DAILY ONLINE ACTIVITIES SUMMARY

Date:	08-06-2020	Name:	Deepika K V
Sem & Sec	8th sem 'A' sec	USN:	4AL16CS030

Online Test Summary			
Subject	SMS		
Max. Marks	60	Score	60

Certification Course Summary			
Course	Getting Started: Hadoop		
Certificate Provider	GreatLearning	Duration	5.5 hrs

Coding Challenges

Problem Statement: Write a C program to check if the number is friendly pair or not.

Status: SUBMITTED

Uploaded the report in Github	YES
If yes Repository name	Codes
Uploaded the report in slack	YES

Online test details:

Deepika K.V., Round 1 cleared ▶ Inbox ×

🖨 ↗

TechGig <user@techgig.com> Unsubscribe
to me ▾

Mon, Jun 8, 3:43 PM (1 day ago) ☆ ↩ ⋮

TECHGIG

Congratulations! Deepika K.V.,

You've cleared Round 1 and scored **60/60** in SMS_VI. That's the maximum score one can reach in this assessment. View and share your achievement.

View Achievement

About The Assessment

SMS_VI
Round 1 ends on: 08 Jun, 2020 (1 Hour)

Warm Regards,
TechGig Team

Certification Course Details:

olympus.greatlearning.in/courses/12888

greatlearning *Learning for Life*   Home   Live Sessions   Certificates

📘 My Courses

Getting Started : Hadoop

Course In Progress

CONTENT   ASSESSMENTS

Hadoop: Master your Big Data

▶ Big Data Touch	23m	◯
▶ Getting Started: Hadoop	14m	◯
📎 What is hadoop.ppt		⤓
▶ Hadoop framework : Stepping into Hadoop	24m	◯
▶ HDFS: What and Why?	20m	◯

Coding Challenge:

```c
#include<stdio.h>


                void printArray(int p[], int n)
                {
                    for (int i = 0; i < n; i++)
                       printf("%d ",p[i]);
                    printf("\n");
                }



                void partition(int n)
                {
                    int p[n], true=1;
                    int k = 0;
                    p[k] = n;



                    while (true)
                    {
                        printArray(p, k+1);



                        int rem_val = 0;
                        while (k >= 0 && p[k] == 1)
                        {
                            rem_val += p[k];
                            k--;
                        }



                        if (k < 0)  return;



                        p[k]--;
                        rem_val++;



                        while (rem_val > p[k])
                        {
                            p[k+1] = p[k];
```

```c
            rem_val = rem_val - p[k];
            k++;
        }


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


int main()
{
    int n;
    printf("Enter the number: ");
    scanf("%d",&n);
    partition(n);


    return 0;
}
```