



ACM-ICPC Thailand National Programming Contest 2010 (Online Version)

4th September 2010

Hosted by Prince of Songkla University, Phuket Campus

/*This set has been modified: One problem with statements in Thai language is removed and two problems are added*/

- There are **12** problems (A-L) to solve within **360** minutes (6 hours).
- Solve as many problems as you can, in an order of your choice.
- Use C or C++ or Java to program at your convenience for any problems.
- Input and output of each program are **standard input** and **output**.

Problem A:	Mouse and Cheese
Problem B:	Bisectors
Problem C:	Warp Speed
Problem D:	High-Precision Number
Problem E:	Configuration
Problem F:	Two Longest Paths
Problem G:	A Minimum Land Price
Problem H:	Hacker's Crackdown
Problem I:	Shuffle
Problem J:	Maximum GCD
Problem K:	Palindrome Again
Problem L:	L's in Circle



acm International Collegiate Programming Contest

IBM

event sponsor

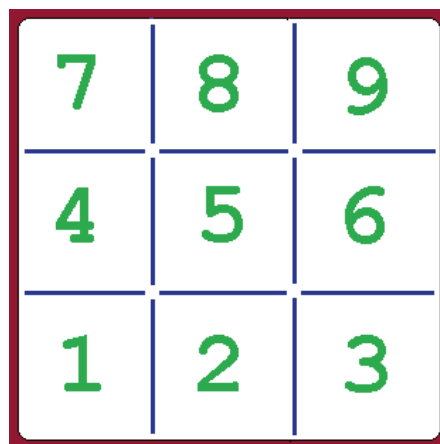


Problem A

Game – Mouse and Cheese

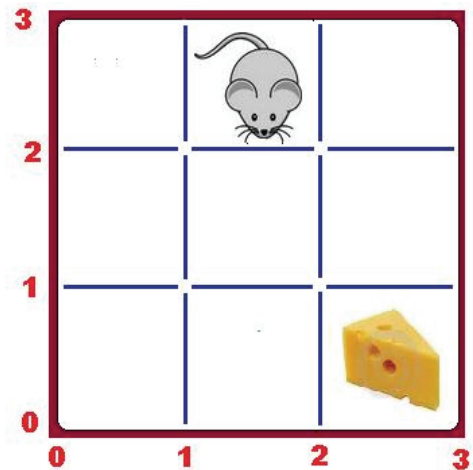
Input: Standard Input
Output: Standard Output

SOHA and TARA have recently invented a new game called “Mouse and Cheese”. As the name suggests, this game involves a mouse searching for a piece of cheese. The game is played on a 3x3 board as shown in the diagram below. Each cell is uniquely numbered with an integer between 1 and 9. The board contains 12 sticks (the blue lines in the diagram).



Initially a mouse and a piece of cheese are placed on two different cells. In the example to the right, the mouse is placed on ‘cell 8’ and the cheese is placed on ‘cell 3’.

The rule of the game goes like this: It’s a two-player game where the players make moves alternately. SOHA, being player 1, goes first. In each move, the player selects a stick and throws it away. After each move, if the mouse can reach the cell containing the cheese without touching any of the remaining sticks, then that player is declared as the winner? If both play perfectly, who wins?



Before the game starts, some of the sticks are removed. You will be given the coordinates of these sticks and the cell numbers of the mouse and cheese. A stick is represented using the coordinates of its end points. The lower left of the grid is the origin (0, 0). The two sticks surrounding the cheese, in the example above, has coordinates “2 0 2 1” and “2 1 3 1”. Note that the end-points of a stick can be given in any order. That is “2 0 2 1” could be given as “2 1 2 0”.

Input

The first line of input is an integer $T(T < 1000)$ that determines the number of test cases. Each case starts with 3 integers S , C and R . S and C denotes the location of the mouse and cheese respectively. Each of the next R lines contains 4 space-separated integers that give the coordinates of the sticks removed from the board before the start of the game.

Notes and Constraints

$1 \leq S, C \leq 9$

$S \neq C$

$0 \leq R \leq 12$

For each case, all the removed sticks will be distinct.

Output

For each case, output the case number first, starting with 1, followed by the name of the player who wins. If the mouse can already reach the cheese before the game starts, output "No Cheese!" instead. Look at the samples for exact format.

Sample Input	Output for Sample Input
3	Case 1: No Cheese!
1 2 1	Case 2: SOHA
1 1 1 0	Case 3: TARA
1 4 0	
7 2 2	
1 2 1 3	
2 2 2 3	

Problemsetter: Sohel Hafiz

Special Thanks: Arifuzzaman Arif

Problem B

Bisectors

Input: Standard Input
Output: Standard Output

We all probably know how to find equation of bisectors in Coordinate Geometry. If the equations of two lines are $a_i x + b_i y + c_i = 0$ and $a_j x + b_j y + c_j = 0$, then the equations of the bisectors

of the four angles they create are given by $\frac{a_i x + b_i y + c_i}{\sqrt{a_i^2 + b_i^2}} = \pm \frac{a_j x + b_j y + c_j}{\sqrt{a_j^2 + b_j^2}}$. Now one has to

be quite intelligent to find out for which angles to choose the '+'(plus) sign and for which angles to choose the '-'(minus) sign. You will have to do similar sort of choosing in this problem. Suppose there is a fixed point (C_x, C_y) and there are n ($n \leq 10000$) other points around it. No two points from these n points are collinear with (C_x, C_y) . If you connect all these point with (C_x, C_y) you will get a star-topology like image made of n lines. The equations of these n lines are also given and only these equations must be used when finding the equation of bisectors. This n lines create $n(n-1)/2$ acute or obtuse angles in total and so they have total $n(n-1)/2$ bisectors. You have to find out how many of these bisectors have equations formed using the + sign. The image below shows an image where $n=5$, $C_x=5$ and $C_y=2$. This image corresponds to the only sample input.

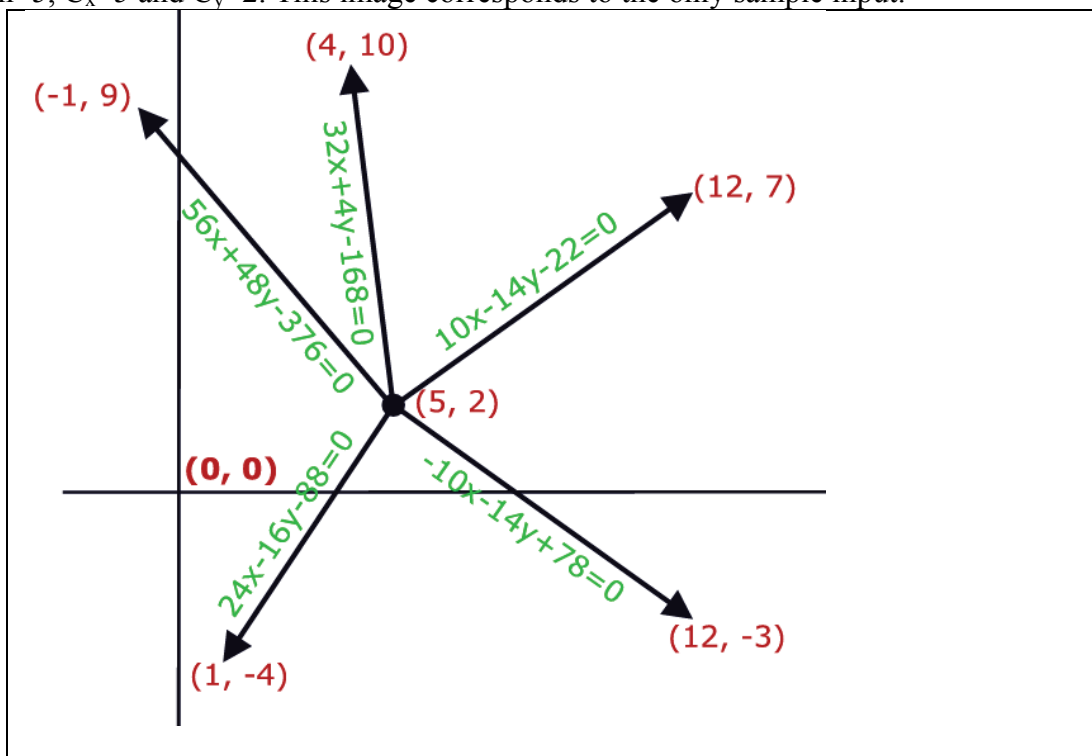


Figure: Five lines above create $5(5-1)/2=10$ angles and these angles has 10 bisectors. Of these 10 bisectors, the equation of only 4 are formed using the + sign

of the formula
$$\frac{a_i x + b_i y + c_i}{\sqrt{a_i^2 + b_i^2}} = \pm \frac{a_j x + b_j y + c_j}{\sqrt{a_j^2 + b_j^2}}$$

Input

The input file contains maximum 35 sets of inputs. The description of each set is given below:

First line of each set contains three integers C_x, C_y ($-10000 \leq C_x, C_y \leq 10000$) and n ($0 \leq n \leq 10000$). Each of the next n lines contains two integers x_i, y_i ($-20000 \leq x_i, y_i \leq 20000$) and a string of the form $a_i x + b_i y + c_i = 0$. Here (x_i, y_i) is the coordinate of a point around (C_x, C_y) and the string denotes the equation of the line segment formed by connecting (C_x, C_y) and (x_i, y_i) . You can assume that $(-100000 \leq a_i, b_i \leq 100000)$ and $(-2000000000 \leq c_i \leq 2000000000)$. This equation will actually be used to find the equations of bisectors of the angles that this line creates.

Input is terminated by a set where the value of n is zero.

Output

For each set of input produce one line of output. This line contains an integer number P that denotes of the $\frac{n(n-1)}{2}$ bisector equations how many are formed using the + sign in the bisector

equation
$$\frac{a_i x + b_i y + c_i}{\sqrt{a_i^2 + b_i^2}} = \pm \frac{a_j x + b_j y + c_j}{\sqrt{a_j^2 + b_j^2}} .$$

Sample Input	Output for Sample Input
5 2 5 12 7 10x-14y-22=0 1 -4 24x-16y-88=0 4 10 32x+4y-168=0 -1 9 56x+48y-376=0 12 -3 -10x-14y+78=0 10 10 0	4

Problemsetter: Shahriar Manzoor
Special Thanks: Derek Kisman



Problem C

Warp Speed

Input: Standard Input

Output: Standard Output

In the not-so-distance future, space engineers can invent a new technology for traveling through space and name it as “warp-drive”. This warp-drive can make a spaceship travel faster than light speed. It works by bending an amount of distance in space and make a ship travel through that bended space in a single “hop”. And since the time spent for each hop is equal, the total traveling time for any spaceship (that is equipped with this warp-drive) depends on the number of hops it make. Moreover, these engineers notice that distance of a hop depends on some kind of force fields in the space where that hop is operated. To travel from a beginning to an ending point, a spaceship may have to pass through many force fields, which may make that spaceship hop so many times.

And from their experiments, they found some facts as follow.

- A number of force field types is quite small.
 - So one English character can be used to name each of these force filed types.
- A spaceship can pass through every single force field in a single hop.
- A spaceship can pass through a certain sequences of two or more force fields in a single hop. And these certain sequences are kept as rules of single-hop-able sequences.
 - An example of list of these rules is shown as in the following table.

Rules of Single-hop-able Sequences
ab
abcd
abd
bde

- In this example, there are 4 rules, which means that a spaceship can pass through each of them using a single hop. The first one is “ab”, which means that a spaceship can pass through 'a' and 'b' force field in a single hop. The second one is “abcd” which means that it can pass through 'a', 'b', 'c' and 'd' in one hop.
- Please be notified that there is no “abc” sequence/rule in this example, so even though the ship can pass through “abcd” sequence, it is unable to pass through “abc” (in a single hop). It has to make 2 hops, the first hop is to pass through “ab” sequence and the second hop is to pass through “c” force field.

Goal

Suppose that you are an engineer on a battle spaceship. Your duty is to drive your spaceship through space as fast as possible. Your spaceship has a probe device that can identify a sequence of force fields along the path to your destination. In order to accomplish your duty, you have to build a computer program to find the minimum hop based on rules and any given sequences of force fields.

Input

Input is standard inputs which contain 2 parts of data which are separated by a blank line.

The first part is a set of rules of single-hop-able sequences. The number of rules is between 1 and 10,000.

- Each line in this first part contains one rule.
- Each rule is a (sub)sequence of force fields specified by a string of alphabets (a-z, A-Z).
- The order in every (sub)sequence is from left to right.
- The size of any (sub)sequence is between 2 and 20.

The second part is a set of force fields along path that a spaceship has to pass through. The number of sequence is between 1 and 200.

- Each line in this second part contains one sequence of force field in one path.
- Each sequence is specified by a string of alphabets (a-z, A-Z).
- The order in every sequence is from left to right.
- The size of any sequence is between 2 and 500.

The blank line after the second part is the termination of the input.

Output

For each sequence in the second part of input, write 2 parts of output as follows

- In the first line, write the total number of solutions followed by a space and the number of minimum hops.
- In the following lines, write each of the solutions in each line. Each solution contains a sequence of force fields as given in the input, where a space is inserted between each hop sub sequences. If there are two or more solutions, they must be sorted by ascending and lexicographical (ASCII) order.

Sample Input	Output for Sample Input
ab abcd abd bde CgeF abc abcd abde abdeabde abdeCgeFabde	1 2 ab c 1 1 abcd 2 2 a bde abd e 4 4 a bde a bde a bde abd e abd e a bde abd e abd e 4 5 a bde CgeF a bde a bde CgeF abd e abd e CgeF a bde abd e CgeF abd e

Problemsetter: Chalatif Thumkanon

More Explanations

There are 5 rules and 5 input sequences in this sample input.

In the first sample output, there is 1 solution with 2 hops. The first hop is from the first rule.

In the second output, there is 1 solution with 1 hop using the second rule for the whole sequence.

In the third output, there are 2 solutions with 2 hops. Its first solution is from the 4th rule and the second solution is from the 3rd rule.

In the fourth output, there are 4 solutions with 4 hops. The first solution is applied using the 4th rule twice. The second and third is applied using the 3rd and 4th rules in different position. The last solution is applied using the 3rd rules twice

The last output is quite similar to the fourth one, but it is also applied with the fifth rule in the middle of the sequence.



acm International Collegiate Programming Contest



event sponsor



Problem D

High-Precision Number

Input: Standard Input
Output: Standard Output

A number with 30 decimal digits of precision can be represented by a structure type as shown in the examples below. It includes a 30-element integer array (`digits`), a single integer (`decpt`) to represent the position of the decimal point and an integer (or character) to represent the sign (+/-). For example, the value `-218.302869584` might be stored as

<code>digits</code>	<input type="text" value="2"/>	<input type="text" value="1"/>	<input type="text" value="8"/>	<input type="text" value="3"/>	<input type="text" value="0"/>	<input type="text" value="2"/>	<input type="text" value="8"/>	<input type="text" value="6"/>	<input type="text" value="9"/>	<input type="text" value="5"/>	<input type="text" value="8"/>	<input type="text" value="4"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
<code>decpt</code>	<input type="text" value="3"/>															
<code>sign</code>	<input type="text" value="-1"/>															

The value `0.0000123456789` might be represented as follows.

<code>digits</code>	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="3"/>	<input type="text" value="4"/>	<input type="text" value="5"/>	<input type="text" value="6"/>	<input type="text" value="7"/>	<input type="text" value="8"/>	<input type="text" value="9"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
<code>decpt</code>	<input type="text" value="-4"/>															
<code>sign</code>	<input type="text" value="1"/>															

Your task is to write a program to calculate the sum of high-precision numbers.

Input

The first line contains a positive integer n ($1 \leq n \leq 100$) indicating the number of groups of high-precision numbers (maximum 30 significant digits). Each group includes high-precision numbers (one number in a line) and a line with only 0 indicating the end of each group. A group can contain 100 numbers at most.

Output

For each group, print out the sum of high-precision numbers (one value in a line). All zeros after the decimal point located behind the last non-zero digit must be discarded

Sample Input	Output for Sample Input
<pre> 4 4.12345678900000000005 -0.00000000012 0 -1300.1 1300.123456789 0.0000000012345678912345 0 1500.61345975 -202.004285 -8.60917475 0 -218.302869584 200.0000123456789 0 </pre>	<pre> 4.12345678888000000005 0.0234567902345678912345 1290 -18.3028572383211 </pre>

Problemsetter: Seksun Suwanmanee



Problem E

Configurations

Input: Standard Input
Output: Standard Output

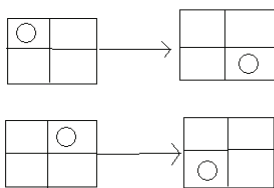
Well, in this problem you are given an $R \times C$ grid ($1 \leq R \leq 10^9$ and $1 \leq C \leq 10$). There will be B blocks ($1 \leq B \leq 100$) in the grid. Each block will be placed in a cell of the grid. There can be more than one blocks in a cell.

Now you are given M identical tokens and you can place them in the first row as you like. A cell cannot contain more than one token and you also cannot place a token in a cell occupied by blocks. Now you can move a token but you have to follow following rules:

1. If there is a token in a cell (r, c) then you can move it to either $(r+1, c-1)$ or $(r+1, c+1)$.
2. You cannot move a token to a cell occupied by blocks.
3. You cannot move a token outside of the grid.
4. You cannot move two or more tokens to the same cell.
5. All the tokens should be moved to i -th row before any token can be moved $(i + 1)$ -th row.

Now let $S = \{(1, c_1), (1, c_2), \dots, (1, c_M)\}$ be the set of cells of where you placed M identical tokens and $W(S) =$ number of ways you can move these tokens to last row. You have to find the sum of W for every possible S .

For $R = 2, C = 2, M = 1$ and $B = 0$ the answer is 2.



Input

First line contains number of test cases $1 \leq T \leq 500$. For each test case, the first line contains $1 \leq R \leq 10^9$, $1 \leq C \leq 10$ and $0 \leq M \leq C$ respectively. The second line contains $0 \leq B \leq 100$, followed by B lines and each of those B lines contains two integers r and c , ($1 \leq r \leq R$ and $1 \leq c \leq C$) indicating the cell position of each block.

Output

For each test cases you have to output the answer in a single line as shown in the sample output. As the answer can be very large you have to mod the output with 12345.

Sample Input	Output for Sample Input
3 1000000000 10 0 0 1000000000 10 2 0 10202 10 2 4 10 3 11 2 20 3 20 5	Case 1: 1 Case 2: 4973 Case 3: 3205

Problemsetter: Kazi Rakibul Hossain

Special Thanks: Md. Arifuzzaman Arif, Jane Alam Jan



Problem F

Two Longest Paths

Input: Standard Input
Output: Standard Output

Tekuhp is a tourist city. There are N intersections in the city, connected with M one-way roads. Each one-way road connects from some intersection to another. There maybe many roads that connect a pair of intersections. To make the city very amazing, the roads are constructed so that it is ***not possible*** to start at some intersection, travel along the roads, and return to the starting intersection. (It remains a strange secret how people of Tekuhp return home from work each day.)

There are two groups of tourists planning to visit the city. They want to travel along the roads from some intersection to another. However, both groups do not want to run into each other. So they want two paths P_1 and P_2 , each P_i , for $1 \leq i \leq 2$, starts at some intersection s_i and ends at intersection t_i , such that both paths share ***no intersections***, including the starting and the ending intersections. However, it is possible that a path P_i may contain only one node, i. e., $s_i = t_i$.

Tourists also want to visit many places. Since you are a good planner, you want to maximize the total number of intersections in both path.

Input

First line of the input contains an integer T ($1 \leq T \leq 10$), the number of test cases. After that T test cases follow.

Each test case starts with integers N and M ($1 \leq N \leq 300$; $1 \leq M \leq 3,000$), where N denotes the number of intersections and M denotes the number of roads. The intersections are numbered from 1 to N . After that M lines, describing road connection, follow. Each line contains two integers A and B denoting that there is a one-way road from intersection A to intersection B .

Output

The output must contain T lines, each line for each test case. For each test case, the output contains an integer L denoting the maximum number of intersections in two non-intersecting paths.

Sample Input	Output for Sample Input
3	3
3 2	8
1 2	4
2 3	
8 9	
1 2	
2 3	
3 4	
5 6	
6 7	
7 8	
2 6	
6 3	
3 7	
4 2	
1 2	

Problemsetter: Jittat Fakcharoenphol



acm International Collegiate Programming Contest



event sponsor



Problem G

A Minimum Land Price

Input: Standard Input
Output: Standard Output

Manager of ACM-ICPC Thailand Contest Council is planning to buy lands in Phuket to build the office building for national programming skill camp and programming contest that will be held on Phuket regularly in the future. The land price in Phuket is becoming more expensive in every year. The price increases in the exponential growth curves by a factor of year. If the land i whose initial cost is L_i bought in t years from now, its price will be $2^t (L_i)^t$. All land prices are different. ACM-ICPC can buy only one land per year. You have to help the manager to buy the lands at lowest price within the budget of 5,000,000 millions baht.

For example, if we want to buy 3 lands with costs 7 2 and 10 in 3 consecutive years, the total price will be calculated as follow.

$$(2 \times 7) + (2 \times 2^2) + (2 \times 10^3) = 2022 \text{ millions baht}$$

Input

First line of the input contains an integer T ($1 \leq T \leq 10$), the number of test cases. Each test case contains integer L_i which is the cost of land in million baht. There are less than 40 lands in each test case. The line contains 0 (zero) indicates the end of each test case.

Output

For each test case, print out the minimum price for purchasing all lands. If the total price exceeds the budget (5,000,000 millions baht), print out "Too expensive".

Sample Input	Output for Sample Input
3	134
7	17744
2	Too expensive
10	
0	
20	
29	
31	
0	
42	
41	
40	
37	
20	
0	

Problemsetter: Amarin Deemagarn
Special Thanks Seksun Suwanmanee



Problem H

Hackers' Crackdown

Input: Standard Input
Output: Standard Output

Miracle Corporations has a number of system services running in a distributed computer system which is a prime target for hackers. The system is basically a set of N computer nodes with each of them running a set of N services. Note that, the set of services running on every node is same everywhere in the network. A hacker can destroy a service by running a specialized exploit for that service in all the nodes.

One day, a smart hacker collects necessary exploits for all these N services and launches an attack on the system. He finds a security hole that gives him just enough time to run a single exploit in each computer. These exploits have the characteristic that, its successfully infects the computer where it was originally run and all the neighbor computers of that node.

Given a network description, find the maximum number of services that the hacker can damage.

Input

There will be multiple test cases in the input file. A test case begins with an integer N ($1 \leq N \leq 16$), the number of nodes in the network. The nodes are denoted by 0 to $N - 1$. Each of the following N lines describes the neighbors of a node. Line i ($0 \leq i < N$) represents the description of node i . The description for node i starts with an integer m (Number of neighbors for node i), followed by m integers in the range of 0 to $N - 1$, each denoting a neighboring node of node i .

The end of input will be denoted by a case with $N = 0$. This case should not be processed.

Output

For each test case, print a line in the format, "Case X : Y ", where X is the case number & Y is the maximum possible number of services that can be damaged.

Sample Input	Output for Sample Input
<pre>3 2 1 2 2 0 2 2 0 1 4 1 1 1 0 1 3 1 2 0</pre>	<pre>Case 1: 3 Case 2: 2</pre>

Problemsetter: Mohammad Mahmudur Rahman
Special Thanks Manzurur Rahman Khan



Problem I

Shuffle

Input: Standard Input
Output: Standard Output

My office is quite far from my home. Every day, I go there by bus. That takes a lot of time. More over, due to traffic jams, the time to reach my office is not fixed. To make the best use of this time, I carry my music player all the time.

My music player has a ‘shuffle’ mode. In that mode, the music player selects a track randomly that has not been played in that session, and starts playing. It will continue doing so, until, I have stopped playing, or it has played all tracks in that session. For the purpose of this problem, you can assume that all tracks are of equal duration. To select randomly, the player keeps list of the tracks already played. That is, if I listen to one track now, from next shuffle session, that track will be marked as played. To select a random track, the player always use such strategy that, selecting an old track (Already listened) is twice more probable than a new track (Not listened).

I go to my office in the morning and return home at night. It takes at least p minutes and at most q minutes to reach my office. The time required will always be an integer. You can assume that all travel times are equiprobable (that is, if $p = 2$ and $q = 4$, and if $P(t)$ is the probability to reach office in t minutes, then $P(2) = P(3) = P(4)$). If all tracks are of 1 minute, how many days on average are required to listen to all the songs?

Input

First line of input contains an integer $T(\leq 10053)$, the number of test cases.
Each test case contains 5 integers, $N (1 \leq N \leq 75)$, $p_1, q_1, p_2, q_2 (0 \leq p_1, q_1, p_2, q_2 \leq 30, p_1 \leq q_1, p_2 \leq q_2)$, where N is the number of songs, p_1, q_1 are the minimum and maximum number of songs listened during day, and p_2, q_2 are the minimum and maximum number of songs during night.

Output

For each test case, output the test case, followed by the expected number of days required to listen all songs. If it’s not possible to listen to all songs, output “IMPOSSIBLE”. Errors less than 10^{-6} will be tolerated.

Sample Input	Output for Sample Input
5	Case 1: 1.3333
1 0 1 0 1	Case 2: 2.0000
1 0 1 0 0	Case 3: 2.0000
1 0 0 0 1	Case 4: 4.0000
2 1 1 0 0	Case 5: 4.9630
3 0 1 0 2	

Problemsetter: Manzurur Rahman Khan
Special Thanks: Arifuzzaman Arif

Description of sample case 1:

Each day, I will listen to either 1 song, or no song. Probability of listening to 1 song is

$$\frac{1}{2} + \frac{1}{2} \times \frac{1}{2} = \frac{3}{4}$$

Probability of not listening to any song is $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$

So, probability of listening the song on the first day = $\frac{3}{4}$

Probability of listening the song on the second day = $\frac{1}{4} \times \frac{3}{4}$

Probability of listening the song on the second day = $\frac{1}{4^2} \times \frac{3}{4}$

And so on. So, the expected number of days will be:

$$1 \times \frac{3}{4} + 2 \times \frac{3}{4^2} + 3 \times \frac{3}{4^3} + \dots \infty$$

$$= \frac{3}{4} \times \left(1 + \frac{2}{4} + \frac{3}{4^2} + \dots \infty \right)$$

The infinite sum evaluates to $\frac{16}{9}$, and thus, expected days = $\frac{3}{4} \times \frac{16}{9} = \frac{4}{3} = 1.33333333 \dots$



acm International Collegiate Programming Contest

IBM

event sponsor



Problem J

Maximum GCD

Input: Standard Input
Output: Standard Output

Given the N integers, you have to find the maximum GCD(greatest common divisor) of every possible pair of these integers.

Input

The first line of input is an integer $N(1 < N < 100)$ that determines the number of test cases. The following N lines are the N test cases. Each test case contains $M(1 < M < 100)$ positive integers that you have to find the maximum of GCD.

Output

For each test case show the maximum GCD of every possible pair.

Sample Input	Output for Sample Input
3	20
10 20 30 40	1
7 5 12	25
125 15 25	

Problemsetter: Amarin Deemagarn



acm International Collegiate
Programming Contest

IBM

event
sponsor



Problem K

Palindrome Again

Input: Standard Input

Output: Standard Output

You are given a string S of length N . Can you find a string P which satisfies the following conditions?

1. Length of P will be N
2. Distance between S and P will be less than or equal to K
3. P will be a palindrome.
4. P can contain only characters 'a', 'b', ..., 'z'

You have to calculate, in how many ways you can choose P . This number can be very large, so print the answer modulo 1000000000 (10^9).

Notes:

- A string is a sequence of characters. For this problem consider that all strings can contain only 'a', 'b', ..., 'z', i.e. 26 available characters.
- The length of the string is defined by the number of characters in the string. For example, length of "abcba" is 5.
- A string is called palindrome when it is the same string when written from forwards or backwards. For example – "abcba", "abba", "a" are palindrome but "abc" is not a palindrome.
- Distance between two string of same length is the number of mismatches of corresponding characters. For example, distance between "abcb" and "bcba" is 4 because no character of first string matches to the character of the corresponding index of second string, but distance between "abc" and "cba" is 2.

Input

Input starts with an integer T (T is around 5000), the number of test cases.

Each test case consists of two lines. First line contains two integers N ($1 \leq N \leq 1000$) and K ($0 \leq K \leq 1000$). Second line contains a string S of length N . S contains only characters from 'a', 'b', ..., 'z'.

Output

For each test case output the number of test cases followed by the number of ways the string can be chosen modulo 1000000000 (10^9). See sample output for exact format.

Sample Input	Output for Sample Input
3 3 2 kxk 4 1 addc 4 3 Addc	Case 1: 51 Case 2: 2 Case 3: 76

Problemsetter: Arifuzzaman Arif
Special Thanks Manzurur Rahman Khan



acm International Collegiate Programming Contest

IBM event sponsor



Problem L

L's in a Circle

Input: Standard Input
Output: Standard Output

Packing problems are often very interesting. In this problem we will try to put five L's in a circle. But before that we will try to define L-shape for this problem.

We can divide a $(2a \times 2a)$ square into four $(a \times a)$ squares as shown in figure 1. Now if we remove one of those smaller squares from the figure, the new shape we get is referred as L-shape or L in this problem. The length of the smaller sides of this shape is a .

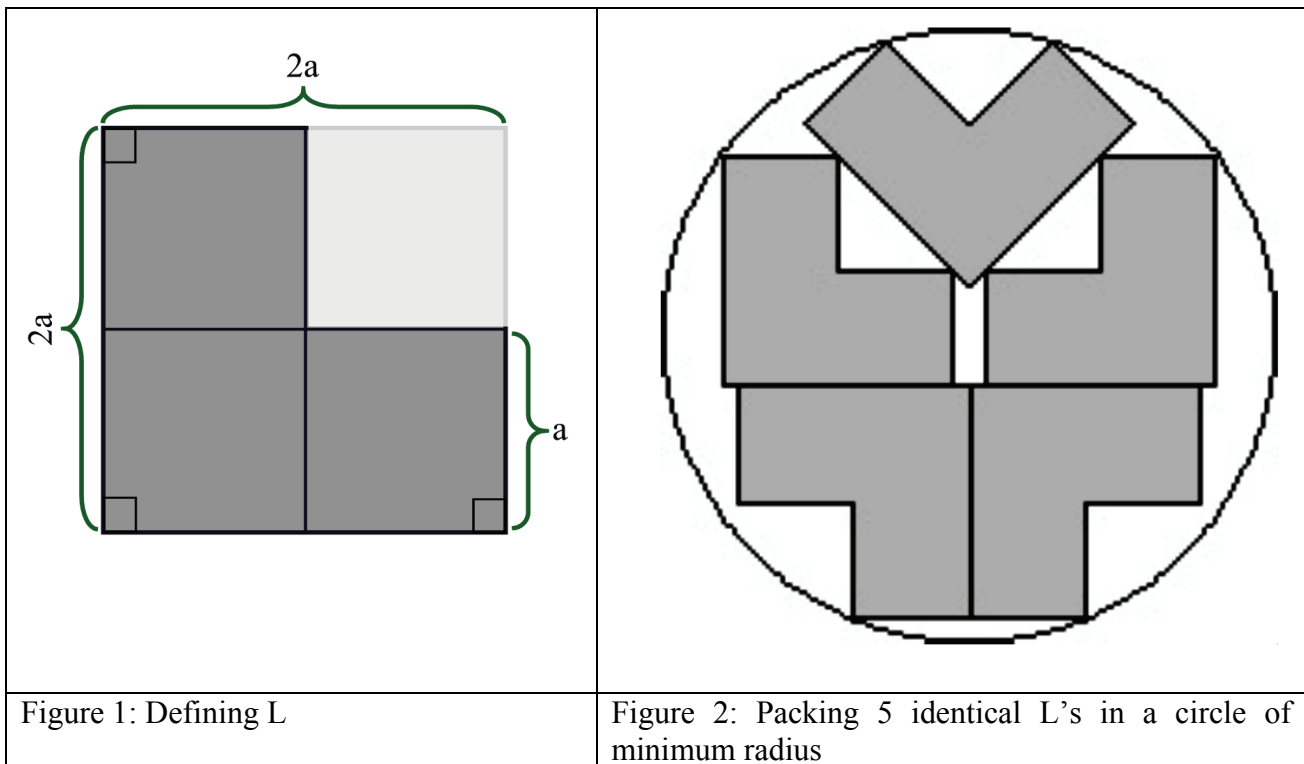


Figure 2 shows how five identical L's can be put into a circle of minimum possible radius (This is found by David Cantrell but not formally proved). Given the length of the smaller side a of an L, your job is to find the minimum possible radius of the circle in which five such L's can be put in the fashion shown in figure 2.

Input

Input file contains at most 5000 lines of inputs. Each line contains a floating-point number a ($0 < a < 10000$). The meaning of a is given in the problem statement. This floating-point numbers should have 7 digits after the decimal point. Input is terminated by a line containing a -1 . This line should not be processed.

Output

For each line of input produce one line of output. This line contains a floating-point number R that denotes the Minimum possible radius of the desired circle. This value should have 10 digits after the decimal point. Errors less than $\max(10^{-9}, 10^{-9} * RJ)$ will be ignored. Here RJ is the value of R produced by judge solutions.

Sample Input	Output for Sample Input
0.0000010	0.0000026406
0.0000001	0.0000002641
-1	

Problemsetter: Shahriar Manzoor
Special Thanks Derek Kisman