Problem A. Painting of building

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

In one company designers decided to repaint the gray wall of their office with a length of n meters in corporate colors: blue and orange. To do this, they bought an innovative robot painter.

The robot can paint the walls in accordance with the program recorded in it. The program is a sequence of commands. Each command is given the two integers and a color and tells the robot what section of the wall in which color to paint. For example, a wall with a BBOOO coloring plan can be obtained using a program consisting of two commands: paint in blue from first meter to fifth, and then in orange from third to fifth.

One of the programmers of the company was instructed to write a program, after executing which, the robot would paint the walls in accordance with the specified scheme. He easily coped with the task and even wrote a program containing the minimum possible number of commands. After that, he asked himself: how many programs of the same length exist that will paint the wall according to the coloring plan?

Input

The first line contains the number t — the number of tests. The following are descriptions of t tests.

In a test a single line consisting of letters B and O is given, where the letter at the i-th position is responsible for whether the i-th meter of the wall is to be painted blue or orange.

The total length of the lines does not exceed 500 000.

Output

For each of the t test cases, print a single number — the number of programs of minimal length leading to painting the wall according to the plan. Since the answer may be large, output the answer modulo $10^9 + 7$.

Example

standard input	standard output
3 BBOOO BBOBOOB OBOB	7 3 16

Problem B. Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

King of the Bytelandia holds an annual intellectual game. According to the rules of the game, all participants are divided into two teams, and a set of hints is given to each of the players. During the game, participants can exchange tips by the following rules: each player of the first team can approach to one player of the second and ask him for a hint, which he doesn't know yet. If there are several such tips, a member of the second team will tell him any of his choice. Each member of the first team can ask a hint only for one player of the second team, while several players can contact the player of the second team. The first team wins if it manages to collect all the clues. Help the captain of the first team to find out whether his team can collect all the clues, regardless of the answers of the second team.

Input

The first line of the input contains three integers n, m and k — the sizes of the teams and the number of hints, respectively ($1 \le n, m \le 500; 1 \le k \le 5000$). The following $n + m$ lines contain information about the tips available to players at the beginning of the game in the following format: the first number in the line corresponds to the number of tips for the player, and the following numbers contain the numbers of tips — natural numbers not exceeding k.

The players of the first team are numbered from 1 to n, and the second team is numbered from $n + 1$ to m.

Output

If the first team can collect all the hints, output 1. In the first line of the output print n numbers, for each member of the first team, specify which of the players of the second team he should ask for a hint.

If there are several answers, output any.

If the first team cannot collect all the hints, just output 2.

Examples

standard input	standard output
3 2 4 1 1 1 2 1 3 2 1 4 1 3	1 1 1 1
3 2 4 1 1 1 2 1 3 3 1 2 4 3 1 3 4	2

Problem C. Internet wire

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Popular in Flatland Internet service provider «Jota» decided to expand its influence. To achieve this, «Jota» decided to hold a new Internet wire. The wire can be considered as the line on the plane.

«Jota» knows that there are n potential subscribers in Flatland, the i-th subscriber lives in a house located at the coordinates (x_i, y_i), and no more than one subscriber lives in one house. The Internet wire can be configured so that it will distribute the Internet to all the subscribers who are exactly at some selected distance from it. In other words, if you set up the Internet wire to the distance d, it will distribute the Internet to those subscribers, the Cartesian distance from the point where the subscriber's house is located to the line defining the Internet wire will be d.

Help «Jota»'s engineers build an Internet wire such that the number of subscribers covered by this Internet line is maximum.

Input

The first line contains one positive integer t — the number of test cases in the input data.

The description of each test consists of several lines. The first line contains a single integer n ($1 \le n \le 10^3$) — the number of subscribers.

The next n lines contain two integers x_i, y_i each — coordinates of the i-th subscriber ($-10^9 \le x_i, y_i \le 10^9$). It is guaranteed that no two subscribers are at the same point.

The sum of n for all tests does not exceed 10^3.

Output

Print a single integer — the maximum number of subscribers that «Jota» can cover.

Example

standard input	standard output
2 5 1 1 2 2 5 5 2 1 -3 -4 4 0 0 6 0 3 3 3 6	5 3

Problem D. Boolean Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Vitalik thought for a long time how to solve a simple problem, and in the process of inventing a solution he drew a graph on the board. This graph is the root tree. At this point, Borya approached the board and began to write expressions in some vertices, such that boolean variables were assigned different values.

Then they introduced a rule that for each vertex of the tree you can determine the value of the variable in this vertex. Namely, if the assignment of some variable was written in some vertex, this value is used for this variable. If the value cannot be determined in this way, the value of this variable is taken from the nearest ancestor of this vertex, in which the assignment of the value of this variable is written. If the vertex has no such ancestor, then the value is undefined.

After several operations, they wanted to calculate how many leaves of the tree for a given variable the value is true, in how many false, and in how many not defined.

And then they realized that it is difficult, and that they cannot do it. So they ask you to help them, and after each time you add a value assignment to variable, determine for how many leaves the value of this variable is true, for how many false, and for how many undefined.

Input

The first line contains the number T — the number of tests. The following is a description of T tests.

The first line of the test contains two integers n and m where n — the number of vertices of a given tree, and m — additions of new assigments a value to a variable ($1 \le n, m \le 10^5$).

The second line contains the n of integers where i-th integer is the number of ancestor i-th vertex. For the root, this number is zero. Guarantees, you are given a tree.

Each of next m lines contains three integers — vertex number v, number of variable x and b — zero or one, the value of the variable equal to false or true, respectively. The variable is a positive integer not greater than 10^5.

For each test set the sum of all n and the sum of all m are not greater than 10^5.

Output

For each of the test cases print $m <$ lines containing three numbers — the number of leaves in the tree such that for assigned variable the value is true, false and undefined, respectively.

Example

standard input	standard output
2 4 1 0 1 1 3 3 1 1 8 4 0 1 1 3 3 3 5 5 3 1 1 5 2 1 3 2 0 1 1 0	1 0 1 4 0 1 2 0 3 2 2 1 4 1 0

Problem E. One More Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Petya advised Vasya a new game. Vasya really liked this game, and he wanted to beat Petya.

The game field consists of a system of levels, rooms and one-way corridors between them.

On the i-th level there are exactly i rooms with numbers from 1 to i on them. Every room has the two passages: from j-th room on i-th floor one passage leads to the j-th room on the $i+1$-th floor, and another leads to the $j+1$-th room on the $i+1$-th floor. Every passage has some length. Player needs to go from the first floor to the last by shortest possible route.

Vasya has chosen the next strategy in the game: being in the room he goes through shortest passage to the next floor. If both passages have the same length, he chooses one of them equiprobably.

Vasya noticed that his path can have different length depending on his random choices. You are her to calculate expectation of length of his path.

Input

The first line of the input contains one number t — number of tests cases in the input.

Each of the test cases starts with the line with one integer n, where $n+1$ is the number of floors in the building ($1 \le n \le 1000$).

Each of n next lines contains 2 positive integers for each room on the floor — j-th pair of integer on i-th line describes the lengths of the passages from j-th room on i-th floor to j-th and $j+1$-th rooms on the next floor correspondingly.

All lengths of passages doesn't exceed 1000 and the sum of n in each input doesn't exceed 1000.

Output

For each test case print one number — expectation of length of Vasya's route. Your answer should has absolute of relative precision not more that 10^{-6}.

Example

standard input	standard output
1 2 2 2 3 3 4 5	5.5000000000000000000

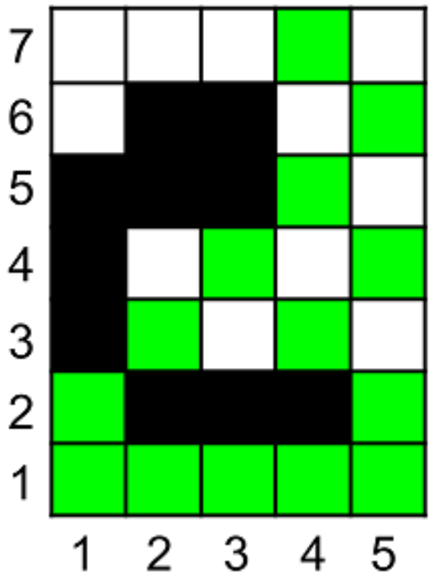
Problem F. Robot

Input file:	standard input
Output file:	standard output
Time limit:	8 seconds
Memory limit:	256 megabytes

There are field consisting of n columns and m rows. Robot can start his route in any cell in first row. Then he can make some (possibly zero) number of turns. Let the robot stand in column x and row y, then in one turn he can go to a cell $(x + 1,\ y + 1)$ or to a cell $(x - 1,\ y + 1)$. Of course, the robot can not go out borders of the field.

The robot's route is complicated by the fact that there are rectangular obstacles on the field. Their sides are parallel to the sides of the field. A robot can't go to cell that covered with an obstacle. Each pair of obstacles do not intersect, but can touch.

You need to count the number of cells that the robot can go to.



Input

The first line contains the number T — the number of tests. The following is a description of T tests.

The first line of the test contains two integers n and m — number of columns and rows of field ($1 \le n \le m \le 10^9$).

The next line contains a single integer k — number of obstacles on the field ($1 \le k \le 10^5$).

The following k lines contain a description of the obstacles. Eachline consists of of the four integers x_1, y_1, x_2, y_2, denoting the coordinates of the opposite angles of obstacles ($1 \le x_1 \le x_2 \le n$, $2 \le y_1 \le y_2 \le m$).

It is guaranteed that the obstacles do not intersect.

The total number of obstacles in the input file is not greater than 10^5.

Output

For each test case, print one integer — number of cells that the robot can go to.

Example

standard input	standard output
1 5 7 3 2 2 4 2 1 3 1 5 2 5 3 6	14

Problem G. Fibonacci

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Fibonacci numbers are one of the most well-known and studied sequences of numbers. Patterns based on these numbers are found in the most unexpected areas, for example, in the architecture of medieval buildings or the wild life. Your friend, a researcher, thinks that he has found a new pattern, but he will need your help to test his hypothesis.

Recall that the Fibonacci numbers are defined by the following recurrence relation:

$F_1 = 1, F_2 = 1, F_n = F_{n-1} + F_{n-2}$.

Help the researcher calculate the sum of k-th powers of the first n Fibonacci numbers ($\sum\limits_{i=1}^{n} F_i^k$) modulo $10^9 + 23$.

Input

The first line of the input contains integer t — the number of test cases in input ($1 \le t \le 100$).

Each of the next t lines contains two integers n and k each — number of Fibonacci numbers and the required power ($1 \le n \le 10^9; 1 \le k \le 10^{18}$).

Output

Print the answer to each test case in the separate line.

Example

standard input	standard output
3 1 1 3 2 1000 10	1 6 107235489

Problem H. Entertaining cryptography

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

Vasya and Petya decided to play spies. Vasya made a word of n lowercase English letters, and told Petya that the polynomial hash of his word is x. Your task is to help Petya to find out how many different words Vasya could have thought. Since this number can be quite large, output it modulo 998 244 353.

The polynomial hash from the string $s = a_0a_1 \ldots a_{n-1}$ is the value of the sum of $\sum\limits_{i=0}^{n-1} ord(a_i) \cdot p_i$, taken modulo m, here $ord(c)$ is the ordinal number of the character c in the Latin alphabet, for example, $ord(a) = 1$, $ord(b) = 2$, $ord(z) = 26$.

Input

The first line of input contains four integers n, m, p and x — the length of the string, the modulo in polynomial hash, the base of polynomial hash and the given value of hash of the string $(1n \leq 10^6, 2 \leq m \leq 10^4, 1 \leq p < m, 0 \leq x < m)$.

Output

Print a single integer — the number of words that Vasya could have thought, taken modulo 998 244 353.

Example

standard input	standard output
`1 10 8 7`	`2`

Problem I. Digit roots

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Recently, Grisha came up with a new game: he writes out all the numbers from a to b in order. Dima, noticing this, suggested writing out not numbers, but their digital roots, in order to save the space on the board. Grisha really liked this idea, so he decided to find out as soon as possible what the digital root of a number is.

The digital root of a number is calculated as follows: the sum of digits of this number is taken, if it is written in one digit, then it is a digital root, otherwise the sum of digits is calculated from this sum until a single digit is obtained. So, the digital root 16 is 7 ($1 + 6 = 7$), and the digital root 9991 is 1 ($9 + 9 + 9 + 1 = 28$; $2 + 8 = 10$; $1 + 0 = 1$).

Grisha began to write digital roots on the board, and Dima became wondered what the numbers on the board would be written most often.

Your task is to find, by the numbers a and b, all the digital roots that will be written on the board the greatest number of times.

Input

The first line contains a positive integer t ($1 \le t \le 10^4$) — the number of test cases in the input data.

Each test contains two natural numbers a and b — the boundaries of the segment that Grisha writes out on the board ($1 \le a \le b \le 10^{15}$).

Output

Print t lines.

For each test case, print the number n — the number of different numbers that will occur on the board the greatest number of times. Then, space-separated, print n numbers in ascending order — the numbers that will appear on the board the most times.

Example

standard input	standard output
4 1 5 7 7 9 12 12 22	5 1 2 3 4 5 1 7 4 1 2 3 9 2 3 4

Problem J. Subway

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Vasya and Petya ride the subway every day, so they decided to buy their own travel cards instead buying a tokens each time. Each card can be used for a certain number of days, after which it becomes inactive.

With each pass through the turnstile it is displayed how many more days the card can be used (including the current day). But, unfortunately, the scoreboard on which this number of days is displayed can only show one-digit and two-digit numbers. If there are at least three digits in the displayed number, the number 99 will appear on the display. For example, if there are 5 days left on the card, then the number 5 appears on the turnstile, if 12 days, then the number 12, and if 123 days, then on the turnstile the number 99 appears. If 0 days remain on the card, it becomes inactive and you can no longer pass through the turnstile with it.

Now Vasya has a on the card, and Petya has b on the card. Every day they take the subway and every day they look at the numbers that displayed on the turnstile. And it became interesting to them: after how many days for the first time the number on the turnstile of one of them will be exactly k times larger, than the number on the turnstile of the other. Help your friends to figure out the answer to this question.

Input

The first line of the input contains one number t — number of tests ($1 \le t \le 100\,000$).

Each of next t lines contains three numbers a, b and k — number of days on Vasya's and Petya's cards and required quotient ($1 \le a, b \le 2 \cdot 10^9$; $1 \le k \le 100$).

Output

For each test case print one number: after how many days one of the friends on the turnstile will have a number which is k_i times larger than number of another friend.

If this doesn't happen until the day when one of the friends's card becomes inactive, output -1.

Example

standard input	standard output
5 2 1 2 100 99 2 17 13 10 3 3 1 1 1 2	0 98 -1 0 -1

Problem K. Knapsack problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

As a homework for Computer Science, Petya and Vasya were asked to write a program that would solve a backpack problem. The task is formulated as follows: given N items with weights a_i, is it possible to choose a subset of things with a total weight equal to exactly W?

Petya and Vasya successfully coped with the task, having received Accepted in the testing system. But when discussing the problem, it turned out that Petya and Vasya used different approaches to the solution.

It turned out that Vasya actually counted the number of ways to select a subset of items with weight W, and when printing the answer he compared this number with zero. To simplify the implementation, Vasya did not count the number of ways, but its remainder from dividing by some number m. Petya immediately declared that Vasya's approach was wrong, but he could not come up with a test in which it gives the wrong answer. Can you find such a test?

For a given m, create a test for a backpack problem in which the number of ways to gain weight W is divisible by m without remainder.

Input

The only line of the input contains an integer m — the modulo for which calculations were made in the Vasya's program ($1 \le m \le 10^{18}$).

Output

If there is no such test, output a single number -1.

Otherwise, in the first line output two integers N and W separated with a space ($1 \le N \le 200; 1 \le W \le 500$). In the next line print N integers a_i ($1 \le a_i \le W$).

Example

standard input	standard output
`9`	`6 5` `1 1 1 4 4 4`