

DAILY ONLINE ACTIVITIES SUMMARY

Date:	02-06-2020	Name:	Anvitha Poojary
Sem & Sec	6A	USN:	4AL17CS008
Online Test Summary			
Subject	CGV		
Max. Marks	30	Score	26
Certification Course Summary			
Course	Python for data science		
Certificate Provider	COGNITIVE CLASS .ai	Duration	5hr
Coding Challenges			
Problem Statement: 1. Python program to return a list containing first and last element using list slicing method 2. Write a program to check if given linked list has a loop or not. 3. Given an array of positive integers. Write a C Program to find inversion count of array.			
Status: completed			
Uploaded the report in Github		Yes	
If yes Repository name		https://github.com/anvithapo99/Daily-Report	
Uploaded the report in slack		Yes	

Online test details:

Subject: CGV

CGV Test Total points 26/30 ?

Mention your name and USN without fail, otherwise your form will be rejected.
Choose the correct answer. Don't choose multiple answers.
Each question carries ONE mark and Maximum duration is 30 minutes.
Submission of more than one form is not allowed.
Submit the form before 10.00 AM, otherwise it will be rejected.

Name
Anvitha Poojary

USN
4AL17CS008

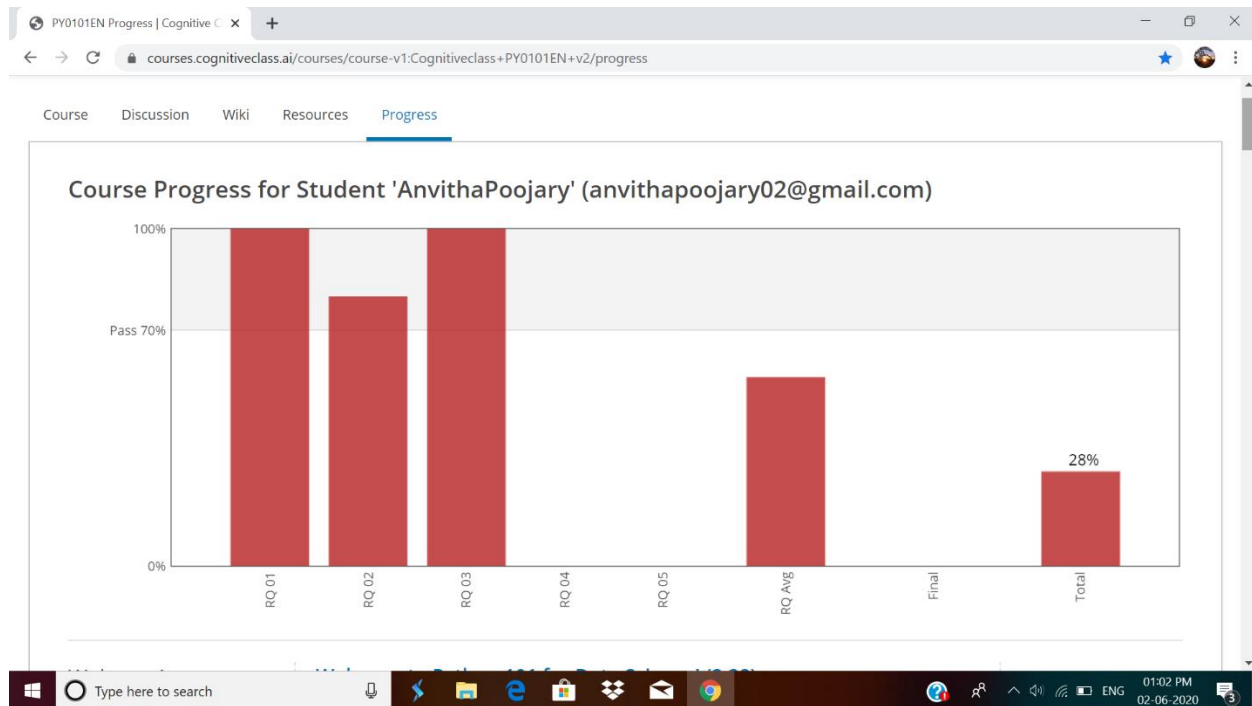
✓ To obtain a display of a three-dimensional world-coordinate scene, we first set up a coordinate reference for 1/1

Certification course details:

Python for data science

Today I have studied following topics:

- Creation of Loops
- Creation of class
- Methods
- Object
- Some example program
- Functions



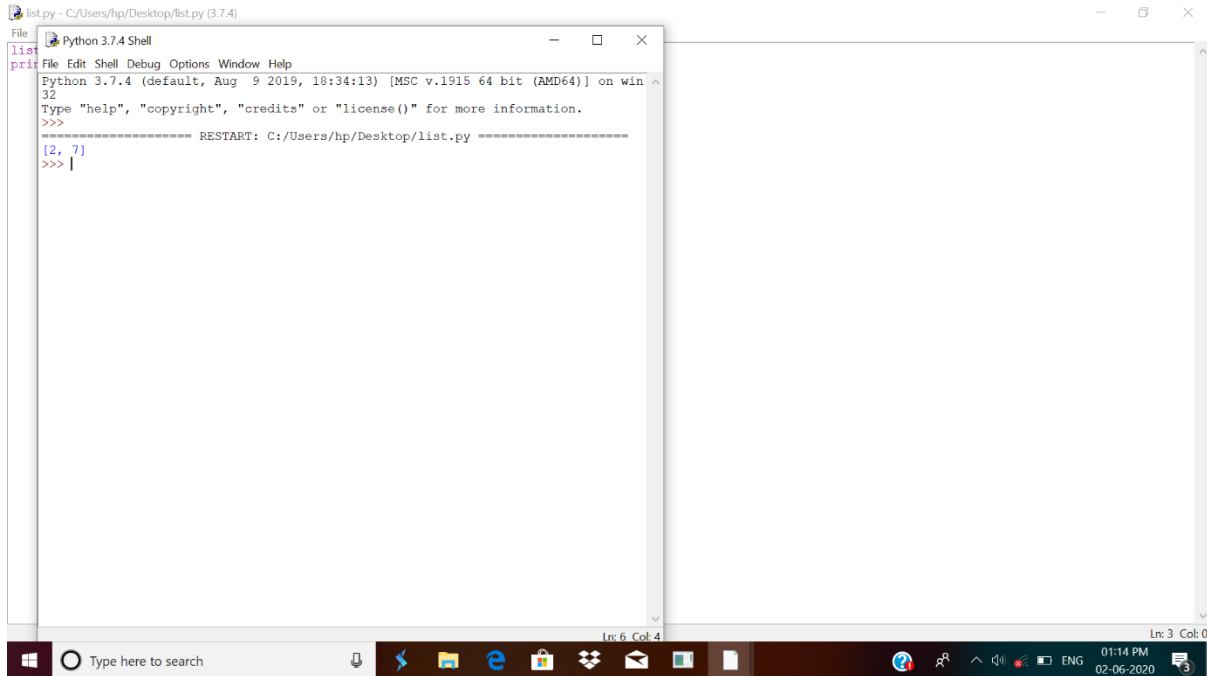
Coding Challenges Details:

1. Python program to return a list containing first and last element using list slicing method

```
list=[2,4,5,6,7]
```

```
print(list[0::4])
```

output:



2. Write a program to check if given linked list has a loop or not.

```
package prog11;
public class LinkedList{

    private Node head;

    private static class Node {
        private int value;
        private Node next;

        Node(int value) {
            this.value = value;
        }
    }

    public void addToTheLast(Node node) {

        if (head == null) {
            head = node;
        } else {
            Node temp = head;
            while (temp.next != null)
                temp = temp.next;

            temp.next = node;
        }
    }

    public void printList() {
```

```

        Node temp = head;
        while (temp != null) {
            System.out.format("%d ", temp.value);
            temp = temp.next;
        }
        System.out.println();
    }

    public boolean ifLoopExists() {
        Node fastPtr = head;
        Node slowPtr = head;
        while (fastPtr != null && fastPtr.next != null) {
            fastPtr = fastPtr.next.next;
            slowPtr = slowPtr.next;
            if (slowPtr == fastPtr)
                return true;
        }
        return false;
    }

    public static void main(String[] args) {
        LinkedList list = new LinkedList();

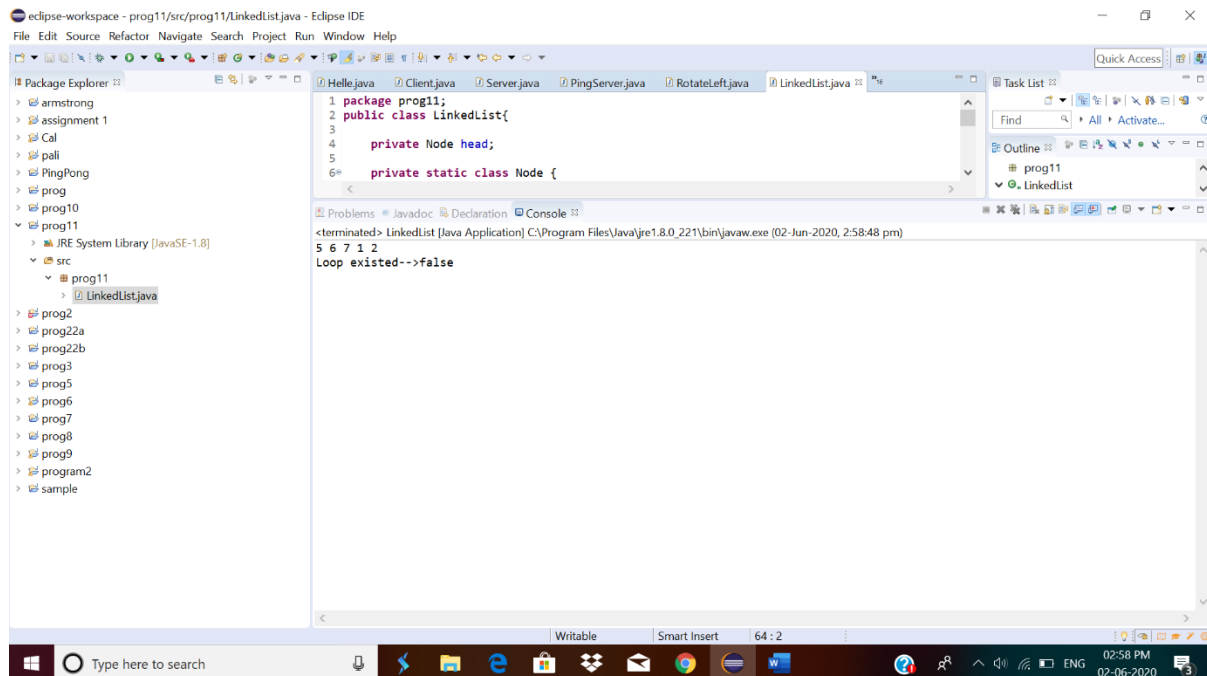
        list.addToTheLast(new Node(5));
        list.addToTheLast(new Node(6));
        list.addToTheLast(new Node(7));
        list.addToTheLast(new Node(1));
        list.addToTheLast(new Node(2));

        list.printList();

        System.out.println("Loop existed-->" + list.ifLoopExists());
    }
}

```

Output:



3. Given an array of positive integers. Write a C Program to find inversion count of array.

Inversion Count: For an array, inversion count indicates how far (or close) the array is from being sorted. If array is already sorted then inversion count is 0. If array is sorted in reverse order that inversion count is the maximum.

Formally, two elements $a[i]$ and $a[j]$ form an inversion if $a[i] > a[j]$ and $i < j$.

Input:

The first line of input contains an integer T denoting the number of test cases. The first line of each test case is N , the size of array. The second line of each test case contains N elements.

Output:

Print the inversion count of array.

Constraints:

$1 \leq T \leq 100$

$1 \leq N \leq 10^7$

$1 \leq C \leq 10^{18}$

Example:

Input:

1

5

2 4 1 3 5

Output:

3

Explanation:

Testcase 1: The sequence 2, 4, 1, 3, 5 has three inversions (2, 1), (4, 1), (4, 3)

```
#include <stdio.h>

int getInvCount(int arr[], int n)
{
    int i,j;

    int inv_count = 0;

    for ( i = 0; i < n - 1;i++)
        for ( j = i + 1; j < n; j++)
            if (arr[i] > arr[j])
                inv_count++;

    return inv_count;
}

int main(int argv, char** args)
{
    int arr[] = { 2,4,1,3,5 };

    int n = sizeof(arr) / sizeof(arr[0]);

    printf(" Number of inversions are %d \n", getInvCount(arr, n));

    return 0;
}
```

Output:

Online C Compiler - online editor × +

onlinegdb.com/online_c_compiler

Run Debug Stop Share Save {} Beautify

Language C

main.c

```
1 #include <stdio.h>
2 int getInvCount(int arr[], int n)
3 {
4     int i,j;
5     int inv_count = 0;
6     for ( i = 0; i < n - 1;i++)
7         for ( j = i + 1; j < n; j++)
8             if (arr[i] > arr[j])
9                 inv_count++;
10
11     return inv_count; |
12 }
13 int main(int argv, char** args)
14 {
15     int arr[] = { 2,4,1,3,5 };
```

input

Number of inversions are 3

...Program finished with exit code 0
Press ENTER to exit console.

Type here to search

03:13 PM
02-06-2020