# DAILY ONLINE ACTIVITIES SUMMARY

| | | | |
|---|---|---|---|
| **Date:** | **08/06/2020** | **Name:** | **Shruthi** |
| **Sem & Sec** | **8 B** | **USN:** | **4AL16CS100** |

## Online Test Summary

| | | | |
|---|---|---|---|
| **Subject** | **SMS** | | |
| **Max. Marks** | **60** | **Score** | **60** |

## Certification Course Summary(Internship)

| | | | |
|---|---|---|---|
| **Task** | **Creating an UI using an ETL tool and customizing the UI features and validations using Talend product.** | | |
| **Company** | **Gain-Insights** | **Duration** | **8 hrs** |

## Coding Challenges

**Problem Statement:**

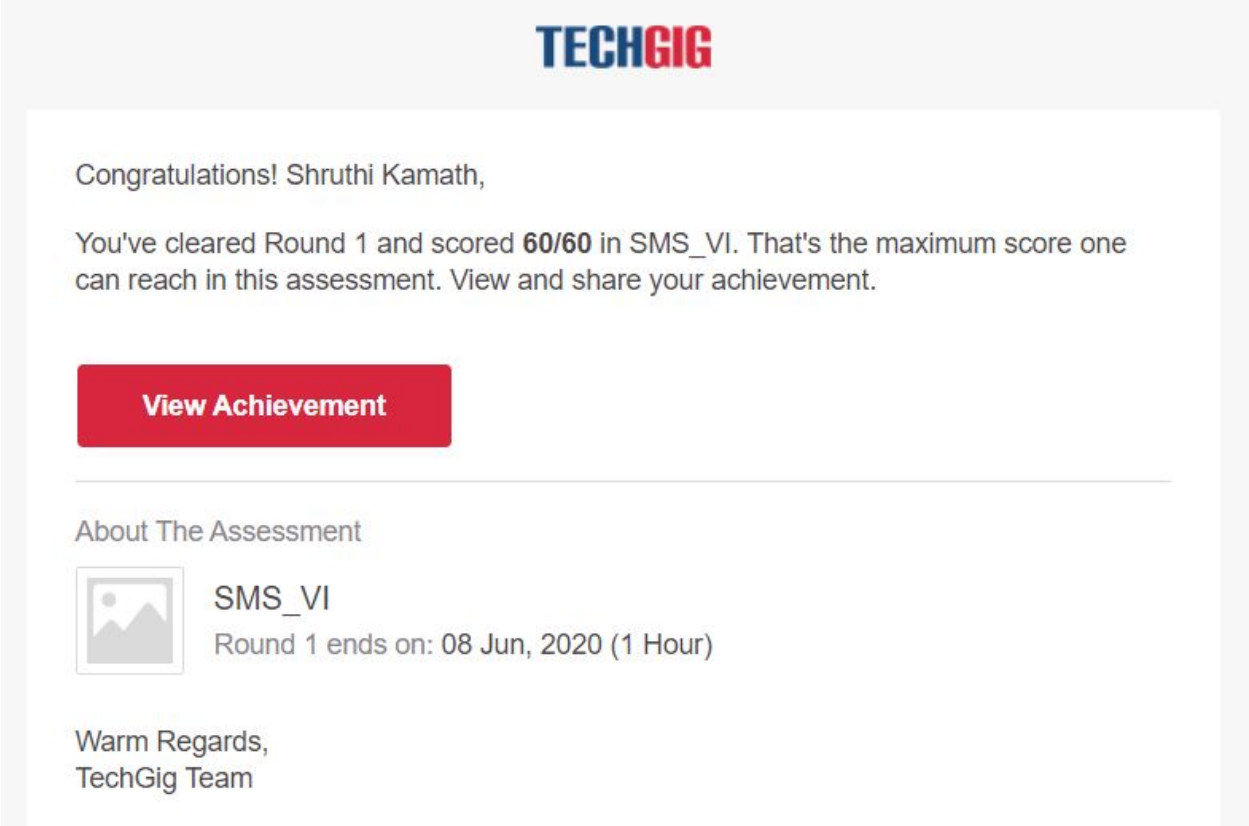
**1) generate all unique partition of integer**

**Status:completed**

| | |
|---|---|
| **Uploaded the report in Github** | **Yes** |
| **If yes Repository name** | alvas-education-foundation/ shruthikamath |

| | |
|---|---|
| **Uploaded the report in slack** | **Yes** |

## ONLINE TEST




## CODING CHALLENGE:

PROGRAM 1 :

```
def printArray(p, n):
```

```
    for i in range(0, n):
        print(p[i], end = " ")
    print()

def printAllUniqueParts(n):
    p = [0] * n     # An array to store a partition
    k = 0         # Index of last element in a partition
    p[k] = n     # Initialize first partition
                 # as number itself
    while True:
            printArray(p, k + 1)
            rem_val = 0
            while k >= 0 and p[k] == 1:
                rem_val += p[k]
                k -= 1
            if k < 0:
                print()
                return
            p[k] -= 1
            rem_val += 1

            while rem_val > p[k]:
```

```
            p[k + 1] = p[k]
            rem_val = rem_val - p[k]
            k += 1

        p[k + 1] = rem_val
        k += 1

print('All Unique Partitions of 2')
printAllUniqueParts(2)

print('All Unique Partitions of 3')
printAllUniqueParts(3)

print('All Unique Partitions of 4')
printAllUniqueParts(4)
```