**CP Club 365Days Challenge**

**Programming language – C++**

**Problem Statement**

https://practice.geeksforgeeks.org/problems/equilibrium-point-1587115620/1?page=1&curated[]=1&sortBy=submissions

**Your Code**:

// 0x66Day of 0x365Days challenge

// VEDANT BHARAD

// 22-12-2022

//{ Driver Code Starts

#include <iostream>

using namespace std;

// } Driver Code Ends

class Solution{

    public:

    // Function to find equilibrium point in the array.

    // a: input array

    // n: size of array

    int equilibriumPoint(long long a[], int n) {

        // Your code here

        if(n==1)

            return 1;

        long total = 0;

        long sum=0;

        for(int loop=0; loop<n; loop++){

            total+=a[loop];

        }

        for(int loop=1; loop<n; loop++)

        {

            sum+=a[loop-1];

            if(total-sum-a[loop]==sum) {

                return loop+1;

            }

        }

        return -1;

    }

};

//{ Driver Code Starts.

int main() {

    long long t;

    //taking testcases

    cin >> t;

    while (t--) {

        long long n;

        //taking input n

        cin >> n;

        long long a[n];

        //adding elements to the array

        for (long long i = 0; i < n; i++) {

            cin >> a[i];

        }

        Solution ob;

        //calling equilibriumPoint() function

        cout << ob.equilibriumPoint(a, n) << endl;

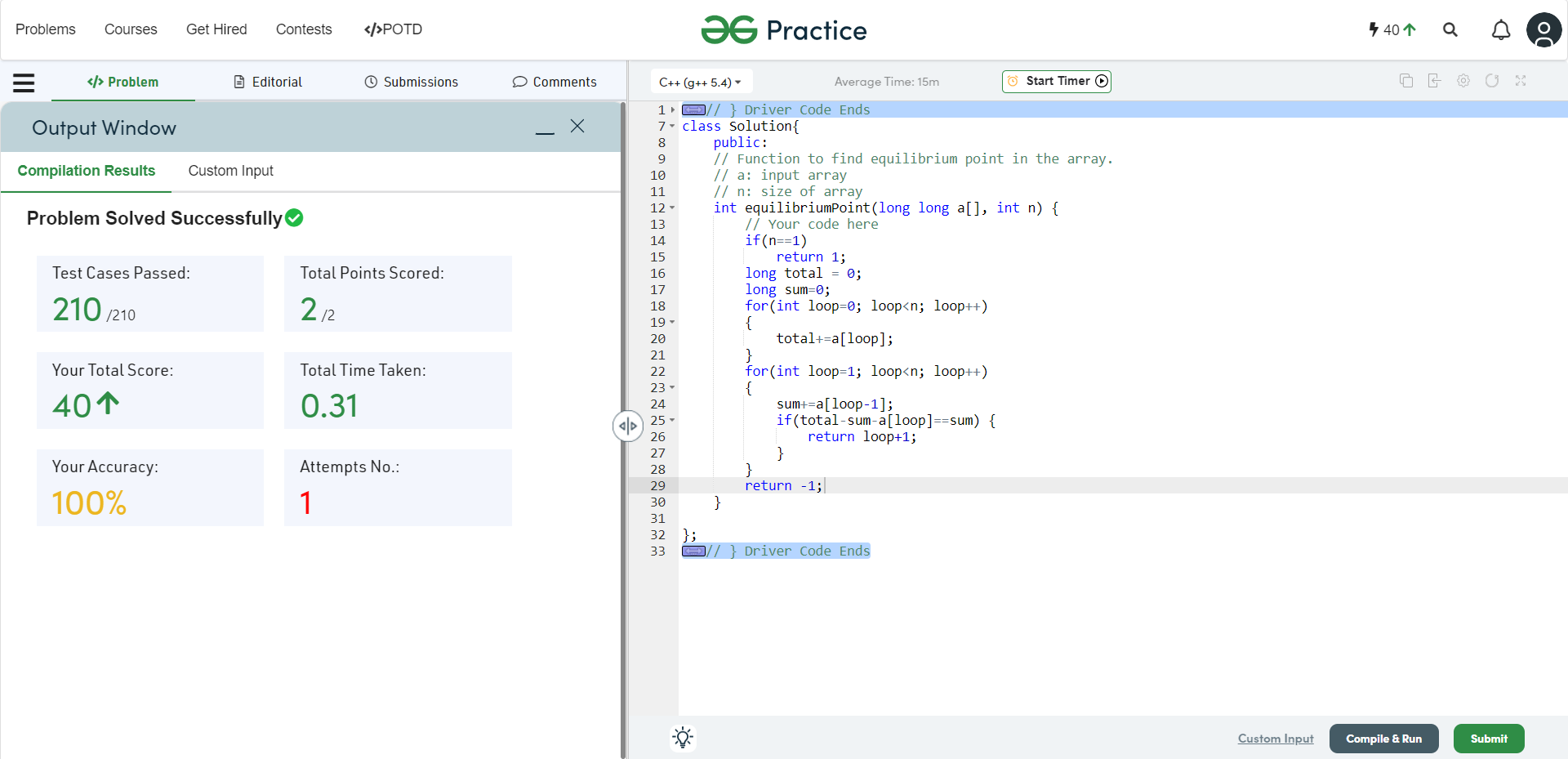
    }

    return 0;

}

// } Driver Code Ends

**Output (Screen Shot)**:



**Understanding about problem:**

* In this task I need to return first Equilibrium Point in an array.
* Equilibrium Point in an array is a position such that the sum of elements before it is equal to the sum of elements after it.

Note: If you can't understand the problem, feel free to contact us and we'll help you. Please don't copy and paste from anywhere.

ALL THE BEST

Team CP Club