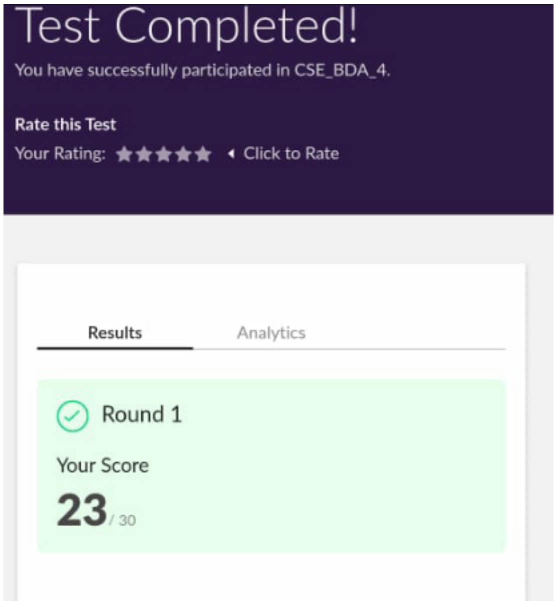
DAILY ONLINE ACTIVITIES SUMMARY

Date:	29-05-2020	Name:	Sinchana Kamath
Sem & Sec	8th sem B sec	USN:	4AL16CS102
Online Test Summary			
Subject	BDA		
Max. Marks	30	Score	23
Certification Course Summary			
Course	Introduction to ethical hacking		
Certificate Provider	https://www.greatlearning.in/academy	Duration	6hrs
Coding Challenges			
Problem Statement:Given an array arr[] of size N and an integer K. The task is to find the last remaining element in the array after reducing the array.			
Status: completed			
Uploaded the report in Github		yes	
If yes Repository name		sinchana Kamath	
Uploaded the report in slack		yes	

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



Test Completed!

You have successfully participated in CSE_BDA_4.

Rate this Test

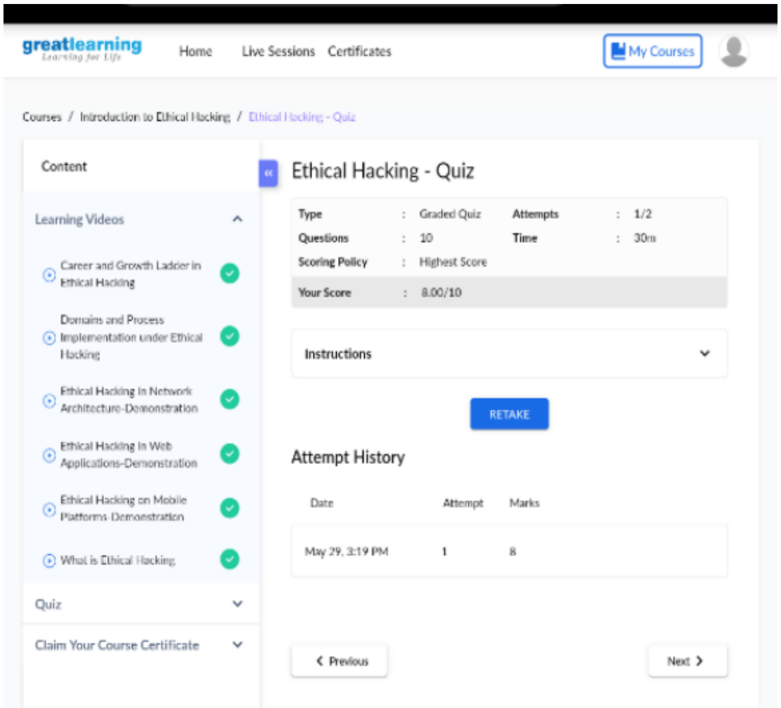
Your Rating: ★★★★★ Click to Rate

Round 1

Your Score

23/ 30

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



Courses / Introduction to Ethical Hacking / Ethical Hacking - Quiz

Content

Learning Videos

- Career and Growth Ladder in Ethical Hacking ✓
- Domains and Process Implementation under Ethical Hacking ✓
- Ethical Hacking in Network Architecture-Demonstration ✓
- Ethical Hacking in Web Applications-Demonstration ✓
- Ethical Hacking on Mobile Platforms-Demonstration ✓
- What is Ethical Hacking ✓

Quiz

Claim Your Course Certificate

Ethical Hacking - Quiz

Type	:	Graded Quiz	Attempts	:	1/2
Questions	:	10	Time	:	30m
Scoring Policy	:	Highest Score			
Your Score	:	8.00/10			

Instructions

Attempt History

Date	Attempt	Marks
May 29, 3:19 PM	1	8

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

Coding was given n it was uploaded for github and slack

Program 1....

```
void moreThanNdK(int arr[], int n, int k)
{
   // k must be greater than 1 to get some output
   if (k < 2)
      return;

   /* Step 1: Create a temporary array (contains element
      and count) of size k-1. Initialize count of all
      elements as 0 */
   struct eleCount temp[k-1];
   for (int i=0; i<k-1; i++)
       temp[i].c = 0;

   /* Step 2: Process all elements of input array */
   for (int i = 0; i < n; i++)
   {
       int j;
```

```
/* If arr[i] is already present in
   the element count array, then increment its count */
for (j=0; j<k-1; j++)
{
    if (temp[j].e == arr[i])
    {
         temp[j].c += 1;
         break;
    }
}

/* If arr[i] is not present in temp[] */
if (j == k-1)
{
    int l;

    /* If there is position available in temp[], then place
      arr[i] in the first available position and set count as 1*/
    for (l=0; l<k-1; l++)
    {
        if (temp[l].c == 0)
        {
                temp[l].e = arr[i];
                temp[l].c = 1;
                break;
            }
        }

        /* If all the position in the temp[] are filled, then
           decrease count of every element by 1 */
        if (l == k-1)
            for (l=0; l<k; l++)
                temp[l].c -= 1;
    }
}

/*Step 3: Check actual counts of potential candidates in temp[]*/
for (int i=0; i<k-1; i++)
{
    // Calculate actual count of elements
    int ac = 0;  // actual count
    for (int j=0; j<n; j++)
        if (arr[j] == temp[i].e)
            ac++;
        // If actual count is more than n/k, then print it
        if (ac > n/k)
           cout << "Number:" << temp[i].e
                << " Count:" << ac << endl;
    }
}
```