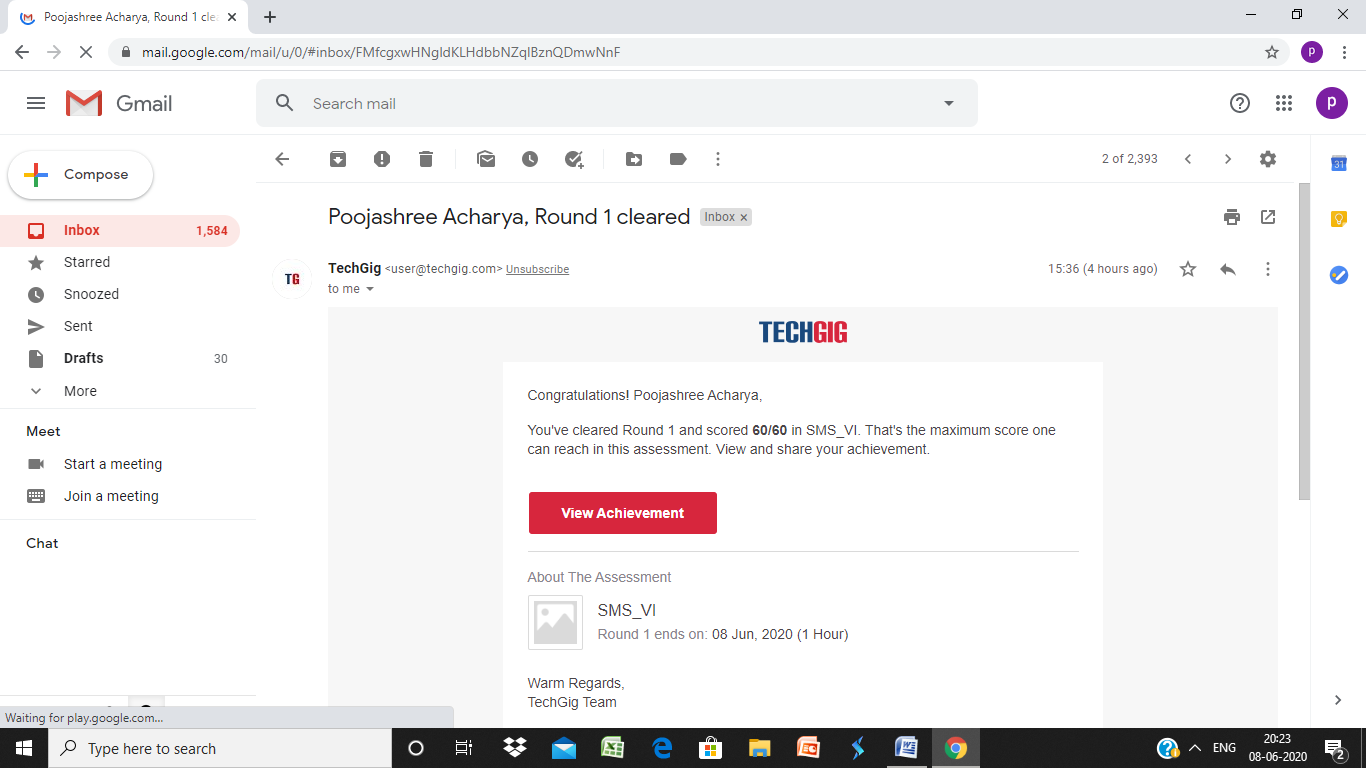
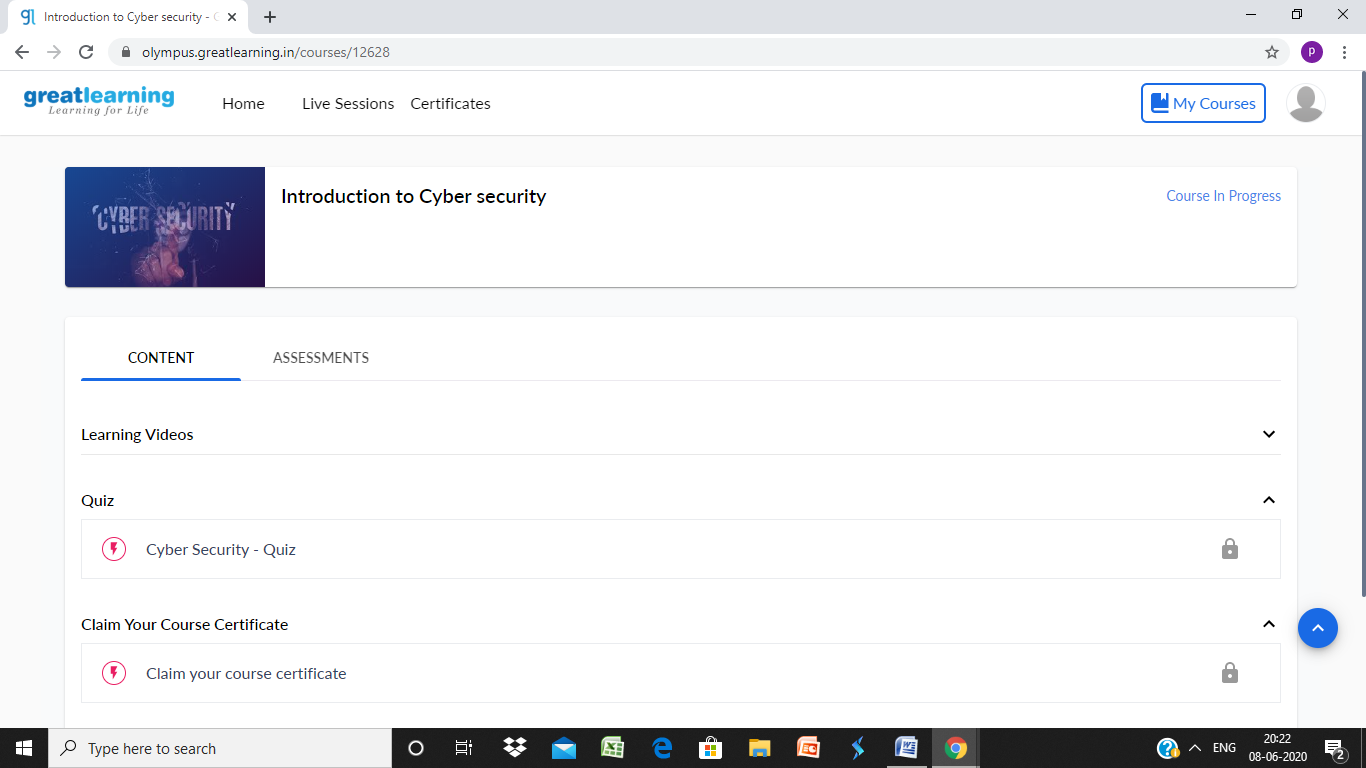
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **8-6-2020** | | | | | **Name:** | **poojashree** | |
| **Sem & Sec** | **8th sem A sec** | | | | | **USN:** | **4al16cs065** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **sms** | | | | | | |
| **Max. Marks** | | **60** | | **Score** | | | **60** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Introduction to cyber security** | | | | | | | |
| **Certificate Provider** | | | **Great learning academy** | | **Duration** | | | **5.5hr** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:**  **1.** **C Program to Generate All the Set Partitions of n Numbers Beginning from 1 and so on** | | | | | | | | |
| **Status:completed** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | **Poojashree** | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | |

**Online test**



**Certification course**

****

**Coding**

**Program 1**

1. #include <stdio.h>
2. #include <stdlib.h>
3. typedef struct {
4. int first;
5. int n;
6. int level;
7. } Call;

10. void print(int n, int \* a) {
11. int i ;
12. for (i = 0; i <= n; i++) {
13. printf("%d", a[i]);
14. }
15. printf("**\n**");
16. }

19. void integerPartition(int n, int \* a){
20. int first;
21. int i;
22. int top = 0;
23. int level = 0;
24. Call \* stack = (Call \* ) malloc (sizeof(Call) \* 1000);
25. stack[0].first = -1;
26. stack[0].n = n;
27. stack[0].level = level;
28. while (top >= 0){
29. first = stack[top].first;
30. n = stack[top].n;
31. level = stack[top].level;
32. if (n >= 1) {
33. if (first == - 1) {
34. a[level] = n;
35. print(level, a);
36. first = (level == 0) ? 1 : a[level-1];
37. i = first;
38. } else {
39. i = first;
40. i++;
41. }
42. if (i <= n / 2) {
43. a[level] = i;
44. stack[top].first = i;
45. top++;
46. stack[top].first = -1;
47. stack[top].n = n - i;
48. stack[top].level = level + 1;
49. } else {
50. top--;
51. }
52. } else {
53. top --;
54. }
55. }
56. }
58. int main(){
59. int N = 1;
60. int \* a = (int \* ) malloc(sizeof(int) \* N);
61. int i;
62. printf("**\n**Enter a number N to generate all set partition from 1 to N: ");
63. scanf("%d", &N);
64. for ( i = 1; i <= N; i++)
65. {
66. printf("**\n**Integer partition for %d is: **\n**", i);
67. integerPartition (i, a);
68. }
69. return(0);
70. }