Problem A. Bart's Nightmare

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	3 seconds *(4 seconds for Java)*
Memory limit:	256 Mebibytes

Bart often has to stay at school after lessons for bad behavior. The most common punishment for him is to repeatedly write a certain phrase on the blackboard; such phrases usually begin with "I will not...".

Bart once had a bad dream. In this dream, Bart had to write a given phrase again, but this time, he had no blackboard. Instead, he found himself standing on a row of tiles. Each tile had a symbol written on it. Flames of hell were burning all around the tiles.

In order to write a symbol from a tile, Bart has to jump on it. Initially, Bart could jump on any tile; later, when standing on a tile, Bart can either move to an adjacent tile without jumping, or jump on the tile he is standing on with his left or right foot. When he jumps on a tile with some symbol on it, that symbol is added to the end of the current phrase. Initially, the current phrase is empty; Bart's goal is to obtain the exact phrase he is given. After that, if he jumped on a tile with his left foot, the symbols on all the tiles to the left, including the tile on which he jumped, disappears. Similarly, if he jumped on a tile with his right foot, the symbols on all the tiles to the right, including the tile on which he jumped, disappears. When Bart jumps on a tile which is already empty, nothing happens regardless of the foot he landed on. You may assume that the left side is always towards the start of the row and the right side is towards its end.

Bart thinks that if he succeeds to type the given phrase, he can stop this nightmare. But in the case something goes wrong, he also wants to know the number of different intervals on the initial row which he can use to obtain the given phrase. An interval $[i, j]$ of the row $s_1 s_2 \dots s_n$ is just a substring $s_i s_{i+1} \dots s_{j-1} s_j$ for some $1 \le i \le j \le n$. Intervals $[i_1, j_1]$ and $[i_2, j_2]$ are different if $i_1 \ne i_2$ or $j_1 \ne j_2$. Using a certain interval means no jumping outside that interval.

Input

The first line of input contains T, the number of test cases ($1 \le T \le 100$). Test cases follow. Each test case consists of two lines. The first of these lines describes the row of tiles of length N_i ($N_i > 0$). The second line contains the string to be obtained by jumping on that row; that string has length M_i ($1 \le M_i \le \min(N_i, 4096)$). Both lines consist only of Latin letters and underscore characters ('_'). Note that lowercase and uppercase letters are considered different. The sum of all N_i is no more than 65 536.

Output

For each test case, output a single line with a single integer on it — the number of different intervals on the given row which Bart can use to obtain the given phrase.

Example

input.txt	output.txt
3 will_ton_I I_will_not ton_I_lliw I_will_not AAA A	1 0 6

Problem B. Lucky Tickets

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	3 seconds *(4 seconds for Java)*
Memory limit:	256 Mebibytes

Lisa and Bart go to school by school bus every morning. Before they arrive, Bart wants to invent a new dirty trick to play on school principal. Lisa tries to interrupt Bart, so he decided to ask her to do something silly to get rid of her.

Last time he made Lisa count lucky tickets of $2N$ digits. Today Bart asked Lisa to find how many N-digit numbers are 2-lucky. A number is 2-lucky if the sum of its digits at positions 1, 2, 4, 8, ..., 2^k is the same as the sum of all its other digits. Positions in the number are counted from left to right, so digit 1 is the highest, digit 2 is second highest, etc. To make things simpler, Bart also gave Lisa the range of numbers he is interested in.

Your task is to write a program that calculates the required number faster than Lisa. She is used to solving such problems, so she does it in a few seconds.

Input

The first line of input contains an integer T, the number of test cases ($1 \le T \le 100$). Next T lines contain two integers A_i and B_i each — the range of numbers Bart is interested in. It is guaranteed that the lengths of both numbers are N_i such that $1 \le N_i \le 1024$. Note that A_i and B_i may have leading zeroes. You should count only N_i-digit 2-lucky numbers in the given range. It is guaranteed that $A_i \le B_i$.

Output

For each test case, write one integer on a line by itself — the number of N_i-digit numbers in the range $[A_i, B_i]$ which are 2-lucky.

Example

input.txt	output.txt
2 00 99 123 987	1 43

Problem C. Convexoid

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds *(3 seconds for Java)*
Memory limit:	256 Mebibytes

In the city of Springfield, UFO sightings are common. Most UFOs are seen near the nuclear power plant. There's even an alien living in Homer's cabinet. Or perhaps it is Bart's mutated bubble gum?.. Doesn't matter much. Homer calls it a convexoid.

Homer hasn't yet realized that the alien is intelligent and therefore, organizes experiments on the alien. Homer already knows that the alien is a convex plane multicellular double-sided organism. Moreover, he discovered that on i-th day, alien grows by vector V_i, that is, if S is the set describing the alien body before i-th day, the alien body after i-th day is the set of points $S' = \{P + t \cdot V_i \mid P \in S, t \in [0, 1]\}$. Today Homer wants to know what would be the alien's surface area after each of the following M days.

Input

The first line of input contains two integers N and M ($1 \le N, M \le 10^5$). Alien has a shape of a convex polygon with N vertices. Next N lines describe the polygon in either clockwise or counter-clockwise order; each of these lines contains two integers X_i and Y_i separated by a space — coordinates of a vertex ($|X_i|, |Y_i| \le 2^{27}$). The following M lines also contain pairs of integers — coordinates of the vectors V_j for the next M days; these coordinates do not exceed 2^{14} by absolute value.

Output

For each of the next M days, output a single number — the surface area of the alien after that day. Remember that the convexoid is two-sided.

Example

input.txt	output.txt
3 2 0 0 1 0 0 1 1 1 1 -1	5 11

Problem D. Walking Around the City

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	3 seconds *(4 seconds for Java)*
Memory limit:	256 Mebibytes

Bart likes skateboarding, but he like to go only to the interesting places. Some of the interesting places are located far from each other, thats why on his way between such two places, he tries to visit other interesting places along the way.

Currently Bart is in interesting place S and he wants to make exactly D trips between pairs of adjacent interesting places and finally get to interesting place T. How many different routes are there for Bart that satisfy the conditions described above?

Input

The first line of the input contains three integers N, M and P — the number of interesting places in the city, the number of Bart's questions and the modulo ($1 \le N \le 100$, $1 \le M \le 100$, $1 \le P \le 10^9 + 7$). The following N contain N numbers each; j-th number of i-th line is 1 if Bart can go from place i to place j directly by skateboard, and 0 otherwise. The next M lines contain Bart's questions. Each of them consists of three integers S, T and D ($1 \le S, T \le N$, $1 \le D < 2^{60}$). Such line describes a question "How much routes, taken modulo P, are there if we go from S to T in exactly D steps?".

Output

For each Bart's question, write the answer on a line by itself.

Example

input.txt	output.txt
4 2 1000 0 1 1 0 0 0 1 1 0 0 0 1 0 1 0 0 1 4 4 2 3 3	2 1

Problem E. Game

Input file:	input.txt
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Once Bart went to the Milhouse's. Milhouse invited Bart to join the following game.

At the start of the game, there is a number $p_0 = 1$ written on a sheet of paper. Then players write numbers sequentially, one number per turn, so Bart's numbers are p_{2i+1}, and Milhouse's numbers are p_{2i}.

The number p_m written on step m must satisfy the following conditions:

- Each p_m should be not greater than a given integer N.
- For any non-negative integer $i < m$ the inequality $p_i \neq p_m$ holds.
- $1 \leq p_m - p_k \leq A$ or $p_m = s_i \cdot p_k$, where $1 \leq k < m$ and integer A and set of integers S are game parameters.

Player who writes the number N wins. Loser gives to winner one candy and — additionally — as much candies as the number of positive integers between 1 and N, inclusive, that are missing on the paper.

Supposing that both players play optimally, find out the game result for Bart.

Input

The first line of input contains one integer T ($1 \leq T \leq 256$) — the number of test cases. Each test case consists of three lines: first of them contain two integers N and A — goal number and one of game parameters ($2 \leq A + 1 \leq N < 2^{60}$), second one contains integer M — number of elements in S, third and last one contains M integers s_i ($2 \leq s_i \leq N$). Sum or M for all test cases is not greater than 256.

Output

For each test case, print out a single line with one integer — the maximum number of candies Bart can get (it can be negative if Bart can't win).

Example

input.txt	output.txt
2 4 1 1 2 6 1 3 2 3 4	-2 3

Problem F. Itchy and Scratchy

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	5 seconds *(10 seconds for Java)*
Memory limit:	256 Mebibytes

Bart likes "The Itchy and Scratchy" TV show. One of the episodes describes a story where Itchy is in a room that has the form of N-dimensional parallelepiped. The room is divided into N-dimensional cubic cells and, initially, Itchy is inside of the one of these cells.

Scratchy wants to catch him, i. e. to get to the same cell where Itchy is. Before every move, Scratchy chooses some integer d. Then, independently for each coordinate, he may either change it by $\pm d$ or leave it unchanged.

From time to time Itchy appears in some cell. All cells have the equal probability to be chosen by Itchy.

As soon as Scratchy realizes where the opponent is, he tries to catch him as fast as he can. For that purpose he can make two moves before Itchy vanishes.

You should write a program calculating the probability to catch Itchy.

Input

The first line of the input contains an integer T ($1 \le T \le 4096$) — the number of test cases. Each test case is described by three lines. The first line contains a positive integer N ($1 \le N \le 10$). The second one contains N positive integers a_i — sizes of the N-dimensional parallelepiped in i-th dimensions respectively ($1 \le a_i \le 512$ in case of $T = 1$, and $1 \le a_i \le 10$ otherwise). The number of cells in the given parallelepiped is not greater than 2^{60}. The third line of each test case contains N positive integers x_i — Scratchy's initial coordinates ($1 \le x_i \le a_i$).

Output

For each test case, you should output a single line with the probability of Scratchy's success in the form of irreducible fraction.

Examples

input.txt	output.txt
1 2 100 100 50 50	1/1
1 3 10 5 2 1 1 1	87/100

Problem G. Billiards

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

"The Itchy and Scratchy" TV show put an idea into Bart's head to play N-dimensional billiards. In this game, pockets are located in the vertices of the N-dimensional box. Bart set the ball and hit it. The ball changes its direction only when it meets the side of the box and the angle of incidence is equal to the angle of reflection. Kinetic energy of the ball is a constant and the ball might fly forever.

Your task is to write a program to find out if the ball falls into some pocket or not.

Input

The first line of the input contains a positive integer T ($1 \le T \le 10\,000$) — the number of test cases. Each test is described by four lines. The first line contains a positive integer N ($1 \le N \le 5$). The second one contains N positive integers — box size a_i for each dimension i. The third line describes the ball's position. It contains N nonnegative integers — coordinates for all dimensions $x_1, x_2, \ldots, x_n$ ($0 \le x_i \le a_i$). The fourth line contains N integers $v_1, v_2, \ldots, v_N$ — the speed vector. At least one of the speed coordinates is not equal to zero. Box sides are parallel to coordinate axes. Opposite box vertices have coordinates $(0, \ldots, 0)$ and $(a_1, a_2, \ldots, a_N)$. It is guaranteed that $1 \le \mathrm{LCM}(v_i), \mathrm{LCM}(a_i) \le 2^{20}$.

Output

For each test case, you should output a single line with the ball's flight duration as an irreducible fraction or "`Never`" if the ball never falls into any pocket. Note that the duration may be zero if the ball starts in a pocket.

Examples

input.txt	output.txt
3 2 7 6 2 3 2 3 2 3 3 2 1 3 -3 2 2 2 1 2 1 1	13/1 1/3 Never

Problem H. Fantastic Throw

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Bart is going to play with Santa's Little Helper. Bart is throwing rings and the dog catches and brings them back. Bart threw two rings and accidentally they flew close together. Homer was looking at the game and wants to know if the rings share common point, fly at a distance or by some magic combined in a chain.

Your task is to write a program that gives Homer the answer. Rings has a form of a circle and are given by normal vector of a plane that includes the circle, circle center coordinates and circle radius.

Input

The first line of the input file contains a positive integer T ($T \le 10^4$) — the number of test cases. First line of a test case consists of seven integers $X_1,\ Y_1,\ Z_1,\ R_1,\ NX_1,\ NY_1,\ NZ_1$ each — circle center coordinates, circle radius and normal vector coordinates for the first circle. Second and last line of a test case consists of the same parameters $X_2,\ Y_2,\ Z_2,\ R_2,\ NX_2,\ NY_2,\ NZ_2$ for the second circle ($-2^{20} \le X_i,\ Y_i,\ Z_i,\ NX_i,\ NY_i,\ NZ_i \le 2^{20},\ 1 \le R_i \le 2^{20}$).

Output

Your program should output "`Touch`" if the rings share a common point, "`Incredible`" if they form a chain, and "`Near`" otherwise.

Example

input.txt	output.txt
3 0 0 0 1 0 0 1 2 0 0 1 0 1 0 0 0 0 1 0 0 1 3 0 0 1 0 1 0 0 0 0 1 0 0 1 1 0 0 1 0 1 0	Touch Near Incredible

Problem I. Equation

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Bart had decided to play the fool in the school. During a break, he wrote on a blackboard some symbols unknown to him:

$$(x^2 - xy - y^2 + 14x - 57y - 451)(y^2 - yz - z^2 + 27y - 46z - 29) = -1$$

By the happy accident, it was an equation that had positive integer solutions. Probably, you will find such a solution by your own. But your task is to write a program that finds a positive integer solution (x, y, z) such that the sum $s = x + y + z$ is as large as possible but not greater than a given integer A.

Input

The first line of input contains a positive integer T ($T \le 100$) — the number of test cases. Each of the next T lines contains a positive integer A ($60 \le A \le 10^{1000}$).

Output

For each test case, print out a single line that contains x, y, z — the solution for this case. If there are several possible answers, print any of them.

Example

input.txt	output.txt
1 60	38 14 8

Problem J. Arcade

Input file:	input.txt
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Bart likes to play slot machines, especially arcades. The point of his favorite game is to control a spaceship in a battle against enemies shooting other spaceships, avoiding enemy missiles, and collecting bonus items left after destruction of any ship.

The game field can be represented as a coordinate plane. The player's spaceship has initial coordinates $(0, 0)$. It moves up with a constant speed V_y, and can also move left or right with a speed V_x.

Assume that Bart shot all the enemies and avoided all the missiles launched at him. Now it is time to collect bonus items.

Your task is to write a program which calculates the maximum possible amount of bonus items Bart can collect and the number of different ways to do it. Two ways are different if they include different sets of bonus items or use different orders for collecting them. Note that if there are multiple bonus items in the same position, you can still choose any order in which you will collect them.

Input

The first line of the input contains two positive integers V_x and V_y ($1 \le V_x, V_y \le 2^{30}$) — the vertical and horizontal speeds.

The second line contains the number of bonus items N ($1 \le N \le 10^5$).

Next N lines contain two integers x_i, y_i — bonus items' coordinates ($-2^{30} \le x_i < 2^{30}$, $0 \le y_i < 2^{30}$).

Output

Output a single line with two integers separated by space — the maximum number of bonus items to collect and the number of different ways to do it modulo $10^9 + 7$.

Example

input.txt	output.txt
1 1 3 1 1 -1 1 0 3	2 2

Problem K. Differential

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Bart doesn't like to go to school and he doesn't like to study. Most of all he is bored of maths. After the equation joke, he gets extremely hard exercises. Now he has to find a derivative of a given complex function. A complex function is defined as follows:

```
<complex function>::=<simple function>(<complex function>)|x|pow(<complex function>,<exponent>)
<exponent>::=<complex function>|<number>
<simple function>::=sin|cos|tg|ctg|ln
<number>::=2|3|4|5|6|7|8|9
```

and $\mathrm{pow}(x, y) = x^y$.

A derivative is something like a complex function but also can have extra operations: multiplications, additions, subtractions, expressions in brackets and constants. The multiplication operator must be omitted. An expression contains several terms and each term can contain several factors.

Bart knows that he can make exchanges between terms and factors. He even can make multipilcation — and that's all... Luckily, Lisa knows something about derivatives. She gave him the following derivation rules:

- In every sum, all the additions should be written before all the subtractions.
- In every term, all the multiplications should be written before all the divisions.
- In case of addition or multiplication, the lexicographically first operand should be written first.
- 0 and 1 are never used as factors.
- 0 is never used as one of two or more terms in addition or subtraction.
- Instead of $\mathrm{pow}(f(x), 1)$ use $f(x)$, and instead of $\mathrm{pow}(f(x), 0)$ use 1.
- Numeric factor should be written before the functions and variables.

Your task is to write a derivative finding program that prints the solution, applying the rules above.

Input

The first line of input contains one positive integer T ($T \le 5000$) — the number of test cases. Each of next T lines consists of one function with correct syntax without spaces. Each function have no more than five opening brackets.

Output

For each test case, print out a single line that contains solution for this case.

Example

input.txt

```
13
cos(x)
sin(x)
ctg(x)
tg(x)
ln(x)
x
pow(x,3)
pow(sin(x),2)
pow(x,2)
pow(x,x)
pow(pow(x,3),3)
pow(pow(pow(pow(pow(x,5),5),5),5),5)
pow(pow(pow(x,2),2),2)
```

output.txt

```
-sin(x)
cos(x)
-1/pow(sin(x),2)
1/pow(cos(x),2)
1/x
1
3pow(x,2)
2cos(x)sin(x)
2x
(ln(x)+x/x)pow(x,x)
9pow(pow(x,3),2)pow(x,2)
3125pow(pow(pow(pow(x,5),5),5),5),4)pow(pow(pow(pow(x,5),5),5),4)pow(pow(pow(x,5),5),4)pow(pow(x,5),4)pow(x,4)
8pow(pow(x,2),2)pow(x,2)x
```

Problem L. Bart is A Tycoon

Input file:	input.txt
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Bart likes skateboarding and knows everything about it. He is going to become a tycoon and to have a skateboard producing factory.

Yet Bart thinks about the following problem. Assume he got M customers and C packs of skateboards. How to sell skateboards with maximum profit?

Input

The first line of input contains four integers N, S, C and R — number of cities, number of the city where Bart's factory is placed, number of packs of skateboards Bart got for sale and penalty for car accident ($1 \le N,\ C \le 100$, $1 \le S \le N$, $0 \le R \le 10^4$).

The second line contains an integer M — number of customers ($1 \le M \le 100$).

Next M lines consist of customer descriptions and contain three positive integers P_i, C_i and M_i each, where P_i is the customer city number ($1 \le P_i \le N$), C_i is the maximal number of packs of skateboards that customer could buy ($1 \le C_i \le 100$) and M_i is the profit that Bart gets from selling one pack to this customer ($1 \le M_i \le 10^4$). A customer can buy any integer number of packs of skateboards between 0 and C_i, inclusive.

Next line consists of one integer E — the number of roads ($0 \le E \le 10^4$). All roads between cities are directed, but car accidents are still possible.

Next E lines consist of road descriptions. Each road is described by six integers V_1, V_2, U, F, P_1 and P_2 — numbers of beginning and ending city on this road, maximum road capacity in trucks, fuel expenses for one truck to pass this road and two special parameters, related to car accident probability, respectively ($1 \le V_1, V_2 \le N$, $V_1 \ne V_2$, $1 \le U \le 100$, $1 \le F \le 10^4$, $0 \le P_1 \le P_2 \le 100$). One truck can carry one pack of skateboards. If during the whole period of deliveries, there will be more than U trucks on the road, there would be a traffic jam and all these trucks won't arrive in time. If $U > 1$, the probability of a car accident for i-th truck on the road is $P_1 + (P_2 - P_1)(i-1)/(U-1)$. In case of $U = 1$, the probability of a car accident for a truck is $P_1 = P_2$.

If a truck has accident, Bart pays R as a penalty and then the truck continues delivery. A truck can have no more than one accident per a road.

Output

Output expected profit with accuracy 10^{-2}. Note that Bart need not sell all C packs of skateboards.

Example

input.txt	output.txt
3 1 3 100 2 2 2 1000 3 1 900 2 1 3 2 5 10 10 1 2 2 5 10 20	2845.00