BUBT Intra-University Programming Contest, Fall 2019-20 (Junior Division)

Organized By:

Dept. of CSE, BUBT

Contest Rules

* Solutions to problems submitted for judging are called runs. Each run is judged as accepted or rejected by the judge, and the team is notified of the results**. *Submitted codes should not contain team name and the file name should not have any white space.***

* Notification of accepted or rejected runs will be suspended at the last hour of the contest time to keep the final results secret.

* A contestant may submit a clarification request to judges only through the Codeto.Win clarification system. If the judges agree that an ambiguity or error exists, a clarification will be issued to all contestants. Judges may prefer not to answer a clarification at all in which case that particular clarification request will be marked as IGNORED in the Codeto.Win clarification page.

* Contestants are not to converse with anyone except members of their team and personnel designated by the organizing committee while seated at the team desk. **They cannot even talk with their team members when they are walking around the contest floor to have food or any other purpose.** Systems support staff or judges may advise contestants on system-related problems such as explaining system error messages.

* While the contest is scheduled for a particular time length (four/five hours), the contest director has the authority to alter the length of the contest in the event of unforeseen difficulties. If the contest duration is altered, every attempt will be made to notify contestants in a timely and uniform manner.

* **A team may be disqualified by the Contest Director** for any activity that jeopardizes the contest such as dislodging extension cords, unauthorized modification of contest materials, distracting behavior, browsing internet except **Codeto.Win** site or communicating with other teams. **The judges on the contest floor** will report to the **Judging Director** about distracting behavior of any team. **The judges can also recommend penalizing a team with additional penalty minutes for their distracting behavior.**

* Ten, Eleven or Twelve problems will be posed. So far as possible, problems will avoid dependence on detailed knowledge of a particular application area or particular contest language.

* Contestants will have foods available in their contest room during the contest. So they cannot leave the contest room during the contest without explicit permission from the judges. **The contestants are not allowed to communicate with any contestant (even contestants of his own team) when they are outside the contest arena.**

* Teams can bring up to 1**00 pages of printed materials** with them. They can also bring five additional books, but they are not allowed to bring calculators or any machine-readable devices like CD, DVD, Pen-drive, iPod, MP3/MP4 players, floppy disks, etc. **Mobile phones MUST be switched off at all times and stored inside a bag or any other place that is publicly non-visible during the entire contest time. Failure to adherence to this clause under any condition will very likely lead to strict disciplinary retaliation and possible disqualification.**

*** The decision of the judges is final.**

*** Teams should inform the volunteers/judges if they don't get a reply from the judges within 10 minutes of submission. Teams should also notify the volunteers if they cannot log in to the Codeto.Win system. These sorts of complain will not be entertained after the contest.**

Problem Setters

01. ***Md. Saifur Rahman****, Assistant Professor, Dept. of CSE*
Bangladesh University of Business and Technology (BUBT)

02. ***Md. Murad Al Wajed,*** *Intake 26, Dept. of CSE*
Bangladesh University of Business and Technology (BUBT)
Software Engineer, Tapstar Apps

03. ***Abu Quwsar Ohi,*** *Intake 33, Dept. of CSE*
Bangladesh University of Business and Technology (BUBT)

04. ***Nazmul Hoda,*** *Intake 33, Dept. of CSE*
Bangladesh University of Business and Technology (BUBT)

05. ***Nazmul Alam Nayan,*** *Intake 35, Dept. of CSE*
Bangladesh University of Business and Technology (BUBT)

06. ***Md. Elias Hassan Naim,*** *Intake 32, Dept. of CSE*
Bangladesh University of Business and Technology (BUBT)

07. ***Maruf Ahmed Rahad,*** *Intake 35, Dept. of CSE*
Bangladesh University of Business and Technology (BUBT)

08. ***Khondokar Tanbir Ahmed,*** *Intake 35, Dept. of CSE*
Bangladesh University of Business and Technology (BUBT)

09. ***M M Mehedi Hasan,*** *Intake 33, Dept. of CSE*
Bangladesh University of Business and Technology (BUBT)

10. ***Walid Khan Jim,*** *Intake 35, Dept. of CSE*
Bangladesh University of Business and Technology (BUBT)

11. ***Md. Farid Miah,*** *Intake 32, Dept. of CSE*
Bangladesh University of Business and Technology (BUBT)

12. ***Md. Mimsad Ahmed Hridoy,*** *Intake 33, Dept. of CSE*
Bangladesh University of Business and Technology (BUBT)

13. ***Monzurul Islam,*** *Intake 35, Dept. of CSE*
Bangladesh University of Business and Technology (BUBT)

Judge Panel Members

1. ***Md. Saifur Rahman,*** *Assistant Professor, Dept. of CSE*
   Bangladesh University of Business and Technology (BUBT)

2. ***Md. Ashraful Islam****, Lecturer, Dept. of CSE*
   Bangladesh University of Business and Technology (BUBT)

3. ***Md. Elias Hassan Naim,*** *Intake 32, Dept. of CSE*
   Bangladesh University of Business and Technology (BUBT)

4. ***Md. Farid Miah,*** *Intake 32, Dept. of CSE*
   Bangladesh University of Business and Technology (BUBT)

5. ***Nazmul Alam Nayan,*** *Intake 35, Dept. of CSE*
   Bangladesh University of Business and Technology (BUBT)

6. ***Abu Quwsar Ohi,*** *Intake 33, Dept. of CSE*
   Bangladesh University of Business and Technology (BUBT)

7. ***M M Mehedi Hasan,*** *Intake 33, Dept. of CSE*
   Bangladesh University of Business and Technology (BUBT)

8. ***Maruf Ahmed Rahad,*** *Intake 35, Dept. of CSE*
   Bangladesh University of Business and Technology (BUBT)

A. Welcome to BUBT IUPC

Input: Standard Input, **Output:** Standard Output
Time Limit: 1 second(s)
Memory Limit: 256 megabytes

Problem Statement:

There is no statement for this problem. Just copy the code and submit it.

```
#include <stdio.h>
int main() {
        printf("Welcome to BUBT IUPC\n");
        return 0;
}
```

Input:

There is no input.

Output:

Print “Welcome to BUBT IUPC” in a single line.

Sample Input/Output:

Sample Input	Sample Output
	Welcome to BUBT IUPC

B. Homework

Input: Standard Input, **Output:** Standard Output
Time Limit: 1 second(s)
Memory Limit: 256 megabytes

Problem Statement:

Emu is a student of Deptartment of CSE, BUBT. Her course teacher gave her homework. Her teacher told her to write a program that takes two integer number inputs **X** and **Y**. Then her teacher told her to apply a condition, that is, if **Y** is less than **X** print "**Hello,BUBTian!**" (Without Quotes) otherwise, print **"Nothing"** (Without Quotes).

Input:

Input Two integer numbers **X** and **Y.** $(0 \leq \mathbf{x}, \mathbf{y} \leq 10^5)$.

Output:

The output is in a new line containing the answer "**Hello,BUBTian!**" or **"Nothing"**.

Sample Input/Output:

Sample 1:

Sample Input	Sample Output
5 6	Nothing

Sample 2:

Sample Input	Sample Output
7 3	Hello,BUBTian!

C. Easy??

Input: Standard Input, **Output:** Standard Output
Time Limit: 1 second(s)
Memory Limit: 256 megabytes

Problem Statement:

Given an integer **X**, you have to find the smallest number that is divisible by **3** and greater than **X**.

Input:

The first line of the input contains an integer **T** denoting the number of test cases. Each of the next **T** lines contains an integer, the value of **X**. $(0 \leq \boldsymbol{X} \leq 10^6)$.

Output:

For each input print the desired output on a single line.

Sample Input/Output:

Sample Input	Sample Output
3 2 4 9	3 6 12

D. The Gourmandizer Initiative!

Input: Standard Input, **Output:** Standard Output
Time Limit: 1 second(s)
Memory Limit: 256 megabytes

Problem Statement:

The majority of students studying in Dhaka come from different parts of the country. Most of them share flats together to save money. Recently your friend, Gony Stark started living in such a shared flat along with 4 other students named Prodip Parker, Komol Barton, Bishal Banner, and Shontu Rogers. But he is facing a very sensitive problem, whenever their maid, Rumanaf finishes cooking, his other four flatmates quickly eats the majority of the food and leaves barely enough for your helpless friend Gony. And can you blame them? After a long and hard day at the university, all of them become incredibly hungry! After a couple of days, Gony couldn't take it anymore so he came to you for help. After a lot of thinking you understood that the only way to ensure your friend Gony gets enough food is if they cook for an extra person! That way even if Gony's flatmates eat more, he surely will get his fair share. Before explaining this cooking arrangement to Rumanaf, you need to calculate how many "units" of food they will have to buy every day.

Input:

The input will only have one integer **A** (0<**A**<1000) representing the number of units of food bought for one person.

Output:

You need to print only a single integer which is the unit of foods Gony will buy such that there is enough food for him. Also, don't forget the trailing newline after the integer!

Sample Input/Output:

Sample Input	Sample Output
6	36

Problem Setter: Nazmul Alam Nayan, Intake 35, Dept. of CSE.

E. Three Consecutive Numbers

Input: Standard Input, **Output:** Standard Output
Time Limit: 1 second(s)
Memory Limit: 256 megabytes

Problem Statement:

Given a number **N**. You have to find out three integers **X**, **Y**, and **Z** where,

1. **X = Y – 1**,
2. **Z = Y + 1**, and
3. **X + Y + Z = N**.

Input:

Every input contains a positive number **N** $(6 \leq \mathbf{N} \leq 1000)$.

Output:

If no valid solution exists, print **-1**. Otherwise, print **X**, **Y**, and **Z** in a line and print a space between every two consecutive integers. Check sample input/output for clarity.

Sample Input/Output:

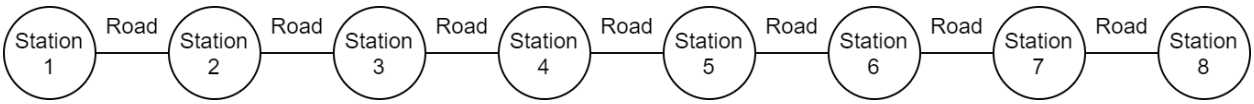
Sample Input	Sample Output
6	1 2 3

F. Selecting Guard (I)

Input: Standard Input, **Output:** Standard Output
Time Limit: 1 second(s)
Memory Limit: 256 megabytes

Problem Statement:

BUBT is a famous university. It's CSE department is very advanced and they produce good programmers nowadays. They arrange a programming contest every semester. But this time the programmers don't want to participate in the contest by risking their lives. That's why Dipu sir set some students to guard the linear street (which is in front of BUBT) for the programmers' safety. There are lots of stations in the street.



The students have to guard all the stations and roads between stations. Students can be placed on stations and they guard the roads and stations adjacent to it. But the students are not well behaved. If a road is guarded by multiple students they start fighting. So Dipu sir asked Zabir to write a program that finds the minimum number of students to guard all the stations and roads. Since Zabir is not a good programmer he needs your help.

In the above figure, if you place a student in station 4, he will guard the two roads connected to station 4 and station 3 and 5 as well.

Input:

The first line of the input is a single integer **T** (**T**<=100) which denotes the number of test cases. Then the next **T** lines contain a single integer **N** (**N**<=100) number of stations of the road.

Output:

For each test case output in a single line, the minimum number of students needed to guard all the stations and roads.

Sample Input/Output:

Sample Input	Sample Output
2 2 8	1 4

Problem Setter: Maruf Ahmed Rahad, Intake 35, Dept. of CSE.

G. Emu's Favorite

Input: Standard Input, **Output:** Standard Output
Time Limit: 1 second(s)
Memory Limit: 256 megabytes

Problem Statement:

You are given an integer input **N** and you have to find whether it is Emu's favorite or not. If it is Emu's favorite then print "**YES**" else print "**NO**".

A number is Emu's favorite if the summation of one or more 7 or -7 equal to **N**.
Example: n = 14 so, 14 = 7+7
n = 21 so, 21 = 7+7+7
So the numbers are Emu's favorite.

Input:

The first line of input contains an integer **T** $(1 \leq \boldsymbol{T} \leq 100)$ denoting the number of test cases. The next **T** lines of input contain an integer **N** $(-10^9 \leq \boldsymbol{N} \leq 10^9)$ to be tested.

Output:

For each test case, the output is in a new line containing the answer **'YES'** or **'NO'** depending on whether the given number **N** is Emu's favorite or not.

Sample Input/Output:

Sample Input	Sample Output
2 567 389	YES NO

Problem Setter: Nazmul Hoda, Intake 33, Dept. of CSE.

H. K'th Element (I)

Input: Standard Input, **Output:** Standard Output
Time Limit: 2 second(s)
Memory Limit: 256 megabytes

Problem Statement:

You are given **N** sorted values and **Q** queries. Each query consists of three values, **l**, **r**, and **k**. Each query states that you have to find the k'th value from the given input starting from index **l** to index **r**. Note that the index of **N** sorted value starts from index 1.

Input:

The input starts with an integer **N** ($N \leq 10^5$), defining the number of values. The next line contains **N** sorted values ($0 \leq values \leq 10^9$). The next line contains an integer **Q** ($Q \leq 10^5$), defining the number of queries. The next **Q** lines contain three space-separated integers **l**, **r**, **k**, ($l \leq r$ and $k \leq r - l + 1$) defining that you have to find the k'th element from the input value that lies in index **l**, to **r**. By k'th value, it is meant that if you sort the values starting from index **l** to **r**, then you have to output the k'th value from the sorted values.

Output:

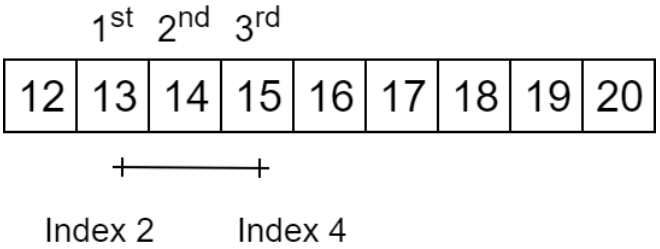
For each query, output the k'th value in a separated line.

Sample Input/Output:

Sample Input	Sample Output
9 12 13 14 15 16 17 18 19 20 3 1 5 4 2 4 2 4 9 6	15 14 20

Explanation:

The second query asks you to find the 2nd value from the **N** input values, starting from the 2nd index value of the input to the 4th index value. The sorted values on this index range are 13, 14 15. The 2nd value is 14, which is the answer to the second query.



I. Easy Balance

Input: Standard Input, **Output:** Standard Output
Time Limit: 1 second(s)
Memory Limit: 256 megabytes

Problem Statement:

You have **N** $(0 < \mathbf{N} < 101)$ integers, $\boldsymbol{a_1, a_2, a_3, \ldots, a_N}$.
Your objective is to make **N** equal integers by transforming some of them.
You may transform each integer at most once. Transforming an integer **x** into another integer **y** costs you $(\boldsymbol{x - y})^{\mathbf{2}}$ taka.

Find the **minimum total cost** to achieve your objective.

Input:

The first line of the input contains an integer number **N**.
The second line contains **N** space-separated integers $\boldsymbol{a_1, a_2, a_3, \ldots, a_N}$ $(-101 < \boldsymbol{a_i} < 101)$.

Output:

Print the minimum total cost to achieve the objective.

Sample Input/Output:

Sample 1:

Sample Input	Sample Output
2 4 8	8

Sample 2:

Sample Input	Sample Output
3 1 1 3	3

Sample 3:

Sample Input	Sample Output
3 4 2 5	5

Sample 4:

Sample Input	Sample Output
4 -100 -100 -100 -100	0

J. My Younger Sister

Input: Standard Input, **Output:** Standard Output
Time Limit: 2 second(s)
Memory Limit: 256 megabytes

Problem Statement:

Do you like chocolates? Maybe, or not, but my younger sister likes chocolates so much and she wants to buy more chocolates. Every day she comes to me with a list of chocolates of **N** types from **1** to **N**, where the price is given of every type of chocolates. Finally, she askes me what amount of money she will need if she wants to buy chocolates from **L** to **R** positions?

I heard that you are a good programmer of BUBT. So, you should write a program, which gives the total price of chocolates from **L** to **R** positions.

Input:

In the first line, you are given one integer **N** ($1 \leq N \leq 10^5$) denoting the total number of different chocolates. The second line contains **N** number of integers like as, $P_1, P_2, P_3, \dots, P_n$. Here P_i ($1 \leq P_i \leq 10^5$) denotes the price of i^{th} chocolate.

In the third line, you are given one integer **T** ($1 \leq T \leq 10^5$), which is the number of test cases. In each test case, you are given two integers **L** and **R** ($1 \leq L \leq R \leq N$).

Output:

You should write a program, which gives the total price of chocolates from position **L** to **R** for each test case. Print the output of each test case in a separate line.

Sample Input/Output:

Sample Input	Sample Output
5 2 5 7 100 500 2 1 3 2 5	14 612

Problem Setter: M M Mehedi Hasan, Intake 33, Dept. of CSE.

K. Century!

Input: Standard Input, **Output:** Standard Output
Time Limit: 1 second(s)
Memory Limit: 256 megabytes

Problem Statement:

Hey! Do you know?
What century are we now? The answer is the 21st century!
Do you know how to calculate it? If you don't, then that's not a problem at all!
Okay, let's explain it.

The 1st century was **1-100** years.
The 2nd century was **101-200** years.
The 21st century is **2001-2100** years and so on.

Your task is easy. Given a **year,** you have to calculate the number of the **century** the year is in.
For example, if a given year is, 100 then answer will be 1.

For more details, see input and output.

Input:

The first line contains integer **T**, defining the number of the test case. The following **T** lines each contain a number **Y** ($\mathbf{1 \le Y \le 10^{100}}$). Here **Y** denotes the year number.

Output:

For each test case, output one line containing Case number and the number of the century the year is in. Check the Input/Output section for further clarification.

Sample Input/Output:

Sample Input	Sample Output
2 100 200	Case 1: 1 Case 2: 2

L. Sorting a Binary Array

Input: Standard Input, **Output:** Standard Output
Time Limit: 1 second(s)
Memory Limit: 256 megabytes

Problem Statement:

Given a binary array that means each element of that array is either 0 or 1. Your task is to sort the array using minimum swaps. You are allowed to swap only adjacent elements.

Input:

The input contains multiple test cases. First-line contains an integer **T** $(1 \leq \mathbf{T} \leq 10)$. The first line of every test case contains an integer **N** $(1 \leq \mathbf{N} \leq 2 * 10^5)$. Then the next line contains **N** elements of that array.

Output:

For each test case, output the minimum number of swaps required to sort the binary array.

Sample Input/Output:

Sample Input	Sample Output
2 5 0 1 0 1 0 8 0 0 1 0 1 0 1 1	3 3

M. Rectangle Counting

Input: Standard Input, **Output:** Standard Output
Time Limit: 1 second(s)
Memory Limit: 256 megabytes

Problem Statement:

In the initial version or v1.0.0, you are given an **N*M** grid. Then count the number of rectangles in this grid by using the simple equation $\frac{(M)(M+1)(N)(N+1)}{4}$. In the next version v1.1.0, you are given **N** points then calculate how many rectangles are formed by these points. It is guaranteed that all rectangles are parallel to the axis.

Example: (0,0), (0,1), (1,1), (1,0), (2,0) and (2,1) number of rectangles are 3. Left rectangle (0,0), (0,1), (1,1), (1,0) right rectangle (1,0), (1,1), (2,1), (2,0) and big rectangle (0,0), (0,1), (2,1), (2,0).

Input:

The first line contains one integer N ($1 \leq N \leq 1000$) - the number of points. Then next N lines contain X and Y ($-10^9 \leq X, Y \leq 10^9$)- coordinates of the X and Y-axis.

Output:

Calculate the number of rectangles is forme d/shaped by these **N** points.

Sample Input/Output:

Sample 1:

Sample Input	Sample Output
6 0 0 0 1 1 1 1 0 2 0 2 1	3

Sample 2:

Sample Input	Sample Output
4 0 0 0 1 1 0 1 2	0

Problem Setter: Md. Farid Miah, Intake 32, Dept. of CSE.