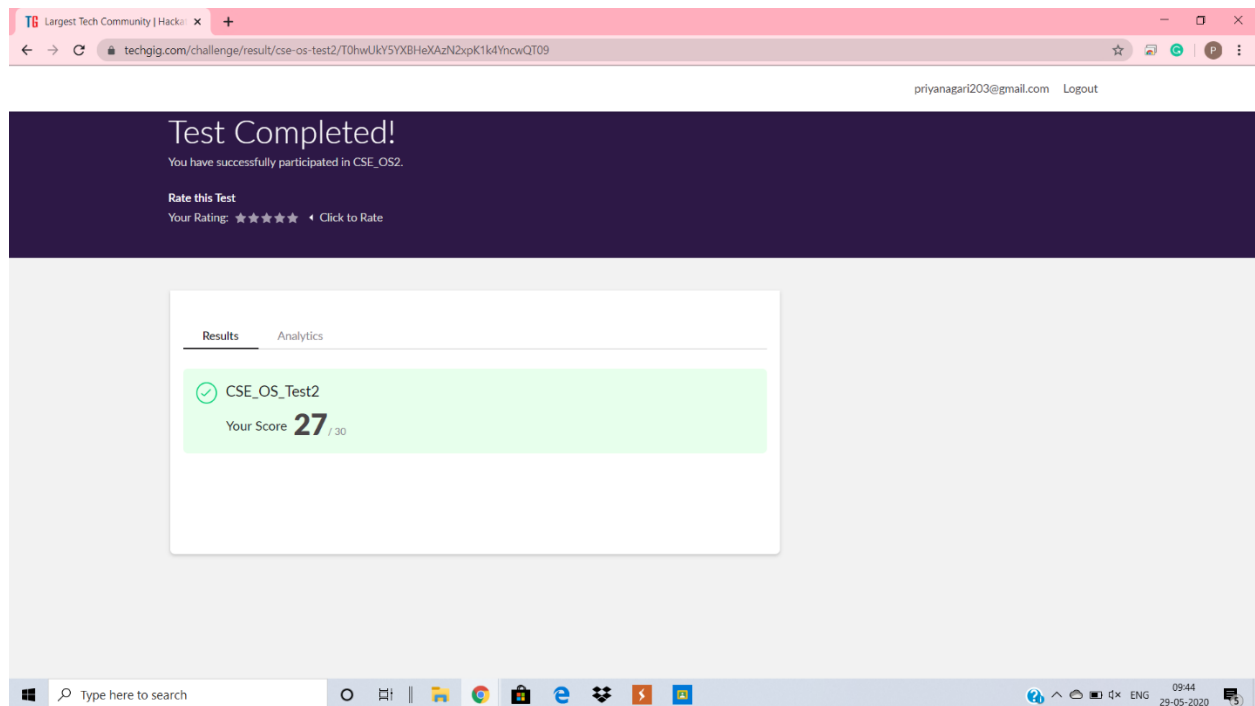


DAILY ONLINE ACTIVITIES SUMMARY

Date:	29/05/2020	Name:	Priya Nagari
Sem & Sec	Fourth SEM section B	USN:	4AL18CS063
Online Test Summary			
Subject	Operating System		
Max. Marks	30	Score	27
Certification Course Summary			
Course	The complete Android app development Masterclass: Build apps		
Certificate Provider	Udemy	Duration	29 hours
Coding Challenges			
<p>1. Given a N X N binary matrix, find the size of the largest '+' formed by all 1s.</p> <p>For above matrix, largest '+' would be formed by highlighted part of size 8.</p> <p>2. Armstrong number is a number that is equal to the sum of cubes of its digits. For example 0, 1, 153, 370, 371 and 407 are the Armstrong numbers</p>			
Status:			
Uploaded the report in Github		YES	
If yes Repository name		Priya_Nagari link: https://github.com/alvas-education-foundation/Priya_Nagari	
Uploaded the report in slack		YES	

Online Test Details: 2nd test

- The Operating system test was scheduled from 9:00AM to 9:45PM. The Portion for the IA was the 2nd module there were 30 questions of one mark & the time assigned was 40 minutes. The questions were mcq type.



Certification Course Details:

Name of the course: The complete Android app development Masterclass: Build apps

Certificate Provider: Udemy

total duration is 29 hours.

Today I completed the lectures about fragments and started with making gym application this concept I need to practice these concepts in Android studio ,so I am daily practicing and completing the challenges given on each topic.

The screenshot displays a web browser window showing a Udemy course page. The main content area features a code editor with the following Java code for `MainActivity`:

```
package org.meicode.debugging;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private static final String TAG = "MainActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(TAG, "onCreate: Debug");
        Log.e(TAG, "onCreate: ");
    }
}
```

Below the code editor, there are tabs for Overview, Q&A, Bookmarks, and Announcements. The 'About this course' section states: "Create Real World Applications using Java and Become A Professional Android App Developer From Scratch Today!". It also provides statistics: "By the numbers", "Skill level: All Levels", "Students: 40819", "Languages: English", "Cautions: Yes", "Lectures: 90", and "Video: 29 total hours".

On the right side, there is a 'Course content' sidebar listing various topics with checkboxes and durations:

- 54. Handle Permissions Correctly (25min)
- 55. Alarm Action - Calendar (26min)
- 56. Fragments (16min)
- 57. Callback Interfaces (20min)
- 58. Challenge - Gym Application (Part 1) (20min)
- 59. Challenge - Gym Application (Part 2) (23min)
- 60. Challenge - Gym Application (Part 3) (26min)
- 61. Challenge - Gym Application (Part 4) (17min)
- Section 6: Shortcuts and Debugging (2 / 2 | 31min)
 - 62. Shortcuts in Android Studio (15min)
 - 63. Exceptions - Logging - Basic Debugging (16min)
- Section 7: Handling Background Tasks

Online Coding Details:

Problem Statement 1: 1. Given a $N \times N$ binary matrix, find the size of the largest '+' formed by all 1s. For above matrix, largest '+' would be formed by highlighted part of size 8.

The screenshot shows a GitHub repository page for 'alvas-education-foundation / Priya_Nagari'. The file 'Largest_Plus.java' is selected, showing its code. The code is a Java program to find the largest '+' formed by 1's in a binary matrix. It includes a class 'largest_plus_formed' with a static method 'findLargestPlus' that takes a 2D array of integers and returns the size of the largest '+'.

```
1 import java.io.*;
2
3 class largest_plus_formed{
4
5     // size of binary square matrix
6     static int N = 10;
7
8     // Function to find the size of the largest '+'
9     // formed by all 1's in given binary matrix
10    static int findLargestPlus(int mat[][])
11    {
12
13        // left[i][j], right[i][j], top[i][j] and
14        // bottom[i][j] store maximum number of
15        // consecutive 1's present to the left,
16        // right, top and bottom of mat[i][j]
17        // including cell(i, j) respectively
18        int left[][] = new int[N][N];
19        int right[][] = new int[N][N];
20        int top[][] = new int[N][N];
21        int bottom[][] = new int[N][N];
```

Problem Statement 2: Armstrong number is a number that is equal to the sum of cubes of its digits. For example 0, 1, 153, 370, 371 and 407 are the Armstrong numbers

The screenshot shows a GitHub repository page for 'alvas-education-foundation / Priya_Nagari'. The file 'ArmStrong.c' is selected, showing its code. The code is a C program to check if a number is an Armstrong number. It includes a function 'check_armstrong' that takes an integer and returns 1 if it is an Armstrong number and 0 otherwise. The main function takes two integers 'a' and 'b' and prints the Armstrong numbers between them.

```
1
2 #include <stdio.h>
3 int check_armstrong(int);
4 int power(int, int);
5
6 int main ()
7 {
8     int c, a, b;
9
10    printf("Input two integers\n");
11    scanf("%d%d", &a, &b);
12
13    for (c = a; c <= b; c++)
14        if (check_armstrong(c) == 1)
15            printf("%d\n", c);
16
17    return 0;
18 }
19
20 int check_armstrong(int n) {
```