The 39^{th} ACM International Collegiate Programming Contest

Asia Regional Beijing Online

Problem List

Problem A	Always Cook Mushroom
Problem B	Building
Problem C	Cowboy
Problem D	Delivery
Problem E	Explosion
Problem F	Frog
Problem G	Grade
Problem H	Hilarity
Problem I	Instrusive
Problem J	Just A Challenge

Problem A: Always Cook Mushroom

Description

Matt has a company, *Always Cook Mushroom (ACM)*, which produces high-quality mushrooms.

ACM has a large field to grow their mushrooms. The field can be considered as a 1000×1000 grid where mushrooms are grown in grid points numbered from $(1, 1)$ to $(1000, 1000)$. Because of humidity and sunshine, the productions in different grid points are not the same. Further, the production in the grid points (x, y) is $(x + A)(y + B)$ where A, B are two constant.

Matt,the owner of *ACM* has some queries where he wants to know the sum of the productions in a given scope(include the mushroom growing on the boundary). In each query, the scope Matt asks is a right angled triangle whose apexes are $(1, 1), (p, 1), (p, q)$ $1 \leq p, q \leq 1000$.

As the employee of *ACM*, can you answer Matt's queries?

Input

The first line contains one integer T, indicating the number of test cases.

For each test case, the first line contains two integers:$A, B(0 \leq A, B \leq 1000)$.

The second line contains one integer $M(1 \leq M \leq 10^5)$, denoting the number of queries.

In the following M lines, the i-th line contains three integers $a, b, x(1 \leq a, b \leq 10^6, 1 \leq x \leq 1000)$,denoting one apex of the given right angled triangle is $(x, 1)$ and the slope of its base is (a, b). It is guaranteed that the gird points in the given right angled triangle are all in valid area, numbered from $(1, 1)$ to $(1000, 1000)$.

Output

For each test case, output $M + 1$ lines.

The first line contains "Case #x:", where x is the case number (starting from 1)

In the following M lines, the i-th line contains one integer, denoting the answer of the i-th query.

Sample

INPUT	OUTPUT
2 0 0 3 3 5 8 2 4 7 1 2 3 1 2 3 3 5 8 2 4 7 1 2 3	Case #1: 1842 1708 86 Case #2: 2901 2688 200

Problem B: Building

Description

Once upon a time Matt went to a small town. The town was so small and narrow that he can regard the town as a pivot. There were some skyscrapers in the town, each located at position x_i with its height h_i. All skyscrapers located in different place. The skyscrapers had no width, to make it simple. As the skyscrapers were so high, Matt could hardly see the sky. Given the position Matt was at, he wanted to know how large the angle range was where he could see the sky. Assume that Matt's height is 0. It's guaranteed that for each query, there is at least one building on both Matt's left and right, and no building locate at his position.

Input

The first line of the input contains an integer T, denoting the number of testcases. Then T test cases follow.

Each test case begins with a number $N(1 \leq N \leq 10^5)$, the number of buildings.

In the following N lines, each line contains two numbers, $x_i(1 \leq x_i \leq 10^7)$ and $h_i(1 \leq h_i \leq 10^7)$.

After that, there's a number $Q(1 \leq Q \leq 10^5)$ for the number of queries.

In the following Q lines, each line contains one number q_i, which is the position Matt was at.

Output

For each test case, first output one line "Case #x:", where x is the case number (starting from 1).

Then for each query, you should output the angle range Matt could see the sky in degrees. The relative error of the answer should be no more than 10^{-4}.

Sample

Input	Output
3 3 1 2 2 1 5 1 1 4 3 1 3 2 2 5 1 1 4 3 1 4 2 3 5 1 1 4	Case #1: 101.3099324740 Case #2: 90.0000000000 Case #3: 78.6900675260

Problem C: Cowboy

Description

There are N cowboys in the arena, tagged with 1 to N. The arena can be considered as an $N \times N$ grid.

At first, every cowboy is standing on a grid point (lower left point is $(1,1)$ and upper right is (N,N)).

Then the god named Matt will give M orders to the cowboys. All orders look like this: **(i,dir,distance)** which means the i-th cowboy moving **distance** step in **dir** direction.

And **dir** will be:

- '1' for East ((x,y)->(x + 1, y) for one step)
- '2' for North ((x,y)->(x, y + 1) for one step)
- '3' for West ((x,y)->(x - 1, y) for one step)
- '4' for South ((x,y)->(x, y - 1) for one step)

Since the cowboys can't get out of the arena, they will stop moving if they reach the boundary of the arena.

The cowboys are so aggressive that they always want to duel, but only the one that is moving can duel another cowboy. If a cowboy A is given an order, he will first turn his direction to the moving direction.

Then A will look at both his left hand direction and right hand direction, and the point now he is standing. If someone B is insight, A and B will have a duel, and the one with a smaller tag number will be the winner and loser will leave the arena, and A will stop moving immediately.

If no one is insight, A will go one step and start looking again. The process will stop if someone is insight, or A has moved for 'distance' steps and no one is insight.

Since the Cowboys' eyesight is not so good, they can only see someone if the distance between them is no more than D.

Since all cowboys are confident, if there are more than one cowboy insight, they will choose to duel the one with the smallest tag number.

Furthermore, Matt will give an order only if the previous cowboy has stopped. It's possible for more than one cowboy on one point. Nothing will happen if Matt gives an order to a cowboy that has left the arean.

Now, given the initial location of the cowboys and the orders, we want to know the final situation of the arena.

Input

The first line of the input contains an integer T, denoting the number of testcases. Then T test cases follow.

For each test case, the first line contains three integers:$N, M, D (1 \leq N \leq 50000, M \leq 50000, 0 \leq D)$.

In the following N lines, the i-th line contains two integers:$X, Y (1 \leq X, Y \leq N)$ indicating the initial location of the i-th cowboy.

In the following M lines, the i-th line contains three integers:$I, Dir, Dis (1 \leq I \leq N, 1 \leq Dir \leq 4, 0 \leq Dis \leq N)$ indicating the i-th order given by Matt.

All the numbers appears in the input will not exceed 10^9.

Output

For each test case, first output one line "Case #x:", where x is the case number (starting from 1).

Then for each cowboy that has not left the arena, output his tag number and final location in one line(Output in tag number order).

Sample

INPUT	OUTPUT
1 5 2 1 1 1 5 5 1 5 5 1 3 3 5 4 1 1 1 2	Case #1: 1 3 1 2 5 5 3 1 5 4 5 1

Problem D: Delivery

Description

Today, Matt goes to delivery a package to Ted. When he arrives at the post office, he finds there are N clerks numbered from 1 to N. All the clerks are busy but there is no one else waiting in line. Matt will go to the first clerk who has finished the service for the current customer. The service time t_i for clerk i is a random variable satisfying distribution $p(t_i = t) = k_i e^{-k_i t}$ where e represents the base of natural logarithm and k_i represents the efficiency of clerk i. Besides, accroding to the bulletin board, Matt can know the time c_i which clerk i has already spent on the current customer. Matt wants to know the expected time he needs to wait until finishing his posting given current circumstances.

Input

The first line of the input contains an integer T, denoting the number of testcases. Then T test cases follow.

For each test case, the first line contains one integer:$N(1 \leq N \leq 1000)$

The second line contains N real numbers. The i-th real number k_i $(0 \leq k_i \leq 1)$ indicates the efficiency of clerk i.

The third line contains N integers. The i-th integer indicates the time $c_i(0 \leq c_i \leq 1000)$ which clerk i has already spent on the current customer.

Output

For each test case, output one line "Case #x: y", where x is the case number (starting from 1), y is the answer which should be rounded to 6 decimal places.

Sample

Input	Output
2 2 0.5 0.4 2 3 3 0.1 0.1 0.1 3 2 5	Case #1: 3.333333 Case #2: 13.333333

Problem E: Explosion

Description

Everyone knows Matt enjoys playing games very much. Now, he is playing such a game. There are N rooms, each with one door. There are some keys(could be none) in each room corresponding to some doors among these N doors. Every key can open only one door. Matt has some bombs, each of which can destroy a door. He will uniformly choose a door that can not be opened with the keys in his hand to destroy when there are no doors that can be opened with keys in his hand. Now, he wants to ask you, what is the expected number of bombs he will use to open or destroy all the doors. Rooms are numbered from 1 to N.

Input

The first line of the input contains an integer T, denoting the number of testcases. Then T test cases follow.

In the first line of each test case, there is an integer N ($N \leq 1000$) indicating the number of rooms.

The following N lines corresponde to the rooms from 1 to N. Each line begins with an integer $k(0 \leq k \leq N)$ indicating the number of keys behind the door. Then k integers follow corresponding to the rooms these keys can open.

Output

For each test case, output one line "Case #x: y", where x is the case number (starting from 1), y is the answer which should be rounded to 5 decimal places.

Sample

Input	Output
2 3 1 2 1 3 1 1 3 0 0 0	Case #1: 1.00000 Case #2: 3.00000

Problem F: Frog

Description

Once upon a time, there is a little frog called Matt. One day, he came to a river.

The river could be considered as an axis. Matt is standing on the left bank now (at position 0). He wants to cross the river, reach the right bank (at position M). But Matt could only jump for at most L units, for example from 0 to L.

As the God of Nature, you must save this poor frog. There are N rocks lying in the river initially. The size of the rock is negligible. So it can be indicated by a point in the axis. Matt can jump to or from a rock as well as the bank.

You don't want to make the things that easy. So you will put some new rocks into the river such that Matt could jump over the river in maximal steps. And you don't care the number of rocks you add since you are the God.

Note that Matt is so clever that he always choose the optimal way after you put down all the rocks.

Input

The first line contains only one integer T, which indicates the number of test cases.

For each test case, the first line contains $N, M, L (0 \le N \le 2 \times 10^5, 1 \le M \le 10^9, 1 \le L \le 10^9)$. And in the following N lines, each line contains one integer within $(0, M)$ indicating the position of rock.

Output

For each test case, just output one line "Case #x: y", where x is the case number (starting from 1) and y is the maximal number of steps Matt should jump.

Sample

INPUT	OUTPUT
2 1 10 5 5 2 10 3 3 6	Case #1: 2 Case #2: 4

Problem G: Grade

Description

Ted is a employee of *Always Cook Mushroom (ACM)*. His boss Matt gives him a pack of mushrooms and ask him to grade each mushroom according to its weight. Suppose the weight of a mushroom is w, then it's grade s is

$$s = 10000 - (100 - w)^2$$

What's more, Ted also has to report the mode of the grade of these mushrooms. The mode is the value that appears most often. Mode may not be unique. If not all the value are the same but the frequencies of them are the same, there is no mode.

Input

The first line of the input contains an integer T, denoting the number of testcases. Then T test cases follow.

The first line of each test cases contains one integers N ($1 \le N \le 10^6$), denoting the number of the mushroom.

The second line contains N integers, denoting the weight of each mushroom. The weight is greater than 0, and less than 200.

Output

For each test case, output 2 lines.

The first line contains "Case #x:", where x is the case number (starting from 1)

The second line contains the mode of the grade of the given mushrooms. If there exists multiple modes, output them in ascending order. If there exists no mode, output "Bad Mushroom".

Sample

INPUT	OUTPUT
3 6 100 100 100 99 98 101 6 100 100 100 99 99 101 6 100 100 98 99 99 97	Case #1: 10000 Case #2: Bad Mushroom Case #3: 9999 10000

Problem H: Hilarity

Description

After June 1st, elementary students of Ted Land are still celebrating "The Sacred Day of Elementary Students". They go to the streets and do some elementary students stuff. So we call them "elementary students". There are N cities in Ted Land. But for simplicity, the mayor Matt only built $N-1$ roads so that all cities can reach each other. Some of the roads are occupied by the "elementary students". They will put an celebration hat on everyone who goes through the road without one. But if someone goes through the road with a celebration hat on his head, "elementary students"will steal the hat for no reason. Since Matt doesn't have a celebration hat, he wants to know how many different paths in his land that he can end up with a hat. Two paths are considered to be different if and only if they have different start city or end city. As the counsellor of the mayor Matt, you have to answer this question for him. The celebration elementary students are not stable: sometimes a new crowd of elementary students go to an empty road; sometimes the elementary students on a road will go back home and remain the road empty. Matt will send you the monitor about the change of elementary students on road and ask you the question above. You will be fired if you answer wrong.

Input

The first line of the input contains an integer T, denoting the number of testcases. Then T test cases follow.

For each test case, the first line contains $N (1 \le N \le 30000)$ describing the number of cities.

Then N lines follow. The ith line of these contains the name of the ith city, it's a string containing only letters and will not be longer than 10.

The following $N-1$ lines indicate the $N-1$ edges between cities. Each of these lines will contain two strings for the cities'name and a integer for the initial status of the edge (0 for empty, 1 for crowded).

Then an integer $M (1 \le M \le 60000)$ describes the number of queries. There are two kinds of query:

- "M i"means the status changing of the ith (starting from 1) road (0 to 1, 1 to 0);
- "Q"means that Matt asks you the number of different paths.

Output

For each test case, first output one line "Case #x:", where x is the case number (starting from 1). Then for each "Q"in input, output a line with the answer.

Sample

Input	Output
1 5 a b c d e a b 1 b c 0 c d 1 d e 1 7 Q M 1 Q M 3 Q M 4 Q	Case #1: 12 8 8 0

Problem I: Instrusive

Description

The legendary mercenary Solid Matt gets a classic mission: infiltrate a military base.

The military base can be seen as an $N \times N$ grid. Matt's target is in one of the grids and Matt is now in another grid.

In normal case, Matt can move from a grid to one of the four neighbor grids in a second. But this mission is not easy.

Around the military base there are fences, Matt can't get out of the base.

There are some grids filled with obstacles and Matt can't move into these grids.

There are also some surveillance cameras in the grids. Every camera is facing one of the four direction at first, but for every second, they will rotate 90 degree clockwisely. Every camera's sight range is 2, which means that if Matt is in the same grid as the camera, or in the grid that the camera is facing, he will be seen immediately and the mission will fail.

Matt has a special equipment to sneak: a cardbox. Matt can hide himself in the card box and move without being noticed. But In this situation, Matt will have to use 3 seconds to move 1 grid. Matt can also just hide in the cardbox without moving. The time to hide and the time to get out of the cardbox can be ignored.

Matt can't take the risk of being noticed, so he can't move without cardbox into a grid which is now insight of cameras or from a grid which is now insight of cameras. What's more, Matt may be in the cardbox at the beginning.

As a live legend, Matt wants to complete the mission in the shortest time.

Input

The first line of the input contains an integer T, denoting the number of testcases. Then T test cases follow.

For each test cases, the first line contains one integer:$N (1 \leq N \leq 500)$

In the following N lines, each line contains N characters, indicating the grids.

There will be the following characters:

- '.' for empty
- '#' for obstacle
- 'N' for camera facing north
- 'W' for camera facing west
- 'S' for camera facing south
- 'E' for camera facing east
- 'T' for target
- 'M' for Matt

Output

For each test case, output one line "Case #x: y", where x is the case number (starting from 1) and y is the answer.

If Matt cannot complete the mission, output '-1'.

Sample

INPUT	OUTPUT
2 3 M.. .N. ..T 3 M.. ### ..T	Case #1: 5 Case #2: -1

Problem J: Just A Challenge

Description

Matt receives a gift from his friend Bobo. It is a tree which has N vertices and $(N-1)$ edges with single character. Yes, it is the tree in computer science.

Matt denotes the number of edges between vertices a and b as $d(a,b)$. The string obtained by concatenating all characters from a to b is defined as $s(a,b)$.

Given string L and R of length M, Matt would like to know the number of (a,b) s.t.

- $d(a,b) = M$,
- $L \leq s(a,b) \leq R$.

where "$\leq$ " means lexicographical comparision.

Input

The first line of the input contains an integer T, denoting the number of testcases. Then T test cases follow.

For each test case, the first line contains two integers N ($2 \leq N \leq 10^5$) and M ($1 \leq M < N$).

Each of the following $(N-1)$ lines contains a_i, b_i, c_i denoting an edge between vertex a_i and b_i with character c_i ($1 \leq a_i, b_i \leq n$). The vertices are conveniently labled with $1, 2, \ldots, n$. c_i is either 'a' or 'b'.

The last two lines contain string L and R ($|L| = |R| = M, L \leq R$). L and R consist of 'a' and 'b'.

Output

For each test case, output one line "Case #x: y", where x is the case number (starting from 1), y is the number of (a,b).

Sample

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>2
2 1
1 2 a
a
b
4 2
1 2 a
2 3 b
3 4 a
aa
ab</pre></td><td><pre>Case #1: 2
Case #2: 2</pre></td></tr>
</table>

Hint

Please use `#pragma comment(linker, "/STACK:16777216")`