



DIU Take-Off Programming Contest

Fall 2019

[Main Round]

Organized By



Problem Analysis

Platform Support



Judging Panel

Chief Judge

Saiful Islam

Senior Lecturer

Department of CSE

Daffodil International University

Special Judge

Ahsan Shuvo

Senior Programmer

Judges

Nesar Ahammed

Judging Director

10th Semester

Suvrajit Karmakar

Judge

12th Semester

Shah Habibul Imran

Judge

6th Semester

Nazmus Sakib

Judge

6th Semester

Md. Erfanul Islam

Judge

6th Semester

Tanima Hossain

Judge

6th Semester

Umme Rukaya Suny

Judge

6th Semester

Farjana Akter

Judge

6th Semester

Index

Problem Name	Setter Name	Reviewer
A. I will be the one	Saiful Islam	-----
B. The Fantabulous Organizer	Aquibuzzaman Md. Sayem	Farjana Akter
C. Dour De!	Hafizur Rahman Arfin	Shah Habibul Imran
D. Last But not Least	Raihanur Rahman	Nesar Ahammed
E. Game of Networking!	Debashish Saha Pranta	Erfanul Islam Bhuiyan
F. Vai Solve Hoyna!!	Mehedi Hasan	Tanima Hossain
G. Dhopash!	Azharul Islam Tazib	Nesar Ahammed
H. She's gonna die anyway!	Muhaiminul Islam Jim	Nazmus Sakib
I. The auto keyword	Pranto Das	Nesar Ahammed
J. Legacy	Mahmud Sajjad Abeer	Tanima Hossain

A. I will be the one

Category: Giveaway

Problem Setter: Saiful Islam

Analysis:

Just copy and paste the given code and replace "Hello World!" with **"DJ NAJ, I will be the one"**.

B. The Fantabulous Organizer

Category: Basic Math

Problem Setter: Aquibuzzaman Md. Sayem

Reviewer: Farjana Akter

Alternate Solution Writer: Muhaiminul Islam Jim

Special Thanks: Nesar Ahammed

Analysis:

There are N computers. Each needs 4 minutes. So to set up N computers we'll need $4*N$ minutes.

Each person takes 2 minutes to finish his job. So, the number of people needed will be: total minutes required/2.

Mathematically: $(4*N)/2$

C. Dour De!

Category: If-Else

Problem Setter: Hafizur Rahman Arfin

Reviewer: Shah Habibul Imran

Alternate Solution Writer: Muhaiminul Islam Jim

Special Thanks: Nesar Ahammed

Analysis:

You have to read the rules carefully and maintain the order of conditions.

The rules are described in hours, but the input is in minutes.

We first need to convert the hours into minutes.

3 hours = $3 * 60 = 180$ minutes

2 hours = $2 * 60 = 120$ minutes

1 hour = $1 * 60 = 60$ minutes

Rules are:

If a runner finishes in 3 hours, he/she will get a "bronze" medal. If($n \leq 180$)

If a runner finishes in 2 hours, he/she will get a "silver" medal. If($n \leq 120$)

If a runner finishes in 1 hour, he/she will get a "gold" medal. If($n \leq 60$)

But you have to maintain the order of conditions, if you put $n \leq 180$ you will get wrong answer.

The solution is (pseudo code):

```
If( $n \leq 60$ ) > print("gold");
```

```
else If( $n \leq 120$ ) > print("silver");
```

```
else if( $n \leq 180$ ) > print("bronze");
```

D. Last But not Least

Category: Basic Geometry, Observation

Problem Setter: Raihanur Rahman

Reviewer: Nesar Ahammed

Alternate Solution Writer: Nesar Ahammed

Special Thanks: Nazmus Sakib (Story) , Fuad Hasan (Image)

Analysis:

Here one side of triangle share one side of rectangle. So, here one side of triangle is equal to one side of rectangle. We provide you the area of Rectangle but both side of rectangle. So we need to divide the area by two and get area of one rectangle.

Area , Side_Of_triangle

Side1_Of_rectangle = side_Of_triangle

Area /= 2

Side2_Of_rectangle = Area/side_Of_triangle [Area of Rectangle = length*weidth]

E. Game of Networking!

Category: Basic Loop, Observation

Problem Setter: Debashish Saha Pranta

Reviewer: Erfanul Islam Bhuiyan

Alternate Solution Writer: Mehedi Hasan

Special Thanks: Mahmud Sajjad Abeer

Analysis:

This is a very simple problem. The problem requires at least two available positions and at least one of them must hold a Zero at any position so that any consecutive position can be converted to Zero if needed, with the power of G.

Simply print "Solution Achhe Habibi!" if $N > 1$ and any of the characters is Zero otherwise print "Beriye jao, beyadob kothakar!"

F. Vai Solve Hoyna!!

Category: String

Problem Setter: Mehedi Hasan

Reviewer: Ummey Rukaya Suny

Alternate Solution Writer: Tanima Hossain

Special Thanks: Mahmud Sajjad Abeer, Debashish Saha Pranta and Nesar Ahammed

Analysis:

Let, s1 and s2 is the given string. then convert all the characters of both strings into lowercase by using the following function.

```
void toLower(char s[],int n){
    for(int i = 0; i < n; i++){
        if(s[i] >= 'A' && s[i] <= 'Z')
            s[i] += 32;
    }
}
```

Let , s1 and s2 is the converted (all uppercase character into lowercase) strings.
then check if s1 and s2 is equal or not by using strcmp function.

G. Dhopash!

Category: Observation , Basic Math

Problem Setter: Azharul Islam Tazib

Reviewer: Nesar Ahammed

Alternate Solution Writer: Nesar Ahammed

Analysis:

Let , 4 variables a,b,c,d is used to store the quantities of each parts.

where,

a = quantities of "Arms"

b = quantities of "Backs"

c = quantities of "Seats"

d = quantities of "Structure"

then , Let M is the number of sofa that can be made by using this given materials.

then , $M = \min [(a/2), (b/2), (c/2), d]$ (as it's required 2 unit of "Arms" and 2 unit of "Backs" and 2 unit of "Seats" and 1 unit of structure to build a SOFA for 2 person).

then you have to check if $M*2$ is greater or equal than given N.

H. She's gonna die anyway!

Category: Mathematics , Observation

Problem Setter: Muhaiminul Islam Jim

Reviewer: Nazmus Sakib

Alternate Solution Writer: Nazmus Sakib

Analysis:

To get the solution for $M > 1$, we have to take 2 of the given terms in consideration. First find the difference between two consecutive terms of the series. We can get the difference by using $dif = \text{abs}(\text{value2} - \text{value1}) / \text{abs}(\text{term2} - \text{term1})$. Now to get the Nth term of the series we can run a loop from term2 to N. And subtract(if $N < \text{term2}$) or add(if $N > \text{term1}$) dif in every step. That way we can get the solution.

For $m=1$, the observation is that a unique summation only exists when $N = K*2 - 1$. Also we can observe that for any difference between two consecutive terms, the summation of the first N numbers of the series will always be the same.

I. The auto keyword

Category: String Processing , Observation

Problem Setter: Pranto Das

Reviewer: Nesar Ahammed

Alternate Solution Writer: Raihanur Rahman

Special Thanks: Mahmud Sajjad Abeer

Analysis:

```
type[3] = {int, float, double};
For each line {
    scan dataType as a string;
    scan variableName as a string;

    if last char of variableName is ';' {
        remove last char of variableName;
        save dataType of this variable as an interger; // 0 for int, 1 for float, 2 for double
    }
    else{
        scan equal sign as a string or char;
        scan variable1 as a string;
        scan +/- operator as a char or string;
        scan variable2 as a string;

        if dataType is not auto {
            save dataType of this variable as an interger; // 0 for int, 1 for float, 2 for double
        }
        else{
            d1 = dataType of variable1;
            d2 = dataType of variable2;

            d = maximum of d1 and d2; // why maximum? see typecasting rules for c/c++.

            save d as the dataType of this variable.

            print the answer.
        }
    }
}
```

J. Legacy

Category: Stopper , Dynamic Programming , Graph

Problem Setter: Mahmud Sajjad Abeer

Reviewer: Tanim Hossain

Alternate Solution Writer: Tanim Hossain

Analysis:

At first glance it may seem there exists greedy solution but there isn't any fixed greedy solution for this problem.

This is a simple dynamic programming problem after sorting the points according to X_i axes. First sum up all Y_i as they're pretty useless and rest of the calculations can be done only by using X_i . Then there exists both $O(N*M*M)$ and $O(N*M)$ DP solution where $M = N + N$. Let the DP state be $dp(pos, id)$. One catch is that the optimal solution may exist in negative x axes and upto some observations you'll find that the optimal **solution lies in range $[-N/2, 3N/2]$** inclusive. Pad the range and then **pos** is the last position used, **id** is the current point's index. Now, write a loop **i** through the points after **pos** and return the **min of solve(i, id + 1) + abs(ar[id] - i)**. That's the $O(N*M*M)$ solution. You can easily reduce the internal loop if you want to make it faster.

There also exists a straight-forward bipartite graph matching problem.