DAILY ONLINE ACTIVITIES SUMMARY

Date:	**31-05-2020**	**Name:**	**Rakesh M Kotian**
Sem & Sec	**8 th sec-b**	**USN:**	**4al16cs072**

Online Test Summary

Subject	**iot**		
Max. Marks	**20**	**Score**	**19**

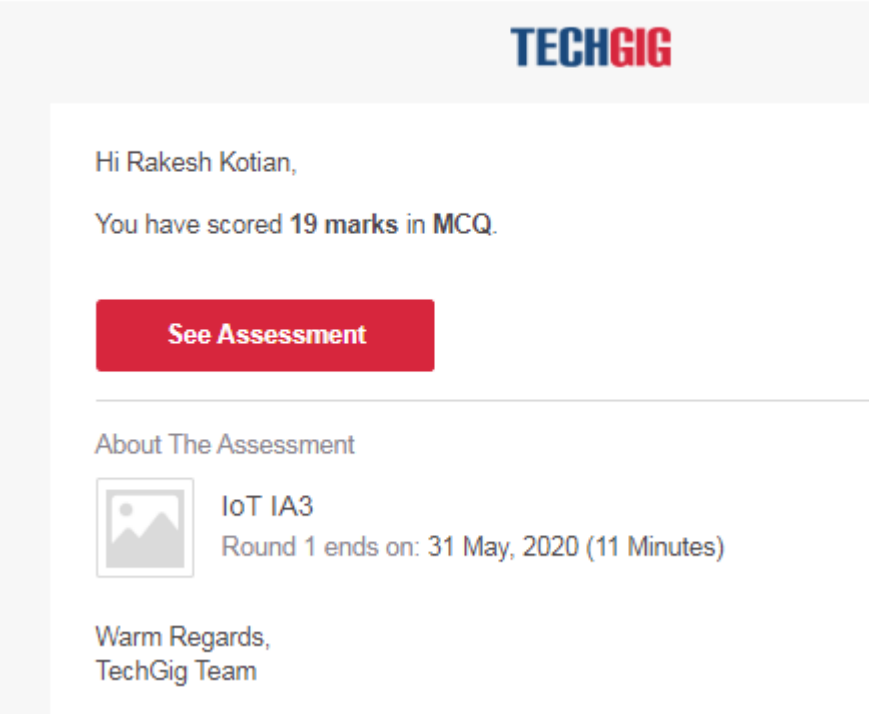
Certification Course Summary

Course	**Python for machine learning**		
Certificate Provider	**Great learning**	**Duration**	**6 hours**

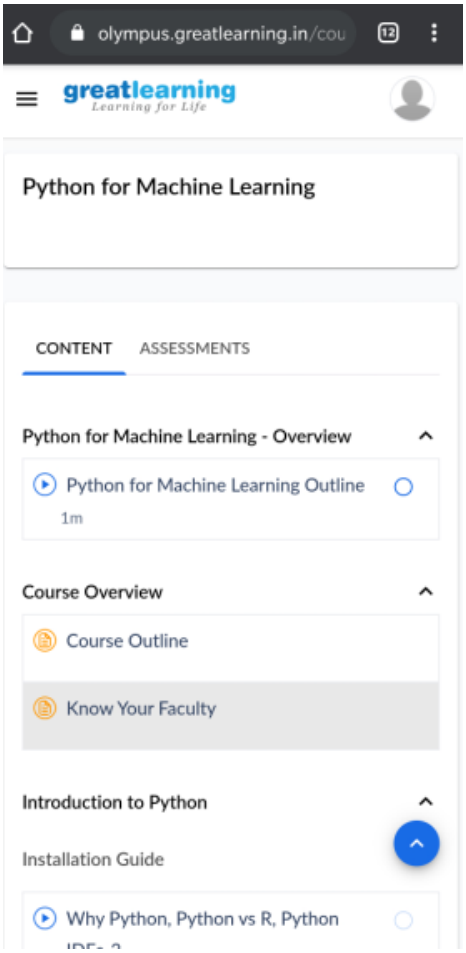
Coding Challenges

Problem Statement:	
Status:solved	
Uploaded the report in Github	**yes**
If yes Repository name	**Rakeshkotian08**
Uploaded the report in slack	**yes**

Online Test Details: (Attach the snapshot and briefly write the report for the same)



Certification Course Details: (Attach the snapshot and briefly write the report for the same)



Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)

Micro purchased an array *A* having *N* integer values. After playing it for a while, he got bored of it and decided to update value of its element. In one second he can increase value of each array element by *1*. He wants each array element's value to become greater than or equal to *K*. Please help Micro to find out the minimum amount of time it will take, for him to do so.

Input:

First line consists of a single integer, *T*, denoting the number of test cases. First line of each test case consists of two space separated integers denoting *N* and *K*. Second line of each test case consists of *N* space separated integers denoting the array *A*.

Output:

For each test case, print the minimum time in which all array elements will become greater than or equal to *K*. Print a new line after each test case.

Constraints:

$1 \le T \le 5$

$1 \le N \le 105$

$1 \le A[i], K \le 106$

```
#include<stdio.h>
int main()
{
   int t,n,i,min=10000000000,k;
   scanf("%d",&t);
   while(t--)
   {
    min=10000000000;
    scanf("%d%d",&n,&k);
    int a[n];
    for(i=0;i<n;i++)
    {
     scanf("%d",&a[i]);
     if(a[i]<min)
      min=a[i];
    }
    //printf("%d ",min);
    if(min>=k)
     printf("0\n");
    else
     printf("%d\n",k-min);
   }
   return 0;

}
```