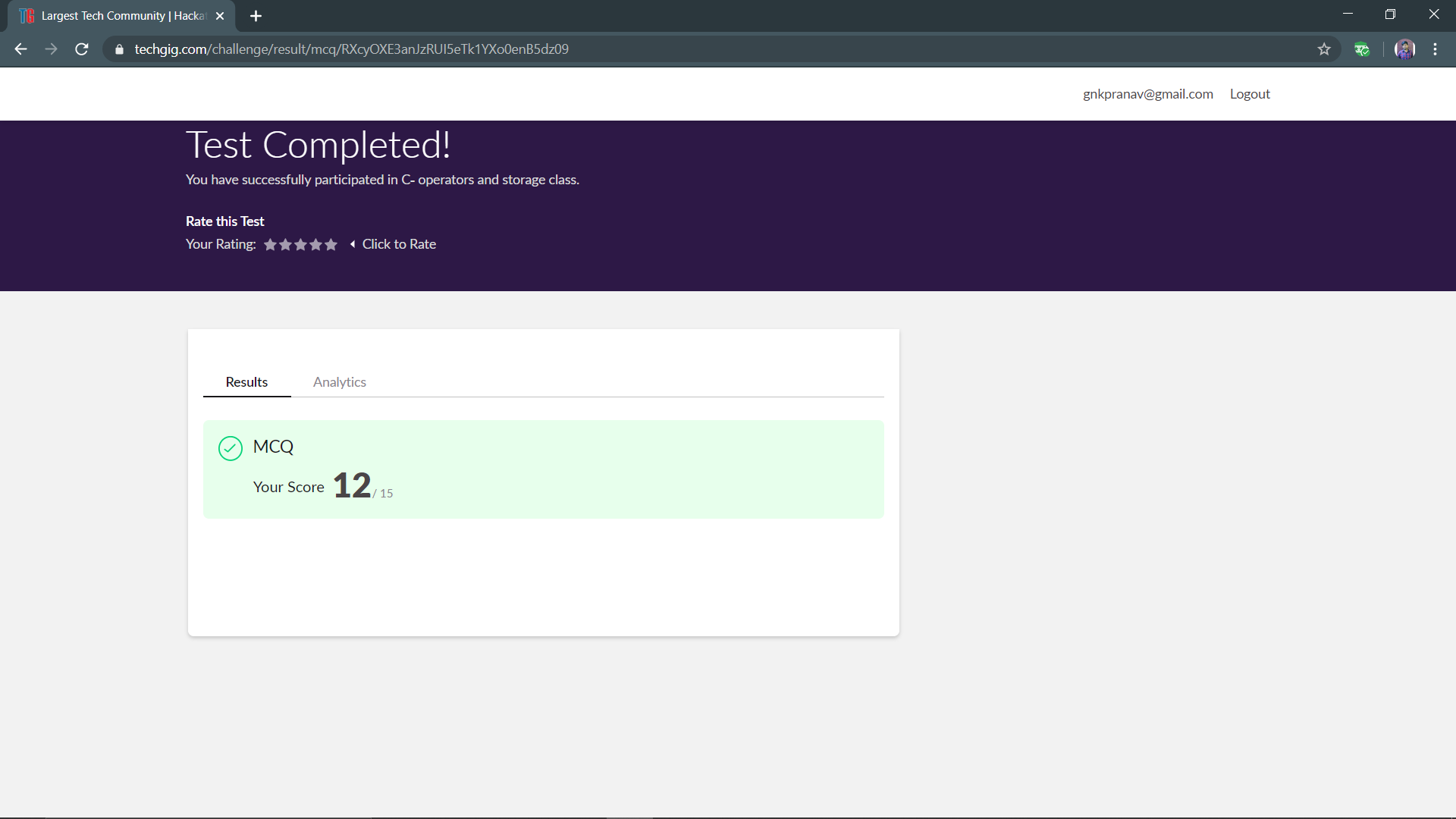
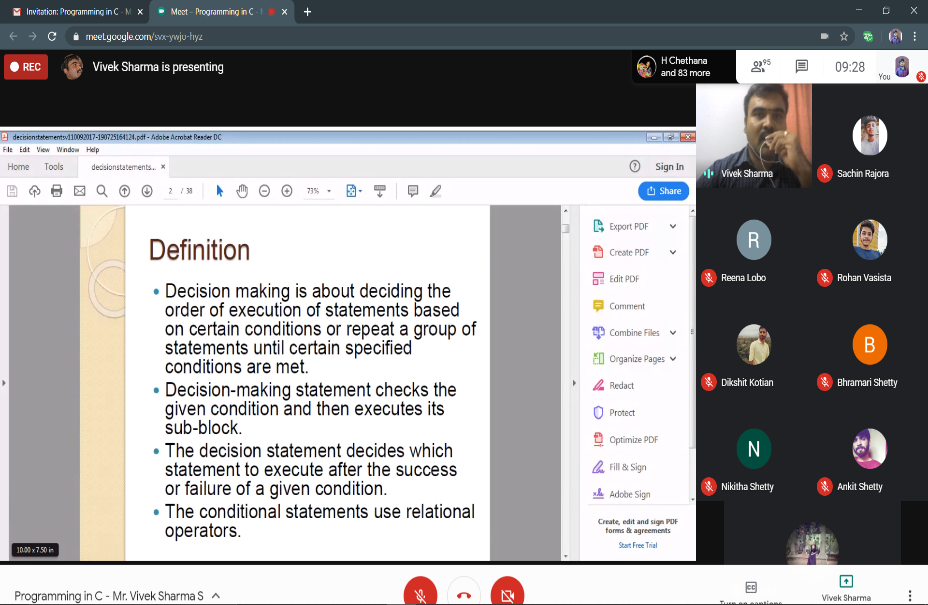
**DAILY ONLINE ACTIVITIES SUMMARY**

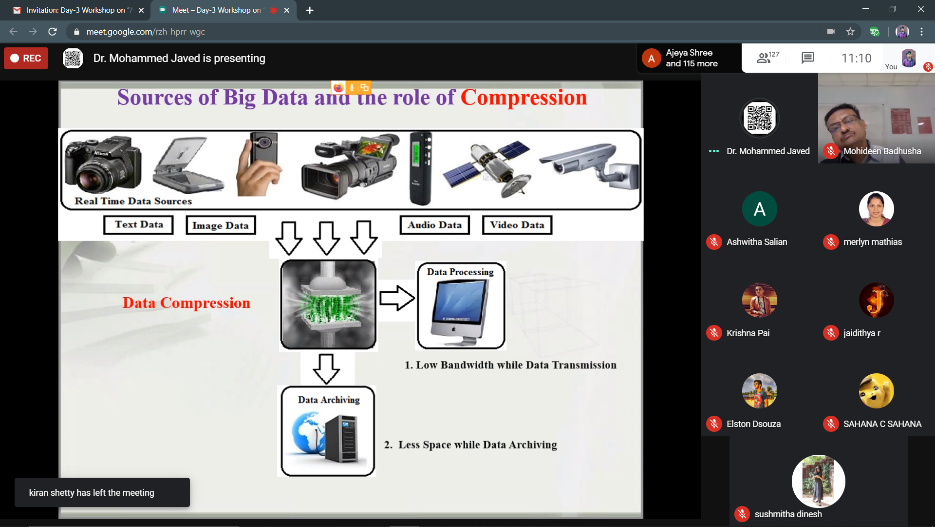
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date:** | **17-06-2020** | | | **Name:** | **SAHANA C** |
| **Sem & Sec** | **6th - B** | | | **USN:** | **4AL17CS116** |
| **PRE-PLACEMENT TRAINING** | | | | | |
| **Subject** | **C-Programing(decision making statements) and Python workshop** | | | | |
| **Max. Marks** | **--** | **Score** | | | **--** |
| **C-Structures**  **Python workshop** | 9:20 to 10:40 Mr. Vivek Sharma  11:00 to 1:40 Dr. Mohideen Badhusha | | | | |
| **Coding Challenges** | | | | | |
| **Problem Statemen:**   1. Java program to find the row, column position of a specified number (row, column position) in a given 2-dimensional array 2. Find the smallest positive integer value that cannot be represented as sum of any subset of a given array sorted in ascending order | | | | | |
| **Status: executed** | | | | | |
| **Uploaded the report in Github** | | | **yes** | | |
| **If yes Repository name** | | | **https://github.com/sahanasanu/Daliy-status** | | |
| **Uploaded the report in slack** | | | **Yes** | | |

ONLINE TEST DETAILS



Training:





Webinar:



Python workshop :

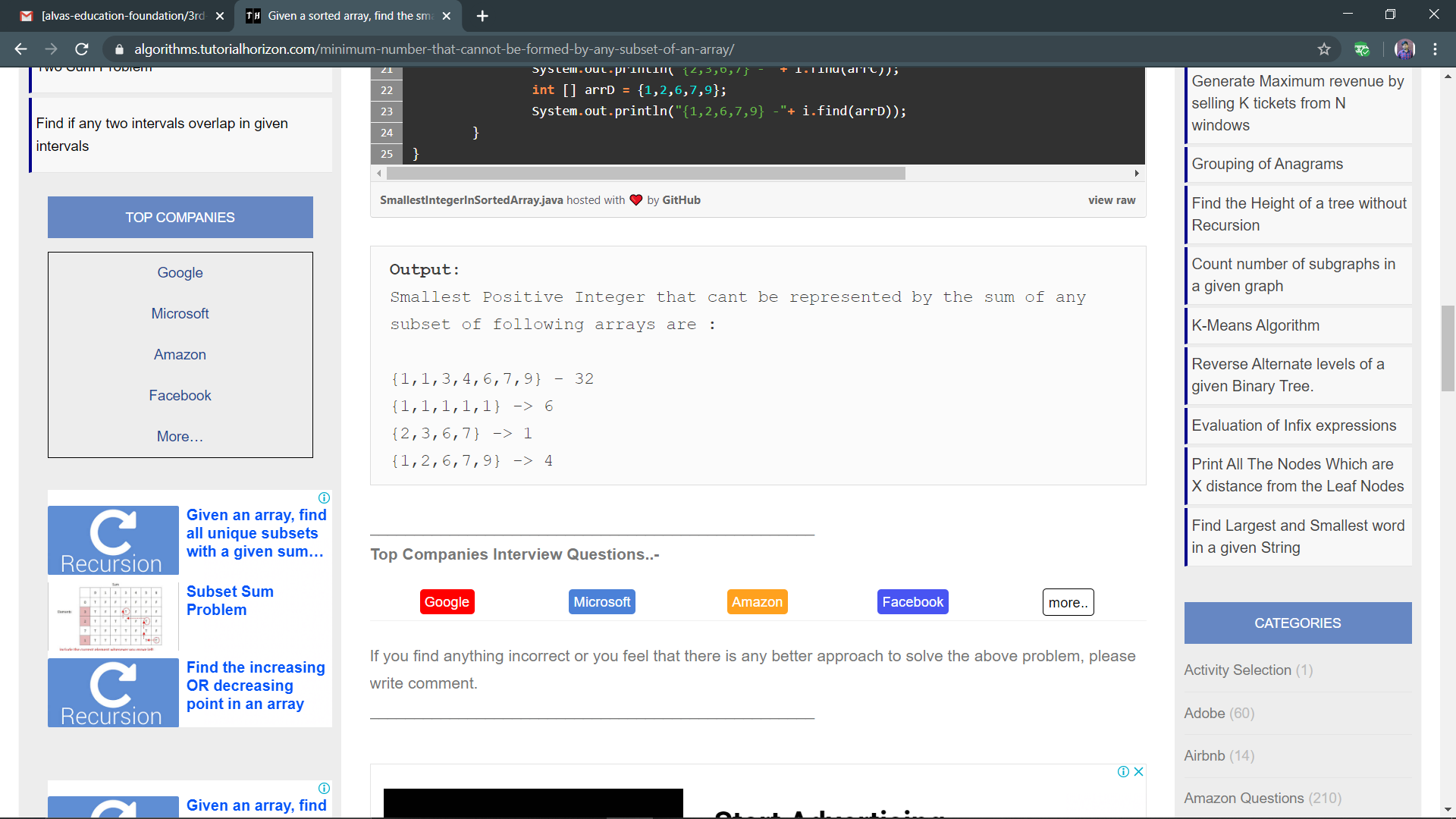
**Assessments:**

**Uploaded in github account and respective links are provided.**

[**https://github.com/sahanasanu/pre-placement-training**](https://github.com/sahanasanu/pre-placement-training)

**ONLINE CODDING CHALLENGE:**

|  |
| --- |
| Find the smallest positive integer value that cannot be repsented as sum  of any subset of a given array sorted in ascending order  public class  SmallestIntegerInSortedArray {  public int find(int [] arrA){ |
| int smlNumber = 1; |
| for(int i = 0;i<arrA.length;i++){ |
| if(arrA[i]<=smlNumber){ |
| smlNumber += arrA[i]; |
| }else{ |
| break; |
| } |
| } |
| return smlNumber; |
| } |
| public static void main(String arg[]){ |
| SmallestIntegerInSortedArray i = new SmallestIntegerInSortedArray(); |
| System.out.println("Smallest Positive Integer that cant be represented by  the sum of any subset of following arrays are : "); |
| int [] arrA = { 1,1,3,4,6,7,9}; |
| System.out.println("{1,1,3,4,6,7,9} -" + i.find(arrA)); |
| int [] arrB = {1,1,1,1,1}; |
| System.out.println("{1,1,1,1,1} -" + i.find(arrB)); |
| int [] arrC = {2,3,6,7}; |
| System.out.println("{2,3,6,7} -" + i.find(arrC)); |
| int [] arrD = {1,2,6,7,9}; |
| System.out.println("{1,2,6,7,9} -"+ i.find(arrD)); |
| } |
| } |



Write a Java program to find the row, column position of a specified number (row, column position) in a given 2-dimensional array

import java.util.\*;

public class abc {

public static void main(String[] args) {

int nums[][] = {{12, 20, 30, 40},

{15, 25, 35, 45},

{24, 29, 39, 51},

{35, 30, 39, 50},

{50, 60, 75, 72}};

int rows = 5;

int search\_element = 39;

int ans[] = Saddleback(nums, rows - 1, 0, search\_element);

System.out.println("Position of "+search\_element+" in the matrix is ("+ans[0] + "," + ans[1]+")");

}

private static int[] Saddleback(int nums[][], int row, int col, int search\_element) {

//numsay to store the row and column of the searched element

int element\_pos[] = {-1, -1};

if (row < 0 || col >= nums[row].length) {

return element\_pos;

}

if (nums[row][col] == search\_element) {

element\_pos[0] = row;

element\_pos[1] = col;

return element\_pos;

}

else if (nums[row][col] > search\_element) {

return Saddleback(nums, row - 1, col, search\_element);

}

return Saddleback(nums, row, col + 1, search\_element);

}

}

