DIU Take-Off Programming Contest

Fall 2021

[Mock Round]

Organized By

Problem Analysis

Platform Support

Judging Panel

Judges

Rahat Islam Srijon

Judging Director

9th Semester

Md. Galib Hossain

9th Semester

Albin Hossain

9th Semester

Mohammad Dipo Sultan

8th Semester

Md. Rana Hossain

7th Semester

Index

A. Heart of the Demon

Category: Giveaway

Problem Setter: Mohammad Dipo Sultan
Reviewer: Md. Galib Hossain
Analysis:

Just have to print "Pera Ase But Chill!" without quotation marks.

B. Worried at No-Worry

Category: Basic Math

Problem Setter: Rahat Islam Srijon
Reviewer: Md. Albin Hossain
Analysis:

You are given k the amount to be paid and x the highest valued banknote. You have to use 1 taka banknotes and x taka banknotes only to match and pay k taka, and you have to do it efficiently. Efficiently means that the total number of banknotes have to be minimum.

The minimum banknotes will be used when we are taking as many X taka banknotes we can and taking 1 taka banknotes as less we can. In other words, maximizing the use of X taka banknotes and minimizing the use of 1 taka banknotes.

The maximum number of X taka banknotes that can be used is K/X.

The maximum number of 1 taka banknotes that can be used is K%X.

So, the total number of banknotes would be K/X+K%X. You have to print it as output.

C. Tanjiro's Floating Numbers

Category: if-else

Problem Setter: Md. Galib Hossain
Reviewer: Md. Rana Hossain
Analysis:

In this problem, one decimal number is given as an input. And if the number has any value after the decimal point except zero then we need to print exactly that number otherwise we need to print only the integer part of that number.

Like 12.00 does not have any value after decimal point then we need to print only 12, and again 12.34 here it has 34 after the decimal point so for it we need to print that number 12.34.

Now, the question is how can we check that there is any value or not after the decimal point.

If you see a decimal number carefully, there you will find two-parts, one integer and another decimal part. For previous examples 12.34 ⟶ 12 + .34. That means, the decimal part is greater than the integer part. If there is no value except zero then the decimal number and the integer part of it will be equal.

And it is our solution. We will just check the decimal number with its integer part if they are equal then we will print the integer part otherwise we will print the decimal number.

D. Three Sided Polygon

Category: if-else+Geo

Problem Setter: Abu Saleh
Reviewer: Mohammad Dipo Sultan
Analysis:

As we have to cut the rod into 3 int pieces such that the sum of the 3 pieces is equal to N, so for making an Equilateral triangle N must be divisible by 3 otherwise we can't cut the rod into 3 equal size. Why must N be divisible by 3 ? Cause we know in an Equilateral triangle all 3 sides are equal. So, N = (N/3 + N/3 + N/3).

Now look into the case of an Isosceles triangle, in an Isosceles triangle exactly two of the sides are equal. So it's easy to find that if N is odd we can cut it into 1, (n-1)/2 , (n-1)/2 where the sum of these 3 sides is equal to N. If N is even we can cut it into 2, (n-2)/2 , (n-2)/2 .

Let's come to the invalid test as N is greater than 3 there is only one 1 invalid test which is 4 , because to make a valid triangle the sum of the lengths of any two sides of a triangle has to be greater than the length of the third side. As there is only one way to cut a 4 length rod which is 1,1,2 . So we see that here the sum of the first two sides is not greater than the 3rd side , which doesn't fill the valid triangle condition.

E. Average Mark

Category: Loop

Problem Setter: Md. Rana Hossain
Reviewer: Rahat Islam Srijon
Analysis:

In this problem, you are given total N students pre-test exam marks and Q queries.In each query you are given a range of roll numbers and asked to calculate the average marks of the students whose roll number in this given range. As (1<=N,Q<=100000), we can not solve this problem using linear search because in this given range linear search complexity come to O(N*N), so we can not pass the time limit. In that case we can use the prefix sum technique to solve this problem.

We designate a prefix sum array prefix. First, because we're 1-indexing the array,set prefix[0]=0,then for indices k such that 1<=k<=n, define the prefix sum array as follows:

$$prefix[k] = \sum_{i=1}^{k} a[i]$$

Basically, what this means is that the element at index k of the prefix sum array stores the sum of all the elements in the original array from index 1 up to k. This can be calculated easily in O(N) by the following formula for each 1<=k<=n.

$$prefix[k] = prefix[k-1] + a[k]$$

Now, when we want to query for the sum of the elements of array a between indices L and R inclusive, we can use the following formula:

$$\sum_{i=L}^{R} a[i] = \sum_{i=1}^{R} a[i] - \sum_{i=1}^{L-1} a[i]$$

Using our definition of the elements in the prefix sum array, we have

$$\sum_{i=L}^{R} a[i] = \text{prefix}[R] - \text{prefix}[L-1]$$

Since we are only querying two elements in the prefix sum array, we can calculate subarray sums and average mark in O(1) per query, which is much better than the O(N) per query that we had before. Now, after an O(N) preprocessing to calculate the prefix sum array, each of the Q queries takes O(1) time. Thus, our total time complexity is O(N+Q) , which should now pass the time limit.

F. Crossing the Bridge

Category: Number Theory

Problem Setter: Md. Albin Hossain
Reviewer: Abu Saleh
Analysis:

Rule of product states that, if there are A ways of doing something and B ways of doing another thing, then there are A · B ways of performing both actions.

There are 2 ways of choosing 1 tile from 1 column. So there are 2 · 2 · 2 ·….. · 2 · 2 = 2^n ways of choosing tiles from n columns.