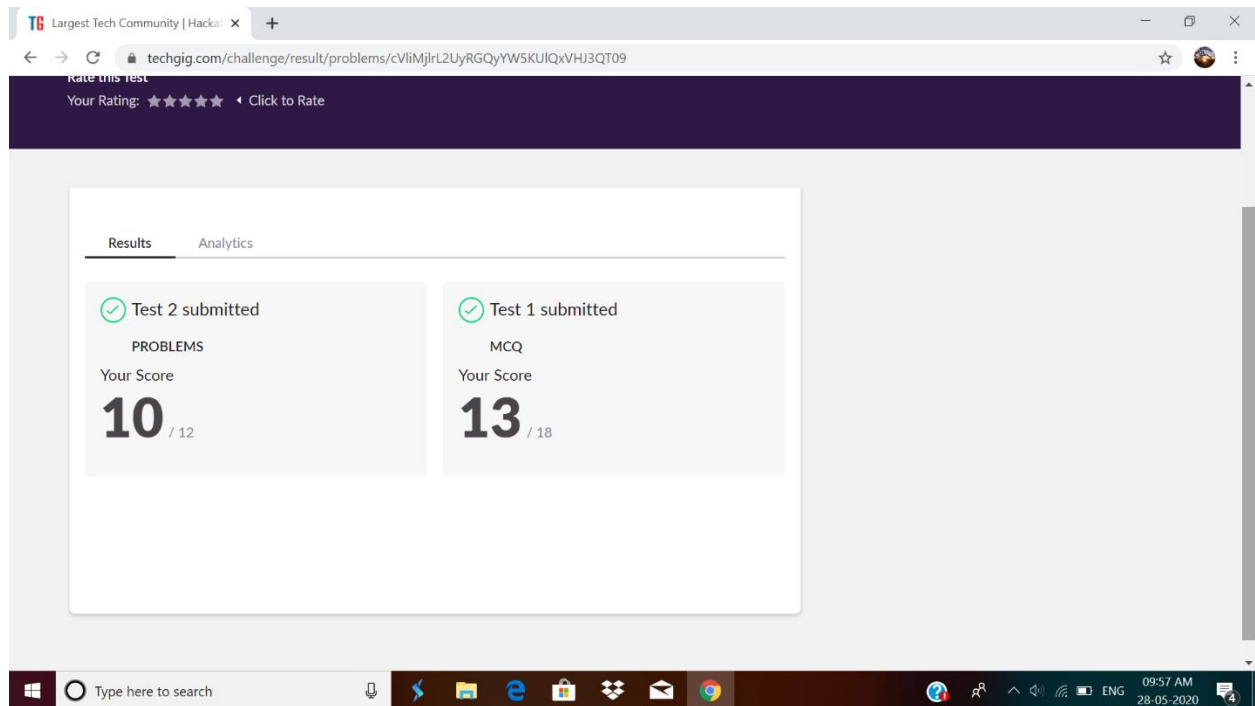


## DAILY ONLINE ACTIVITIES SUMMARY

Date:	28-05-2020	Name:	Anvitha Poojary
Sem & Sec	6A	USN:	4AL17CS008
<b>Online Test Summary</b>			
Subject	OS		
Max. Marks	30	Score	23
<b>Certification Course Summary</b>			
Course	Front end development-CSS		
Certificate Provider	greatlearning	Duration	5hr
<b>Coding Challenges</b>			
<b>Problem Statement:</b> 1. Python program to find digital root of a number 2. JAVA PROGRAM-BALANCED BRACKET			
<b>Status: completed</b>			
Uploaded the report in Github		Yes	
If yes Repository name		<a href="https://github.com/anvithapo99/Daily-Report">https://github.com/anvithapo99/Daily-Report</a>	
Uploaded the report in slack		Yes	

**Online test details:**

## Subject: OS



## Certification course details:

### Front end Development -CSS


Today I have studied following topics:

- Introduction to CSS
- Inline Styling
- Internal styling
- Internal styling with class
- External styling
- Type,id and class selector
- Attribute selector




## CSS



 1. Introduction to CSS  
1m




 Introduction to CSS  
Your Score: 0.33/1


 2. Inline Styling  
5m



 Inline Styling  
Your Score: 0/1

 3. Internal Styling  
6m



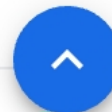
 4. Internal styling with class  
2m



 5. External Styling  
6m



 6. Type ID and Class Selector





Your score: 0.55 / 1



2. Inline Styling



5m



Inline Styling

Your Score: 0/1



3. Internal Styling



6m



4. Internal styling with class



2m



5. External Styling



6m



6. Type, ID and Class Selector



5m



7. Attribute Selector



6m



8. Attribute + Value Selector



## Coding Challenges Details:

### 1. Python program to find digital root of a number

Description:

A digital root is the recursive sum of all the digits in a number. Given n, take the sum of the digits of n. If that value has more than one digit, continue reducing in this way until a single-digit number is produced. This is only applicable to the natural numbers.

`digit_root(0)= 0`

`digital_root(16)`

`=> 1 + 6`

`=> 7`

`digital_root(132189)`

`=> 1 + 3 + 2 + 1 + 8 + 9`

`=> 24 ...`

`=> 2 + 4`

`=> 6`

```
def DigitalRoot(number):
```

```
    addper = 0
```

```
    while number >=10:
```

```
        number = sum(int(digit)for digit in str(number))
```

```
        addper +=1
```

```
    #I highly recommend using return instead of print, but for testing purposes  
    I used print
```

```
    print(number)
```

```
DigitalRoot(132189)
```

**Output:**

The screenshot shows a web browser window with the URL `onlinegdb.com/online_c_compiler`. The interface includes a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The code editor contains a Python script named `main.py` with the following content:

```
1 def DigitalRoot(number):
2     addper = 0
3     while number >= 10:
4         number = sum(int(digit) for digit in str(number))
5         addper += 1
6         #I highly recommend using return instead of print, but for testing purposes I used print
7         print(number)
8     DigitalRoot(132189)
9
10
```

Below the code editor, there is an input field and a console output area. The console shows the message: "...Program finished with exit code 0. Press ENTER to exit console."

## 2. JAVA PROGRAM-BALANCED BRACKET

Write a function that accepts a string consisting entirely of brackets ({} ) and returns whether it is balanced. Every "opening" bracket must be followed by a closing bracket of the same type. There can also be nested brackets, which adhere to the same rule.

`f('(){}{({})}{()})' // true`

`f('()){}' // false`

```
import java.util.Stack;

public class Main {

    public static void main(String[] args) {

        System.out.println(is_parentheses_balanced("(){}{({})}{()})");

    }

    public static boolean matchingPeer(char open , char close){

        if ( open == '(' && close == ' '){

            return true;

        }

    }

}
```

```

    }
    if ( open == '[' && close == '']){
        return true;
    }
    else{
        return false;
    }
}

```

```

public static boolean is_parentheses_balanced(String equation){

```

```

    char[] c = equation.toCharArray();
    Stack <Character> myStack= new Stack <Character> ();
    for (int i = 0; i < c.length; i++){
        if(c[i]=='(' || c[i] == '[' ){
            myStack.push(c[i]);
        }
        else if (c[i]== ')' || c[i]=='']){
            if(matchingPeer(myStack.peek(),c[i]) == true){
                myStack.pop();
            } else {
                return false;
            }
        }
    }
}

```

```

    }

}

if(myStack.isEmpty()){

    return true;

}

else {

    return false;

}

}

}

```

### Output:

The screenshot shows a web browser window with the URL `onlinegdb.com/online_c_compiler`. The page contains a code editor for Java. The code is as follows:

```

1 import java.util.Stack;
2 public class Main {
3     public static void main(String[] args) {
4         System.out.println(is_parentheses_balanced("()[]{}(){}{[]()[]}"));
5     }
6     public static boolean matchingPeer(char open , char close){
7         if ( open == '(' && close == ' '){
8             return true;
9         }
10        if ( open == '[' && close == ' '){

```

The output window shows the result of the program execution:

```

true

...Program finished with exit code 0
Press ENTER to exit console.

```