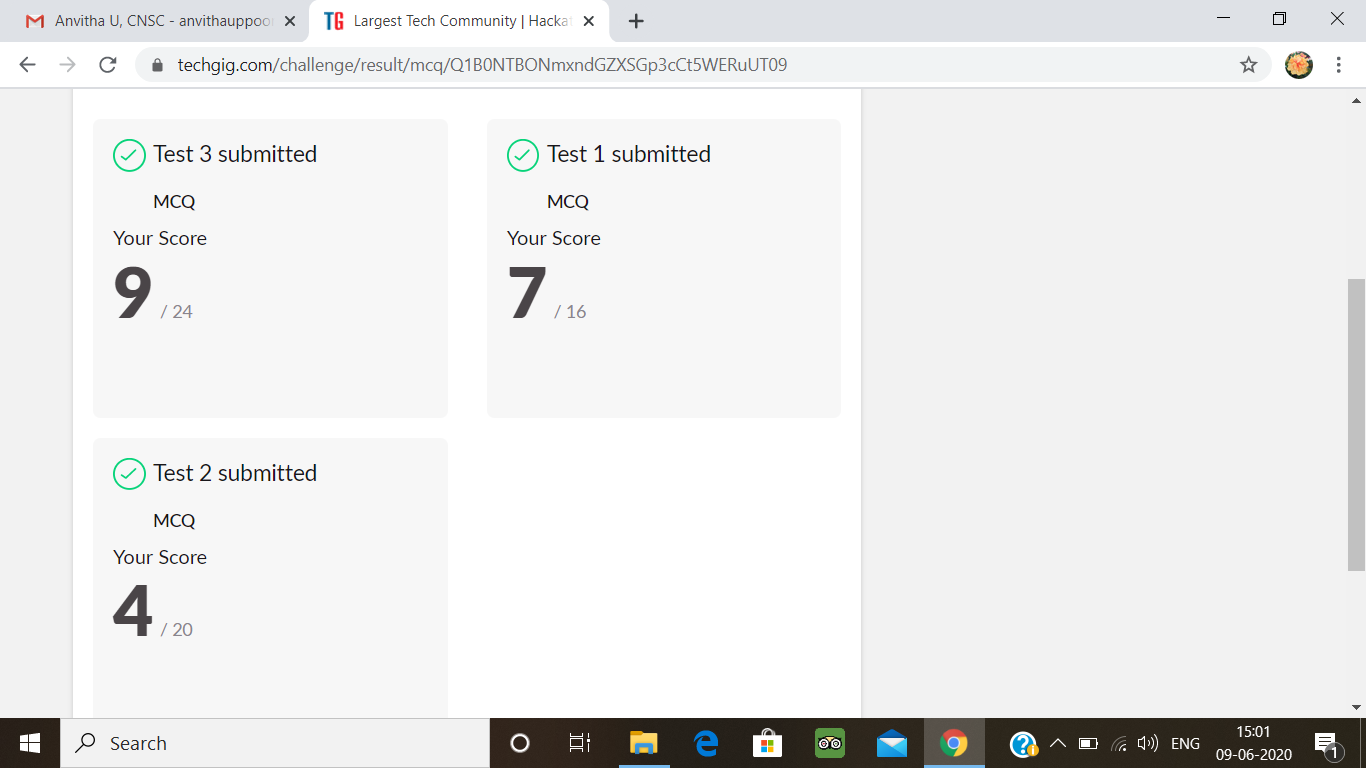
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **10-06-2020** | | | | | **Name:** | **Anvitha U** | |
| **Sem & Sec** | **6th&A** | | | | | **USN:** | **4AL17CS009** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **System Software.** | | | | | | |
| **Max. Marks** | | **30** | | **Score** | | | **25** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **INTRODUCTION TO FULL STACK DEVELOPMENT.** | | | | | | | |
| **Certificate Provider** | | | Greatlearning  Academy | | **Duration** | | | 16hours |
| **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. | | | | | | | | |
| **Status: Done** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **YES** | | | |
| **If yes Repository name** | | | | | <https://github.com/anvithauppoor/online_coding_activity> | | | |
| **Uploaded the report in slack** | | | | | **YES** | | | |

Online Test Details:

Subject:- **System software.**

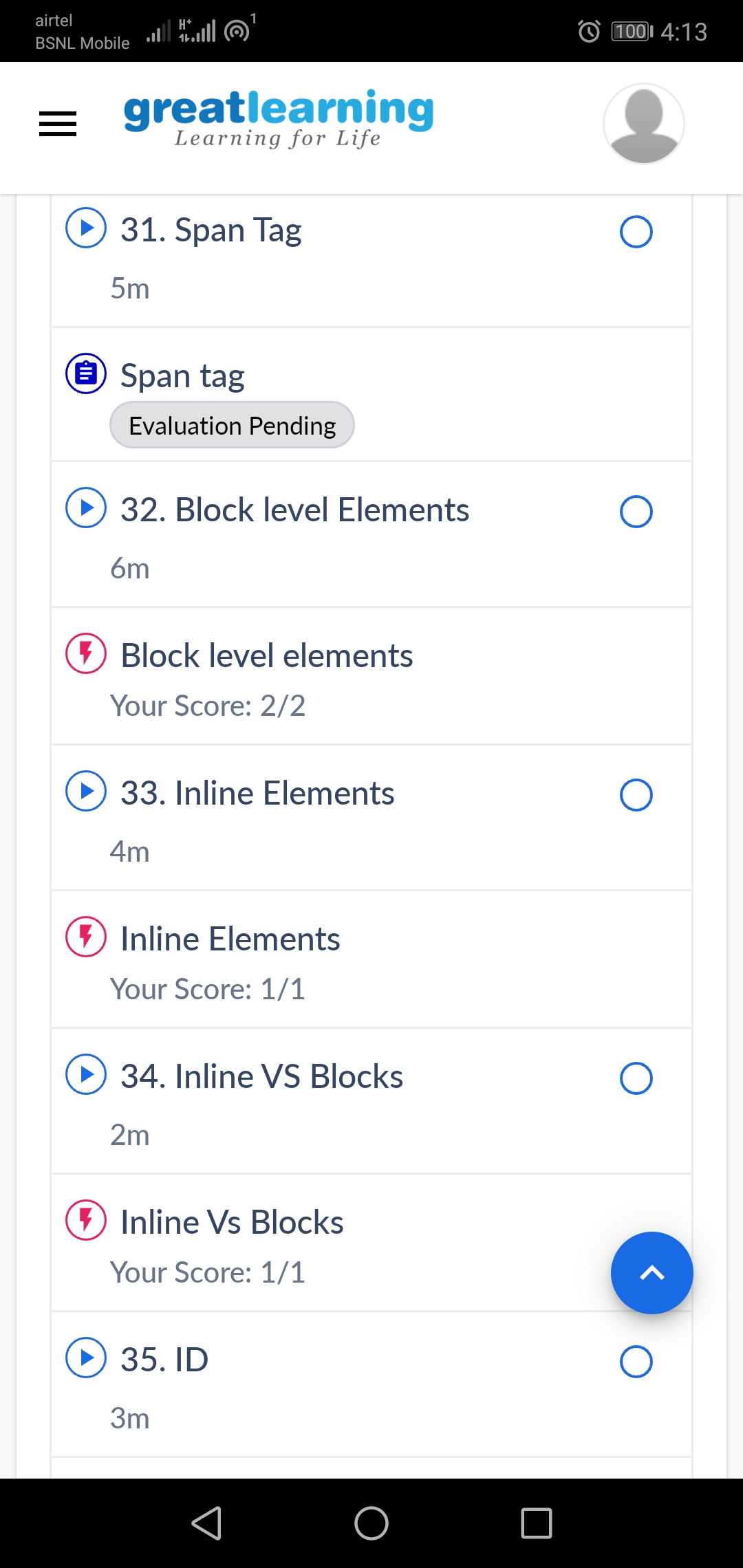


Certification Course Details:

**INTRODUCTION TO FULL STACK DEVELOPMENT.**

Today I have studied :

* Span tag.
* Span tag assignment.
* Block level elements.
* Block level elements assignment.
* inline Elements.
* inline Elements assignment.
* Inline vs Blocks
* Inline vs Blocks assignment.
* ID
* ID assignment.



Coding Challenges Details:

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++)

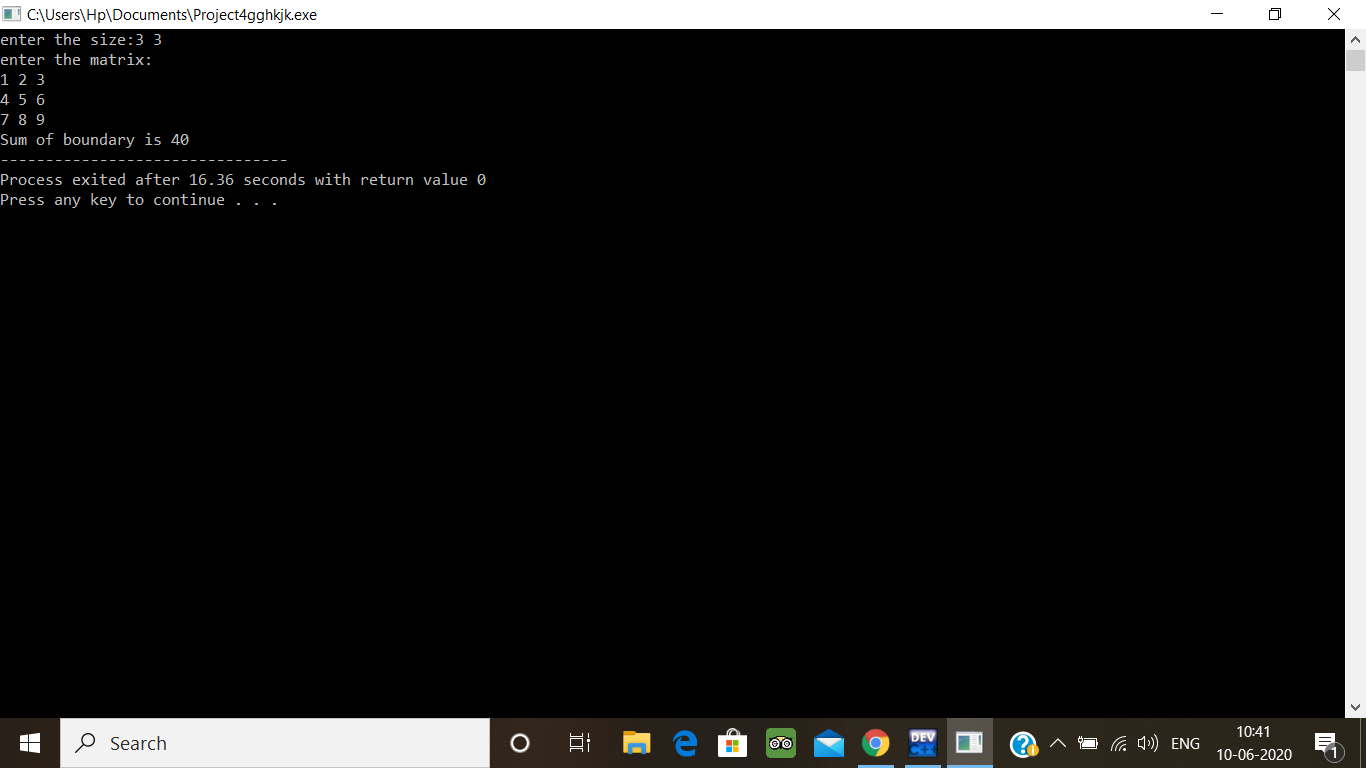
sum4=sum4+\*(\*(a+i)+j);

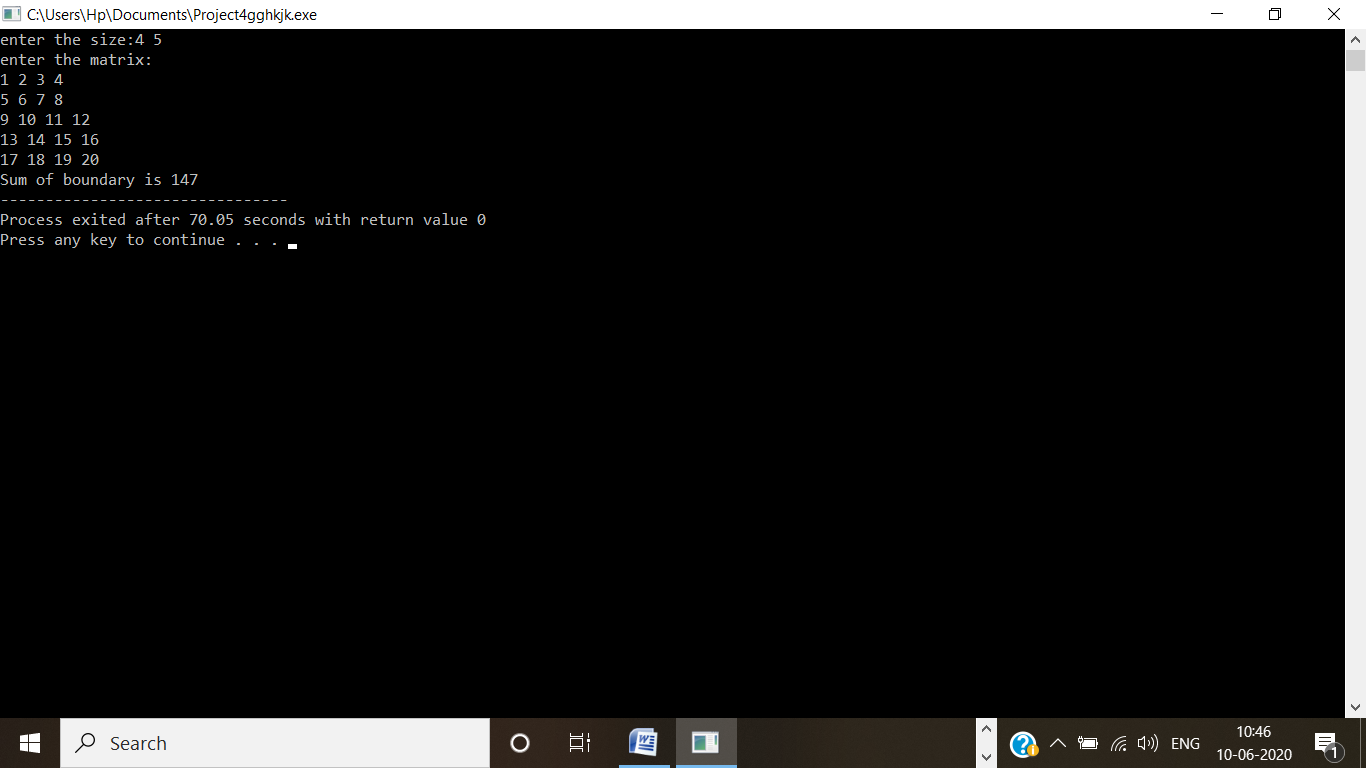
printf("Sum of boundary is %d",sum1+sum2+sum3+sum4);

return 0;

}

**Output:**





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:**

