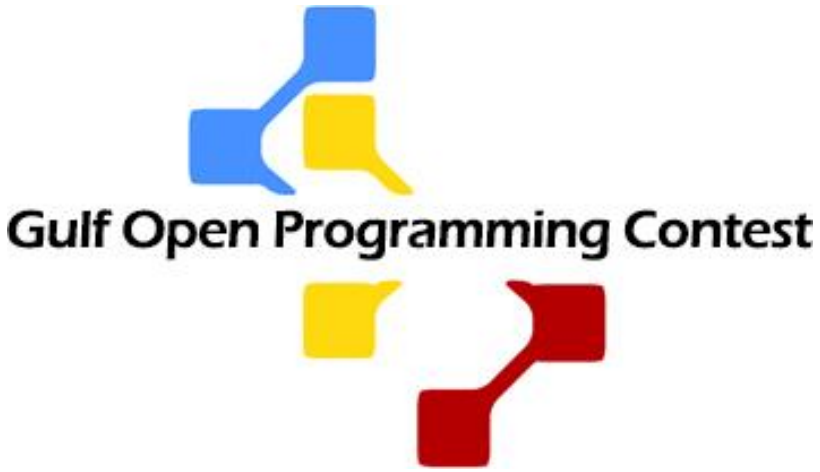

2017 Gulf Open Programming Contest

GOPC.IO

25th February, 2017

The problem set is made of 13 numbered pages.

[A] Delivery Service

Program:	`delivery.(cpp|java)`
Input:	`delivery.in`
Balloon Color:	`Red`

Description

You opened your new restaurant. The food is great, you have a lot of customers, and, most importantly, you can finally quit your job to enjoy your life. Everything is great! Only problem is that your delivery guy is overloaded with orders.

Unfortunately, you only have one bike that have space for one order. So, your delivery guy needs to do a round-trip for each order. Given the one-way trip time for each order, find the time needed to deliver all orders and return back to the restaurant.

Input Format

The input starts with a number ***T*** that represents the number of test cases in the file. Each test case starts with a line that contains one integer ***n*** the number of orders ($1 \leq n \leq 10{,}000$). The next ***n*** lines contain one integer t_n the one-way trip time needed ($1 \leq t_n \leq 100$).

Output Format

The output for each test case is in this form:

```
k. m
```

where ***k*** represents the test case number (starting at 1), ***m*** is the total time needed for the delivery.

Sample Input / Output

delivery.in

```
2
3
10
20
30
2
50
100
```

OUTPUT

```
1. 120
2. 300
```

[B] The Rotating Digits

Program:	`rotate.(cpp|java)`
Input:	`rotate.in`
Balloon Color:	`Yellow`

Description

A single rotation of digits in an integer number would remove the last digit and place it as the first digit, while shifting all other digits one position to the right. For example, if the integer is 12345, a single rotation would make the integer 51234, and two rotations would make it 45123.

Given an integer number consisting of exactly three digits, and a number of rotations, your task is to create the integer formed by the number of rotations given.

Input Format

The input starts with a number T ($1 \leq T \leq 1000$) that represents the number of test cases in the file. Each test case consists of a line that contains two integer, the initial number N ($100 \leq N \leq 999$), and the number of rotations R ($0 \leq R \leq 100$).

Output Format

The output for each test case is in this form:

```
k. ans
```

where ***k*** represents the test case number (starting at 1), and ***ans*** is the integer number formed after the rotations. Note that the digit 0 needs to be maintained when rotating, for example, rotating the integer 100 once yields a number that is represented as 010.

Sample Input / Output

rotate.in

```
2
123 1
489 2
```

OUTPUT

```
1. 312
2. 894
```

[C] Comma-Separated Values

Program:	csv.(cpp|java)
Input:	csv.in
Balloon Color:	Blue

Description

You work in a company where your job is to analyze and make sense of a lot of information. This information is usually sent to you as a spreadsheet saved in a text file. Basically, the data are sent in rows, and each row contains multiple fields which are separated with a special character. The data you receive from your colleagues usually contains multiple special characters, and you know they are far too lazy to stick to one standard.

One solution to this problem is to convert the data into comma-separated values (CSV). This standard is more practical and most spreadsheet applications can interpret it. So, your job is to convert the data you receive into CSV files. It would also be very helpful if you can append the number of fields at the end of each row. Look at the example below for a row sent to you:

```
Some#important#data
```

As you can see, the '#' character has been used as the separator. Your objective is to convert such a row into this (don't forget the number of fields at the end):

```
Some,important,data,3
```

Input Format

The input starts with a number T that represents the number of rows in the file. Each row contains a string with a maximum of 1,000 characters. The string will contain alphabets and those special characters:

`'#', '$', '|', '%'` (The hash, dollar, vertical bar, and percentage characters)

The special characters listed above can be found anywhere in the row. There are no spaces in the row.

Output Format

The output for each row should be the same row with the special characters listed above replaced with a comma. At the end of each row, append an extra field with the number of fields in it.

Sample Input / Output

csv.in

```
2
Some#important#data
Gulf$Open|Programming%Contest
```

OUTPUT

```
Some,important,data,3
Gulf,Open,Programming,Contest,4
```

[D] The Communication Tower

Program:	comm.(cpp|java)
Input:	comm.in
Balloon Color:	White

Description

The military is set to build new bases in an uninhibited area. They are also planning to build a communication tower that would facilitate the communication between these bases.

Now the army have selected the exact points where the bases will be built. However, they haven't decided yet on where to build the communication tower. It has to be built inside one of the bases, but choosing the right base can be a tricky task. The communication tower requires a lot of power to operate, and the power is mainly based on the distance covered.

The most important criteria is to have the tower in the base where the summation of all distances between it and the rest of the bases is minimum. Can you help the army with figuring out the minimum collective distances for any base?

In order to calculate the distance between two points, use the below formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Input Format

The input starts with a number T that represents the number of test cases in the file. Each test case starts with a line that contains one integer, the number of bases ($2 \le n \le 1{,}000$). The next line contains ***2n*** integers representing the points (x, y) where the bases will be built ($0 \le x, y \le 1{,}000$).

Output Format

The output for each test case is in this form:

```
k. ans
```

where ***k*** represents the test case number (starting at 1), and ***ans*** is the minimum collective distances for any base. The answer should be rounded to 2 decimal points. Use 'double' in your calculations.

Sample Input / Output

comm.in

```
2
3
1 1 1 2 1 3
4
2 4 5 9 8 0 4 2
```

OUTPUT

```
1. 2.00
2. 14.37
```

[E] Land Investments

Program:	land.(cpp|java)
Input:	land.in
Balloon Color:	Green

Description

You have decided to invest in buying lands. You see the long-term value of the investment and you plan to buy as much land as you can today and sell them in the future. Your cash is currently limited, and therefore you are looking for the cheapest lands to buy, while making sure the value of the purchase is maximized.

In this problem, you are given a list of lands to buy. For each, you are given the area of the land and its price. Your task is to find the best price to land area ratio.

Input Format

The input starts with a number T ($1 \leq T \leq 1{,}000$) that represents the number of test cases in the file. Each test case starts with a line that contains one integer N ($1 \leq N \leq 1{,}000$) that represents the number of lands. N lines follow, with each line containing two integers, Ai ($1 \leq Ai \leq 10^6$) representing the area of land i, and Pi ($1 \leq Pi \leq 10^9$) representing the price of land i.

Output Format

The output for each test case is in this form:

```
k. ansA ansP
```

where ***k*** represents the test case number (starting at 1), and ***ansA*** and ***ansP*** are the area and price of the best land as described above. If multiple lands have the same lowest price to land area ratio, print the one with the lower price.

Sample Input / Output

land.in

```
2
2
1000 1000000
1000 10000000
3
1000 1000000
1000 1000000
100 100000
```

OUTPUT

```
1. 1000 1000000
2. 100 100000
```

[F] The Drunken Sailor

Program:	`sailor.(cpp|java)`
Input:	`sailor.in`
Balloon Color:	`Brown`

Description

A drunken sailor is walking on a narrow pier. He's only taking steps to the right or to the left across the pier. If he takes too many steps in either direction, he might end up falling off. Given the initial position of the sailor and the series of steps and their directions that he takes, your task is to figure out if he falls off.

Input Format

The input starts with a number T ($1 \leq T \leq 1{,}000$) that represents the number of test cases in the file. Each test case starts with a line that contains three integers, the length of the pier P ($1 \leq P \leq 10^7$), the initial position of the sailor L ($1 \leq L \leq P$), and the number of series of steps S ($0 \leq S \leq 1{,}000$) the sailor will take. S lines follow, with each line containing an integer Si ($0 \leq Si \leq 10^7$) representing the number of steps, followed by a single character being either 'L' for left or 'R' for right. Position 1 on the pier is the left most position, and each step moves the sailor into an adjacent position. The sailor falls off if he takes a step to the left when he's in position 1, or takes a step to the right when he's in position P.

Output Format

The output for each test case is in this form:

```
k. ans
```

where ***k*** represents the test case number (starting at 1), and ***ans*** is the final position of the sailor after taking the series of steps. If the sailor falls off, ***ans*** is "Sailor Falls!" without the quotes instead.

Sample Input / Output

sailor.in

```
2
2 1 1
1 R
4 2 2
1 R
4 L
```

OUTPUT

```
1. 2
2. Sailor Falls!
```

[G] Valet Parking

Program:	`car.(cpp|java)`
Input:	`car.in`
Balloon Color:	`Purple`

Description

Your friend Ahmed got a new part-time job as a valet parker in one of the biggest hotels in your town. It comes with a lot of benefits as he will be able to drive a lot of. After a couple of months, Ahmed noticed something that he believes would make his job more efficient.

At the end of the day, after the night's rush at the hotel, Ahmed noticed that the cars are scattered across the parking. He thought that if the cars are all lined up with no empty spaces in between, it would be more efficient for him to retrieve cars in the morning. If he is going to do this, he needs to know the minimum number of car moves he needs to do in order to line up all the cars in the parking without having any empty spaces between them. The parking spaces are numbered from 1 to n. It doesn't matter where the lineup would start or end.

For example, if a parking lot has 8 spots and there are 3 cars located at spot# 2, 4, and 7, then the minimum number of moves in this case would be 1 (moving the car from spot 7 to spot 3.) Can you write a program to help your friend achieve this task?

Input Format

The input starts with a number T that represents the number of test cases in the file. Each test case starts with a line that contains two integers, the total number of car spots in the parking lot ($1 \leq n \leq 1{,}000{,}000$) and the number of cars ($1 \leq c \leq 1{,}000$). The next line contains ***c*** integers representing the spots each car occupies. It is guaranteed that the number of cars will be less or equal to the number of parking spots. No two cars will occupy the same spot.

Output Format

The output for each test case is in this form:

```
k. ans
```

where ***k*** represents the test case number (starting at 1), and ***ans*** is the minimum number of car moves required to lineup all the cars.

Sample Input / Output

car.in

```
2
8 3
2 4 7
10 4
3 4 6 8
```

OUTPUT

```
1. 1
2. 1
```

[H] The Formatting Dilemma

Program:	`format.(cpp|java)`
Input:	`format.in`
Balloon Color:	`Silver`

Description

It's the end of the semester when each student is happy to finally be finished with all the studying hardships of the semester's courses, but not before turning in their reports. You finally finished yours and about to submit them. But, you found out that you had a lot of formatting issues within your documents. As you don't want to invest a lot of time to format them manually, you decided to create a small program to do the job for you.

As you go through your reports, you find that the most common issues you have are the following:

- Capitalizing the starting word of the sentence (At the beginning of the line, or after a period, a question mark, or an exclamation mark).
- Capitalizing the first-person pronoun "I".
- Having a space before an opening bracket but not after (***[***, ***{*** and ***(***). There should be no space before the open bracket if it's the first character of the line.
- Having a space after a closing bracket but not before it (***]***, ***}*** and ***)***). There should be no space after the closing bracket if followed by a punctuation.
- There should be no space before punctuations (**.**, ***!***, ***?***, **,**, ***;*** and **'**).
- There should be space after some punctuations (**.**, ***!***, ***?***, **,**and ***;***).

Input Format

The input starts with a number ***T*** that represents the number of lines in the file. Each line is of maximum size of 10,000 characters and ends with a new line character.

Output Format

The output for each line should be the formatted line. Only the format rules mentioned in the problem description should be applied.

Sample Input / Output

format.in

```
2
Computer programming ( often shortened to programming ) is a process
that leads from an original formulation of a computing problem to
executable computer programs .
i love programming !looking forward to this year 's contest.
```

OUTPUT

```
Computer programming (often shortened to programming) is a process that
leads from an original formulation of a computing problem to executable
computer programs.
I love programming! Looking forward to this year's contest.
```

[I] The Conference Hall

Program:	`events.(cpp|java)`
Input:	`events.in`
Balloon Color:	`Black`

Description

In most hotels, the conference halls are usually busy throughout the day. Companies and individuals rent the halls for a certain amount of time to host an event or a meeting. The hotel that your friend Ahmed works in is considering changing the strategy of how they book the halls for customers.

In the past, they used to rent out the halls on a first-come-first-served basis. However, they think if they hosted more events per hall, even if they were shorter events, it would be more beneficial since this will give them better visibility and bigger market reach.

The booking requests are usually received beforehand with information about the start time and end time of the event. Your task is to help the hotel management figure out the maximum number of events they can host in a day per hall. Note that the events should not be clashing. The time will be presented in this question as integers.

For example, if the hotel receives 3 bookings to one of the halls, with the start and end time for the first booking being 3 and 10, the second booking timings being 6 and 11, and the third's being 10 and 15, then the maximum number of non-clashing events would be 2.

Input Format

The input starts with a number T that represents the number of test cases in the file. Each test case starts with a line that contains one integer, the number of bookings received by the hotel ($1 \leq n \leq 1{,}000$). The next line contains ***n*** pairs of integers representing the start and end time of each booking ($1 \leq b \leq 1{,}000$). The end time is always bigger than start time.

Output Format

The output for each test case is in this form:

```
k. ans
```

where ***k*** represents the test case number (starting at 1), and ***ans*** is the maximum number of non-clashing events that can be hosted.

Sample Input / Output

events.in

```
2
3
3 10 6 11 10 15
3
3 10 10 15 19 28
```

OUTPUT

```
1. 2
2. 3
```

[J] Redundant Storage

Program:	`storage.(cpp|java)`
Input:	`storage.in`
Balloon Color:	`Gold`

Description

Your company decided to install cameras around its facilities to protect them from unwanted visitors. These cameras record the footage 24/7 and shouldn't miss anything. To ensure a more robust disaster recovery plan, they decided to store the recordings on multiple separate servers.

Each camera will record on multiple servers (recordings can be the same or overlapping). A recording will be from an inclusive start time t_s to an inclusive end time t_e ($[t_s, t_e]$) given in seconds. As an IT administration operator (and more importantly a programmer), you did not feel that the solution implemented was the best possible and decided to verify it yourself.

You have decided to calculate the total amount of missing recordings for each camera (in seconds) in a specific day (total of 86,400 seconds in a day). A missing recording for a camera is a time interval in the day where there's no recording from the specific camera in any of the servers.

Input Format

The input starts with a number T that represents the number of test cases in the file. Each test case starts with a line that contains three integers; c the number of cameras ($1 \leq c \leq 1000$), s the number of servers ($1 \leq s \leq 10{,}000$) and r the number of recording ($1 \leq r \leq 1000$). The next r lines contain four integers; c_n the camera number ($1 \leq c_n \leq c$), s_n the server number ($1 \leq s_n \leq$ s), t_s the start time of record and t_e the end time of record ($1 \leq t_s \leq t_e \leq 86{,}400$),.

Output Format

The output for each test case is in this form:

```
k. m
```

where k represents the test case number (starting at 1), and m is the total number of missing recording for all cameras in seconds.

Sample Input / Output

storage.in

```
2
2 2 3
1 1 1 86400
2 1 1 40000
2 2 30000 86400
1 2 3
1 1 1 10000
1 1 1 10000
1 1 80000 86400
```

OUTPUT

```
1. 0
2. 69999
```