                min = current.data;
            }
            current= current.next;
        }while(current != head);
```

```
        System.out.println("Minimum value node in the list: "+ min);
    }
}
public void maxNode() {
    Node current = head;
    int max = head.data;
    if(head == null) {
        System.out.println("List is empty");
    }
    else {
         do{
            if(max < current.data) {
                max = current.data;
            }
            current= current.next;
           }while(current != head);

        System.out.println("Maximum value node in the list: "+ max);
    }
}

public static void main(String[] args) {
    MinMax cl = new MinMax();
    cl.add(5);
    cl.add(20);
```

```
        cl.add(10);
        cl.add(1);
        cl.minNode();
        cl.maxNode();
    }
}
```

Output:



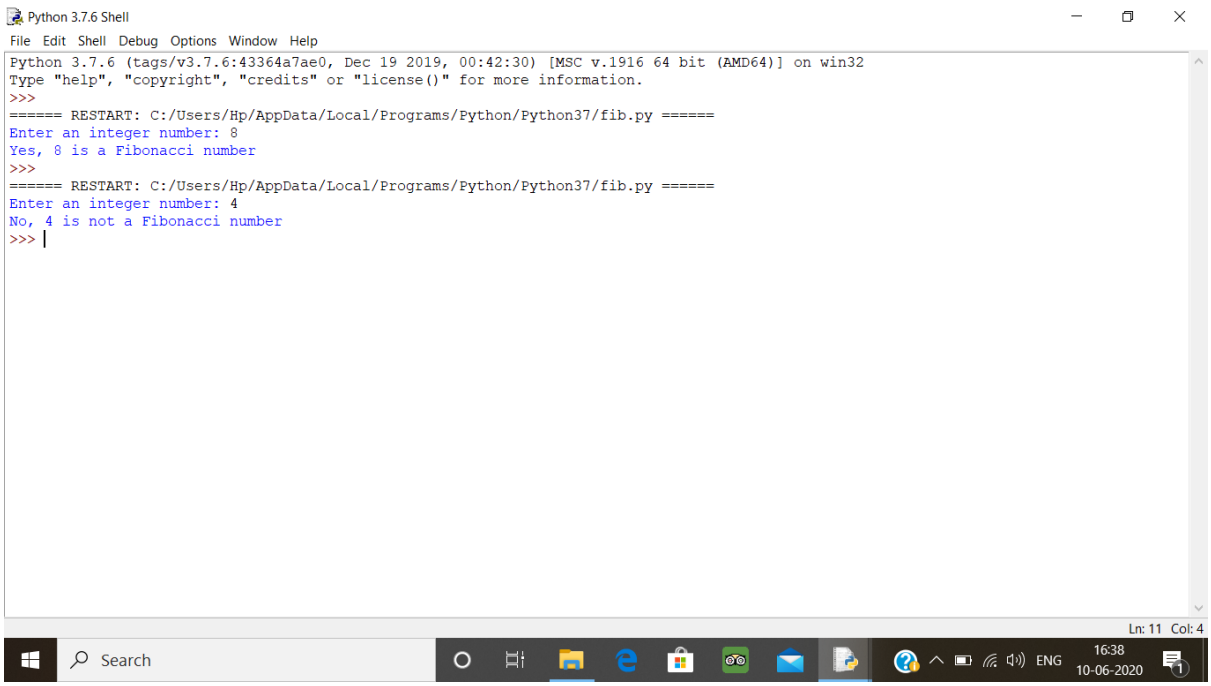
3. Python Program to check whether a given number is a fibonacci number or not

Description:
Input number : 8
Output: 8 is a fibonacci number
Input number : 4
Output: 4 is not a fibonacci number

```
import math

def checkPerfectSquare(n):

    sqrt = int(math.sqrt(n))
```

```
    if pow(sqrt, 2) == n:
        return True
    else:
        return False
def isFibonacciNumber(n):
    res1 = 5 * n * n + 4
    res2 = 5 * n * n - 4
    if checkPerfectSquare(res1) or checkPerfectSquare(res2):
        return True
    else:
        return False
num = int(input("Enter an integer number: "))
if isFibonacciNumber(num):
    print ("Yes,", num, "is a Fibonacci number")
else:
    print ("No,", num, "is not a Fibonacci number")
```

output:


Python 3.7.6 Shell
File Edit Shell Debug Options Window Help
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
====== RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python37/fib.py ======
Enter an integer number: 8
Yes, 8 is a Fibonacci number
>>>
====== RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python37/fib.py ======
Enter an integer number: 4
No, 4 is not a Fibonacci number
>>>
Ln: 11 Col: 4
Search
ENG
16:38
10-06-2020