

12th IIUC Inter-University Programming Contest

I I U P C 2 0 1 5



Problem Set

*Allah grant me the serenity
to accept the problems that I cannot solve
the persistence to solve the problems that I can
and the wisdom to know the difference*

Organized by:
Faculty of Science and Engineering,
International Islamic University, Chittagong.

This 22 page document contains 10 problems

Table of contents:

Problem	Name	Page
A	Arrows	3
B	Burger Stand	5
C	Combination Lock	7
D	Discovering Paths	9
E	Enchanted Forest	10
F	Fisa Flood	12
G	Gift Box	15
H	Hippo Circus	17
I	Inception	19
J	Jerry's Game	21

A

Arrows

We like to do ASCII art which involves creating various symbols or pictures using the ASCII characters. Today we are going to draw arrows!

There are four types of arrows that can be drawn. Left single arrow, left double arrow, right single arrow and right double arrow.

A left single arrow is defined by a *left angle bracket* character ('<') immediately followed by zero or more consecutive hyphen characters ('-'). A left double arrow is a *left angle bracket* character ('<') immediately followed by zero or more consecutive equals characters('='). A right single arrow is zero or more hyphen characters ('-') immediately followed by a *right angle bracket* character ('>'). A right double arrow is zero or more equals characters('=') immediately followed by a *right angle bracket* character ('>').

For example, the following strings are arrows (quotes for clarity only): "==">", "<--", "<", "<===", "--->", ">". But "======" is not an arrow. The length of an arrow is measured by the number of characters it contains.

You will be given a String s . Calculate the length of the longest arrow in s , or -1 if it does not contain any arrows at all.

Input:

First line of the input is an integer T ($1 \leq T \leq 150$), the number of test cases. Each of the following T lines contain a string S , which is no longer than 50 characters and contains only the four different characters mentioned in the problem.

Output:

For each test case, print a line in the following format: "Case X : Y " (without quotes), where X is the test case number starting from 1 and Y is the length of the longest

B**Burger Stand**

You have been planning for a long time to open a roadside burger stand, but you failed to find a sweet spot for the little shop. In Bangladesh, it is fairly common to open small shops right by the side of roads and footpaths, causing troubles to everyone else passing by the roads. On a bright side, many people stop by and buy stuffs from these shops on their way. So you thought, why you don't do the same.

But being a computer science student, you have little more civil sense (surprisingly) than most others. So, instead of opening the shops at will, you have decided to maintain the following conditions when choosing a spot for your burger stand.

1. It is not directly in front of a private driveway.
2. It is not directly in front of a bus stop.
3. It is not 5 meters before a bus stop.
4. It is not 10 meters before a bus stop.
5. It is not directly in front of a side-street.
6. It is not 5 meters before a side-street.
7. It is not 5 meters after a side-street.

A road segment will be represented as a String, where each character describes a section of the road 5 meters in length. So the first character describes the first 5 meters of the road, the second character describes the next 5 meters and so on. We will use 'D' for driveway, 'B' for bus stop, 'S' for side-street and '-' for all other sections of the street. A position is directly in front of an object if it has the same index as the object in the road. A position is before an object if its index is lower than the index of the object in the road. Finally, a position is after an object if its index is higher than the index of the object in the road. Note, your shop only takes one position.

Given the road segment calculate the total number of possible spaces for your burger stand on that road.

Input:

First line will contain an integer T ($1 \leq T \leq 50$), the number of test cases. Each of the following T lines will contain a string S denoting a road segment as described above. There will be at most 50 characters in any line.

Output:

For each case produce a line of output “Case X : Y ”, where X is the test case number starting from 1 and Y is the number of possible spaces for your burger stand. Check sample input and output for details.

Example:

Sample Input	Sample Output
2 ---B--S-D--S-- DDBDDBDDBDD	Case 1: 4 Case 2: 0

Explanation:

For the first case, you can open the burger stand in position 1, 5, 10 and 14. On the second case, there is empty spot.

Problem Setter – Zobayer Hasan

Alternate Writer – Muhammad Ridowan

C

Combination Lock

We are in desperate need of expert hands – hands that can open combination locks in a flash. We already know the initial and target state of the lock, but we need someone who can turn the wheels using the smallest number of steps needed to open the locks fast.



A multi-wheel combination lock has multiple wheels that one can turn either clockwise or counter clockwise to pick a digit from 0 to 9. The digits wrap around, so you get 0 right after you cross 9 in a clockwise fashion. Similarly, you get 9 after you cross 0 in a counter clockwise fashion.

Given the number of wheels on a combination lock, its initial state and the target state, you need to find out the minimum number of steps needed to go from the initial state to the target state. For every change in a digit on a wheel, we will count one step. For example, for a 4 wheel combination lock if we are given the initial configuration “1234” and we want to get to “2546” we need at least 7 steps (1 step to go from 1 to 2, 3 steps to go from 2 to 5, 1 step to go from 3 to 4 and 2 steps to go from 4 to 6).

Input:

The first line of input will contain an integer T ($1 \leq T \leq 200$), the number of test cases. Each of the following T lines will give you n , *initial*, and *target*. Here n ($1 \leq$

I I U P C 2 0 1 5

$n \leq 100$) is an integer specifying the number of wheels in the combination lock. The numeric strings **initial** and **target** denotes the initial and target states of the lock. The length of both strings will be exactly n and they will only contain decimal digits.

Output:

For each line of input, print the case number as “Case X : Y ” where X is the case number (starting from 1) followed by Y , the minimum number of steps needed to go from the initial configuration to the target configuration. Check sample input and output for details.

Example:

Sample Input	Sample Output
2	Case 1: 7
4 1234 2546	Case 2: 1
4 1234 1235	

Problem Setter – Monirul Hasan

Alternate Writer – Zobayer Hasan

D

Discovering Paths

Given a grid with R rows and C columns, you are currently at $(0, 0)$ and you want to go to the position $(R-1, C-1)$. You have only two kind of movement allowed. From any position (i, j) you can go to either $(i+1, j)$ or $(i, j+1)$. You need to find the number of ways you can go to $(R-1, C-1)$ from $(0, 0)$. Easy, right? But here's is a slight problem. All the cells are not available all the time. So while counting the number of ways you need to consider that you can never step into a cell which is not available right now.

Input:

First line will contain an integer T ($1 \leq T \leq 10$), which is the number of test cases. Each case starts with a line R, C and Q . Here, $1 \leq R, C \leq 1000$ and $1 \leq Q \leq 10000$. Then, Q queries follow, each with four integers a, b, c, d . This means the cells inside the rectangle with lower left corner at (a, b) and upper right corner at (c, d) are not available. All the coordinates are given in row major order with 0-based indexing. The lowermost and leftmost point is considered to be $(0, 0)$.

Output:

For each case print a line "Case T", where T is the case number. For each query in a case, print 3 spaces and then "Query X: W", where X is query number and W is the number of ways possible for that particular query. Answer needs to be in modulo 912. Check sample input and output for details.

Example:

Sample Input	Sample Output
1 5 5 2 1 1 2 2 0 1 2 3	Case 1 Query 1: 10 Query 2: 5

Problem Setter – Hasnain Heickal

Alternate Writer – Muhammad Ridowan

E

Enchanted Forest

The forest of El-Dorado has been enchanted by dark magic. Rabbit wizards live in the forest but all the paths among different wizards have been blocked by the magic. Zao, the lord of wizard rabbits, wants to lift up the curse and clean up all the dark magic from the forest, so that all the rabbits can reach each other's houses again like they used to do in the past.

Therefore, he asked for your help. You will be given an adjacency matrix that represents whether there is a bidirectional path between two rabbits or not. Determine the number of different sets of paths that can be cleaned.

Input:

First line of the input is an integer T ($1 \leq T \leq 100$), where T is the number of test cases. For each test case, first you will be given an integer n ($1 \leq n \leq 15$), the number of rabbits, followed by n strings containing n characters each, where j th character of i th element is 'Y' if there's a path between the i th rabbit and the j th rabbit, or 'N' otherwise. All paths are bidirectional.

Output:

For each test case, print a line in the following format: "Case X : Y ", where X is the number of test case starting from 1 and Y is the number of different sets of paths that can be cleaned to achieve Zao's goal. Since the number can be very big, output it in modulo 10000.

Example:

Sample Input	Sample Output
4	Case 1: 4
3	Case 2: 1
NYY	Case 3: 38
YNY	Case 4: 0

YYN 4 NYNN YNY NYNN NYNN 4 NYYY YNY YYNY YYYN 2 NN NN	
--	--

Problem Setter – Muhammad Ridwan
Alternate Writer – Zobayer Hasan

F**Fisa Flood**

This is the story of a distant country called Udesc. No one in this world actually knows about it. Nobody knows its geographical position or any road that leads to this country. Nobody even knows the language of Udesc. They are truly separated from the world! As they are separated from the world, they don't know about modern technologies. But the people of Udesc are very brilliant. They already invented many of the technologies by themselves. For example, they can produce electricity and use electric machines. They have invented cars, aero planes, submarines and trains. They use sophisticated computers, digital cameras, mobile phones, televisions, air conditioners, refrigerators, energy saving lights, microwave ovens, washing machines and many more useful stuffs in their day to day life. They also use internet which is not like ours. They can browse and download any amount of data without any charge! Recently they are trying to grab the nuclear technology.

The people of Udesc are very honest and hardworking. They work hard from dawn to dusk. Nobody remains idle. Everybody keeps busy with his or her own work. The social security of the land is very high. Nobody dies for the lack of medical treatment. There are universities, colleges, schools, banks, parks, theaters, supermarkets, and many other amusement centers. There is no corruption or crime. All through the year, the cops and the judges pass their vacation time.

The only drawback of this country is, they have given all their efforts to mechanical improvements, and so, they forgot to improve their food habit. They have no fast food restaurant! They never took a sip of soft drinks or a bite from a crispy cheese burger. They even don't know the name of French fries. But things have been changing lately. Fisa, a cormorant citizen of Udesc, privately invented how to make soft drinks. And he has established 'Fisa Flood', the first ever factory in Udesc, which only produces soft drinks.

Fisa Flood started producing soft drinks of two flavors. The first one is named 'Fisadasa' which is white in color. The second one is named 'Fisaloca', which is dark

red. Each type is available in different sized cans i.e. 250ml, 500ml, 1 liter, 1.5 liters, 2 liters and 5 liters (really big ones).

The price of Fisadasa and Fisaloca are same. As the people of Udesc don't know about soft drinks, Biqar, the cunning CEO of Fisa Flood takes an innovative marketing approach. Fisa Flood started giving their consumers a promotional offer. If someone returns two empty cans of Fisa Flood product, he will get one new can in return. But things are not as easy as you may think. Here are some details of the promotional offer.

If you give two cans of Fisadasa, you will get one can of Fisadasa. But if you give two cans of Fisaloca, you will not get one can of Fisaloca. Instead, you will still get one can of Fisadasa. If you want to get one can of Fisaloca, you have to return one can of Fisadasa and one can of Fisaloca. One more thing, the promotional cans are specially labeled to separate from normal cans without offer. That means, if you submit two cans after the promotion period, still you will get one new specially labeled can.

Soon, Fisa Flood products become very popular among the people. They got used to it so much that, they hardly drink any water now. Nikas, a young boy, is very fond of this as well. As he does not have much money, and the offer will not last forever, he planned to buy as much cans as possible. So he gathers all his money and buys a lot of Fisadasa and Fisaloca. After that, Nikas starts to drink the cans one by one. Suddenly a thought strikes his mind. He becomes aware of how many cans in total he could drink. He knows, if he has two Fisadasa can, then he will get the third one by returning the first two. Thus he can drink three cans in total. If he has two Fisadasa and one Fisaloca, then he can drink five cans in total (after drinking two Fisadasa cans, gets a new Fisadasa can back. Now he has two new cans, which he can submit to get a new one, totaling five cans).

First, Nikas was only aware of the number of cans of each type he could get. But soon he realizes that, the number of each type may leads to various results depending on the order he drinks the cans. Now, he is wondering what the last can would be, will it be Fisadasa or Fisaloca? Nikas doesn't have time for this, and not really good with math, so he has asked you to find the probability of the last can being Fisadasa or Fisaloca.

Input:

The input starts with a line containing an integer T ($1 \leq T \leq 111$), which denotes the number of test cases. Each of the next T lines describes one test case containing two integers, A and B ($0 \leq A, B \leq 1000$), separated by a single space. A is the number of Fisadasa cans Nikas bought, and B is the number of Fisaloca.

Output:

For each case, you have to produce exactly one line of output. This line contains the test case number starting from 1 along with two floating point numbers, p and q , separated by a single space. Here, p denotes the probability that the last can will be Fisadasa, and q denotes the probability that the last can will be Fisaloca. You have print them rounding off to three decimal places after the point. Check sample input and output for details.

Example:

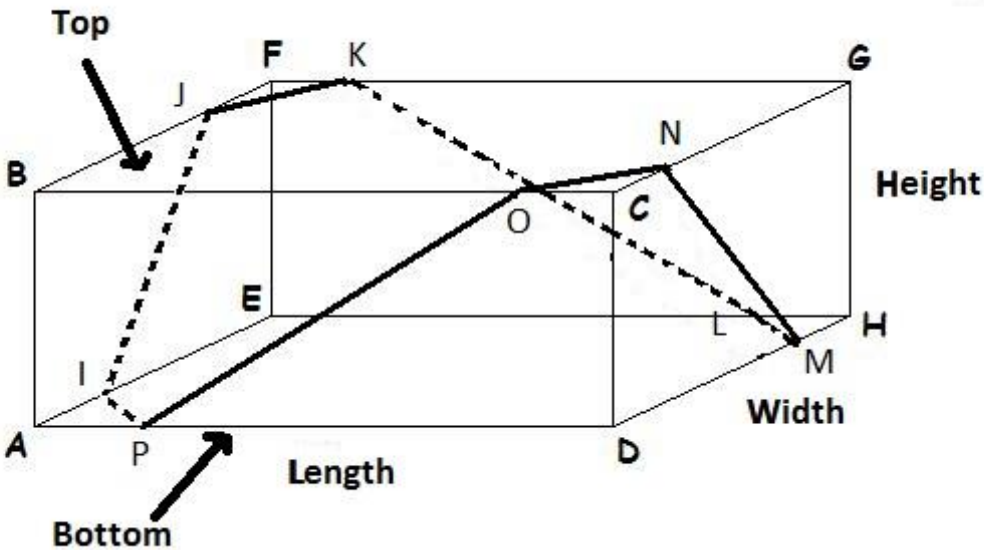
Sample Input	Sample Output
3	Case 1: 1.000 0.000
1 0	Case 2: 1.000 0.000
0 2	Case 3: 0.000 1.000
1 1	

Problem Setter – Asif Mahmood

Alternate Writer – Abdullah Al Raqibul Islam

G	Gift Box
---	----------

You want to give a special gift to your special friend on a special occasion in a very special box! So you have bought some special ribbons and you want to wrap the gift box in a special way. Instead of wrapping it in a traditional way, that is ribbons crossing through the middle of the box parallel to the edges, you are wrapping it in a way that the ribbon makes a complete loop around the box creating two parallel pieces of ribbons on the top and on the bottom of the box (see picture below). The ribbon crosses every face once, except the top and bottom, which it crosses twice. The ribbon rests tightly against the box all the way round because the angle at which it meets an edge is continued onto the next face.



In the above picture, the ribbon is running through I-J-K-L-M-N-O-P points. JK and ON are the parallel pieces on the top while IP and LM are the parallel pieces on the bottom.

If you are given the length, width and height of the box, can you find the length of the ribbon?

Input:

Input starts with an integer T ($1 \leq T \leq 1000$), the number of test cases. Each test case consists of three integers, L , W and H , denoting the length, width and height of the box respectively. You can assume that $1 \leq L, W, H \leq 1000$. Lengths are given in centimeters.

Output:

For each test case, first print the case number starting from 1 as shown in sample input and output section, then print the length of the ribbon in centimeters rounded off to 4 decimal places. Floating point error lower than $1e-3$ will be ignored by the judge. Check sample input and output for details.

Example:

Sample Input	Sample Output
2 60 30 15 20 10 5	Case 1: 174.9286 Case 2: 58.3095

Problem Setter – Zobayer Hasan

Alternate Writer – Asif Mahmood

H

Hippo Circus

You all know about my friend Hippo. Hippo and its other hippopotamus friends are starting a circus. They have been practicing a lot, and they are getting better at the show. I have seen their show several times and I personally think they are really good. So I encouraged them to show in public. And after a lot of arguing and convincing they finally agreed.



So are they are getting ready for their big showdown. Everything is prepared. The performers are working day and night to perfect everything. The tent is almost ready. In a word everything is having the final touch.

In the night before the show the hippopotamuses started to budget the time and encountered a big problem. They were planning for a big entrance where every hippopotamus will enter through the gate and take a bow to the audience. But this is taking too much time. So to shorten this they devised a plan – “One hippo will ride another one”.

The balances of the hippopotamuses are

not so good yet. So a hippo can take only another hippo over it, not more than that. There is another problem, if a hippo carries another hippo, it slows the speed of the hippo. So to help them with the problem they wish your help.

Given a door with height H , and N hippopotamuses with height h_i (height of the i^{th} hippo, $1 \leq i \leq N$), you need to find the minimum time so that every hippo can enter the door and bow. A hippo can only enter the door if its height is less than the height of the door. If a hippo is carrying another hippo, then the summation of their heights must be less than the door's height. A hippo while walking alone,

I I U P C 2 0 1 5

takes T_a time to enter the door and bow. A hippo while carrying another hippo, takes T_d time to enter and bow.

Input:

First line of input will contain an integer, C ($1 \leq C \leq 10$), the number of test cases. Then C cases will follow. First line of each case is four integers, N , H , T_a , and T_d . Next line contains N integers, the height of the hippopotamuses.

Here,

$1 \leq N \leq 100000$

$0 \leq T_a < T_d \leq 10000$

All the heights will be less than 100. Heights of all the hippopotamuses will be less than the H .

Output:

For each case output one line. "Case X : M " (without the quotes), where X is the case number starting from 1 and M is minimum time needed. Check sample input and output for details.

Example:

Sample Input	Sample Output
2	Case 1: 6
3 5 2 3	Case 2: 5
3 4 2	
3 6 2 3	
3 4 2	

Explanation:

In the first case, all the hippo walks alone that's why each take 2 seconds and in a total of 6 seconds. But in the second case, if the first hippo carries the third hippo ($3 + 2 < 6$) then it takes 3 seconds and the other hippo takes 2 seconds.

Problem Setter – Hasnain Heickal

Alternate Writer – Asif Mahmood

I

Inception

Dom Cobb and his partner Arthur perform illegal stuffs by entering the subconscious minds of their targets. They use two-level *dream within a dream* strategy to extract valuable information. Dreamers are awakened either by a sudden shock (a “kick”), or they may die in the dream.

Now Dom wants your help. Because he has a complex assignment. He has to go through a lot of people’s dreams. He travels from one person’s dream to another person’s dream. He is too consumed that he can no longer keep track whether he is awake or in someone else’s dream.

He will give you n queries to process, each of the queries will be in one of the following forms:

“*Sleep X*” – This means person named X will be sleeping and Dom is going into X ’s dream, from the previous person’s dream (if any).

“*Kick*” – This means the current person in whose dream Dom has entered, will be awoken, and Dom will return to the previous person’s dream from where he came to this dream. If Dom is not in anyone’s dream anymore, ignore this query.

“*Test*” – This means Dom wants to know whose dream he is in at the moment. You have to print the person’s name. If Dom is not in anyone’s dream at the moment, you have to print “Not in a dream” (without the quotes).

Input:

First line will contain an integer n ($1 \leq n \leq 10000$), the number of queries. Each of the following n lines will contain one of the three queries mentioned above. For the case of “*Sleep – X*” query, X will be a string composed of only uppercase or lowercase letters and no more than 15 characters long.

Output:

For each of the “Test” queries, print the name of the person whose dream Dom is in right now exactly as it appeared in the input. If Dom is in no one’s dream, output the line “Not in a dream”. Check sample input and output for details.

Example:

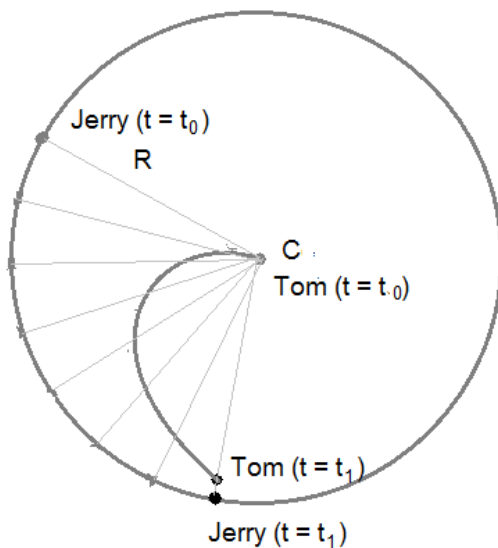
Sample Input	Sample Output
20	Sakin
Sleep Dom	Mushfiq
Sleep Sakin	Asif
Test	Shafi
Sleep Asif	Asif
Sleep Mushfiq	Sakin
Test	Dom
Kick	Not in a dream
Test	Not in a dream
Sleep Shafi	
Test	
Kick	
Test	
Kick	
Test	
Kick	
Test	
Kick	
Test	
Kick	
Test	
Kick	
Test	

Problem Setter – Muhammad Ridowan
 Alternate Writer – Abdullah Al Raqibul Islam

J

Jerry's Game

Tom and Jerry are very fond of cat and mice games, which might be rather obvious to you. Today they are playing a very complicated game. The goals are simple as usual though, Jerry would be running and Tom would have to catch Jerry.



However, today Jerry is running on a perfect circular path with radius R meters, at a constant speed of V m/s. Initially Tom is sitting at the very center of the circle. He wants to catch Jerry as soon as possible, but we all know, Tom is not very intelligent. Instead of calculating an optimal direction to catch Jerry, he is just running towards Jerry.

As Jerry is also moving, the path Tom has taken starts to look like a curve (see picture above). At any

given moment, Tom's position is between Jerry's current position and the center of the circle. Tom is also moving at a constant speed of V m/s, same speed as Jerry. Find the time (in seconds) Tom would need to catch Jerry.

Input:

Input file has T ($T \leq 10000$) test cases, each case consists of two integer R and V . Here, $0 < R, V \leq 10000$.

Output:

For each test case, print the case number and the time Tom will need to catch Jerry. Floating point rounding error lower than $1e-5$ will be ignored by the judge.

Example:

Sample Input	Sample Output
4	Case 1: 0.70685835
45 100	Case 2: 0.00507691
5 1547	Case 3: 0.15707963
1000 10000	Case 4: 1.62854830
5668 5467	

Problem Setter – Zobayer Hasan

Alternate Writer – Monirul Hasan

The End