ABBYY Cup 3.0 - Finals (online version)

A1. Oh Sweet Beaverette

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

— Oh my sweet Beaverette, would you fancy a walk along a wonderful woodland belt with me?

— Of course, my Smart Beaver! Let us enjoy the splendid view together. How about Friday night?

At this point the Smart Beaver got rushing. Everything should be perfect by Friday, so he needed to prepare the belt to the upcoming walk. He needed to cut down several trees.

Let's consider the woodland belt as a sequence of trees. Each tree i is described by the esthetic appeal a_i — some trees are very esthetically pleasing, others are 'so-so', and some trees are positively ugly!

The Smart Beaver calculated that he needed the following effects to win the Beaverette's heart:

- The first objective is to please the Beaverette: the sum of esthetic appeal of the remaining trees must be maximum possible;
- the second objective is to surprise the Beaverette: the esthetic appeal of the first and the last trees in the resulting belt must be the same;
- and of course, the walk should be successful: there must be at least two trees in the woodland belt left.

Now help the Smart Beaver! Which trees does he need to cut down to win the Beaverette's heart?

Input

The first line contains a single integer n — the initial number of trees in the woodland belt, $2 \le n$. The second line contains space-separated integers a_i — the esthetic appeals of each tree. All esthetic appeals do not exceed 10^9 in their absolute value.

- to get 30 points, you need to solve the problem with constraints: $n \le 100$ (subproblem A1);
- to get 100 points, you need to solve the problem with constraints: $n \le 3 \cdot 10^5$ (subproblems A1+A2).

Output

In the first line print two integers — the total esthetic appeal of the woodland belt after the Smart Beaver's intervention and the number of the cut down trees k.

In the next line print k integers — the numbers of the trees the Beaver needs to cut down. Assume that the trees are numbered from 1 to n from left to right.

If there are multiple solutions, print any of them. It is guaranteed that at least two trees have equal esthetic appeal.

Sample test(s)

input
5 1 2 3 1 2
output
8 1 1

input
5 1 -2 3 1 -2
output
5 2 2 5

A2. Oh Sweet Beaverette

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

— Oh my sweet Beaverette, would you fancy a walk along a wonderful woodland belt with me?

— Of course, my Smart Beaver! Let us enjoy the splendid view together. How about Friday night?

At this point the Smart Beaver got rushing. Everything should be perfect by Friday, so he needed to prepare the belt to the upcoming walk. He needed to cut down several trees.

Let's consider the woodland belt as a sequence of trees. Each tree i is described by the esthetic appeal a_i — some trees are very esthetically pleasing, others are 'so-so', and some trees are positively ugly!

The Smart Beaver calculated that he needed the following effects to win the Beaverette's heart:

- The first objective is to please the Beaverette: the sum of esthetic appeal of the remaining trees must be maximum possible;
- the second objective is to surprise the Beaverette: the esthetic appeal of the first and the last trees in the resulting belt must be the same;
- and of course, the walk should be successful: there must be at least two trees in the woodland belt left.

Now help the Smart Beaver! Which trees does he need to cut down to win the Beaverette's heart?

Input

The first line contains a single integer n — the initial number of trees in the woodland belt, $2 \le n$. The second line contains space-separated integers a_i — the esthetic appeals of each tree. All esthetic appeals do not exceed 10^9 in their absolute value.

- to get 30 points, you need to solve the problem with constraints: $n \le 100$ (subproblem A1);
- to get 100 points, you need to solve the problem with constraints: $n \le 3 \cdot 10^5$ (subproblems A1+A2).

Output

In the first line print two integers — the total esthetic appeal of the woodland belt after the Smart Beaver's intervention and the number of the cut down trees k.

In the next line print k integers — the numbers of the trees the Beaver needs to cut down. Assume that the trees are numbered from 1 to n from left to right.

If there are multiple solutions, print any of them. It is guaranteed that at least two trees have equal esthetic appeal.

Sample test(s)

input
5 1 2 3 1 2
output
8 1 1

input
5 1 -2 3 1 -2
output
5 2 2 5

B1. Shave Beaver!

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Smart Beaver has recently designed and built an innovative nanotechnologic all-purpose beaver mass shaving machine, "Beavershave 5000". Beavershave 5000 can shave beavers by families! How does it work? Very easily!

There are n beavers, each of them has a unique id from 1 to n. Consider a permutation $a_1, a_2, ..., a_n$ of n these beavers. Beavershave 5000 needs one session to shave beavers with ids from x to y (inclusive) if and only if there are such indices $i_1 < i_2 < ... < i_k$, that $a_{i_1} = x$, $a_{i_2} = x + 1$, ..., $a_{i_{k-1}} = y - 1$, $a_{i_k} = y$. And that is really convenient. For example, it needs one session to shave a permutation of beavers $1, 2, 3, ..., n$.

If we can't shave beavers from x to y in one session, then we can split these beavers into groups $[x, p_1]$, $[p_1 + 1, p_2]$, ..., $[p_m + 1, y]$ $(x \le p_1 < p_2 < ... < p_m < y)$, in such a way that the machine can shave beavers in each group in one session. But then Beavershave 5000 needs $m + 1$ working sessions to shave beavers from x to y.

All beavers are restless and they keep trying to swap. So if we consider the problem more formally, we can consider queries of two types:

- what is the minimum number of sessions that Beavershave 5000 needs to shave beavers with ids from x to y, inclusive?
- two beavers on positions x and y (the beavers a_x and a_y) swapped.

You can assume that any beaver can be shaved any number of times.

Input

The first line contains integer n — the total number of beavers, $2 \le n$. The second line contains n space-separated integers — the initial beaver permutation.

The third line contains integer q — the number of queries, $1 \le q \le 10^5$. The next q lines contain the queries. Each query i looks as $p_i\ x_i\ y_i$, where p_i is the query type (1 is to shave beavers from x_i to y_i, inclusive, 2 is to swap beavers on positions x_i and y_i). All queries meet the condition: $1 \le x_i < y_i \le n$.

- to get 30 points, you need to solve the problem with constraints: $n \le 100$ (subproblem B1);
- to get 100 points, you need to solve the problem with constraints: $n \le 3 \cdot 10^5$ (subproblems B1+B2).

Note that the number of queries q is limited $1 \le q \le 10^5$ in both subproblem B1 and subproblem B2.

Output

For each query with $p_i = 1$, print the minimum number of Beavershave 5000 sessions.

Sample test(s)

input
5 1 3 4 2 5 6 1 1 5 1 3 4 2 2 3 1 1 5 2 1 5 1 1 5

output
2 1 3 5

B2. Shave Beaver!

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Smart Beaver has recently designed and built an innovative nanotechnologic all-purpose beaver mass shaving machine, "Beavershave 5000". Beavershave 5000 can shave beavers by families! How does it work? Very easily!

There are n beavers, each of them has a unique id from 1 to n. Consider a permutation $a_1, a_2, ..., a_n$ of n these beavers. Beavershave 5000 needs one session to shave beavers with ids from x to y (inclusive) if and only if there are such indices $i_1 < i_2 < ... < i_k$, that $a_{i_1} = x$, $a_{i_2} = x + 1$, ..., $a_{i_{k-1}} = y - 1$, $a_{i_k} = y$. And that is really convenient. For example, it needs one session to shave a permutation of beavers $1, 2, 3, ..., n$.

If we can't shave beavers from x to y in one session, then we can split these beavers into groups $[x, p_1]$, $[p_1 + 1, p_2]$, ..., $[p_m + 1, y]$ $(x \le p_1 < p_2 < ... < p_m < y)$, in such a way that the machine can shave beavers in each group in one session. But then Beavershave 5000 needs $m + 1$ working sessions to shave beavers from x to y.

All beavers are restless and they keep trying to swap. So if we consider the problem more formally, we can consider queries of two types:

- what is the minimum number of sessions that Beavershave 5000 needs to shave beavers with ids from x to y, inclusive?
- two beavers on positions x and y (the beavers a_x and a_y) swapped.

You can assume that any beaver can be shaved any number of times.

Input

The first line contains integer n — the total number of beavers, $2 \le n$. The second line contains n space-separated integers — the initial beaver permutation.

The third line contains integer q — the number of queries, $1 \le q \le 10^5$. The next q lines contain the queries. Each query i looks as $p_i\ x_i\ y_i$, where p_i is the query type (1 is to shave beavers from x_i to y_i, inclusive, 2 is to swap beavers on positions x_i and y_i). All queries meet the condition: $1 \le x_i < y_i \le n$.

- to get 30 points, you need to solve the problem with constraints: $n \le 100$ (subproblem B1);
- to get 100 points, you need to solve the problem with constraints: $n \le 3 \cdot 10^5$ (subproblems B1+B2).

Note that the number of queries q is limited $1 \le q \le 10^5$ in both subproblem B1 and subproblem B2.

Output

For each query with $p_i = 1$, print the minimum number of Beavershave 5000 sessions.

Sample test(s)

input
5 1 3 4 2 5 6 1 1 5 1 3 4 2 2 3 1 1 5 2 1 5 1 1 5

output
2 1 3 5

C1. The Great Julya Calendar

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Yet another Armageddon is coming! This time the culprit is the Julya tribe calendar.

The beavers in this tribe knew math very well. Smart Beaver, an archaeologist, got a sacred plate with a magic integer on it. The translation from Old Beaverish is as follows:

"*May the Great Beaver bless you! May your chacres open and may your third eye never turn blind from beholding the Truth! Take the magic number, subtract a digit from it (the digit must occur in the number) and get a new magic number. Repeat this operation until a magic number equals zero. The Earth will stand on Three Beavers for the time, equal to the number of subtractions you perform!*"

Distinct subtraction sequences can obviously get you different number of operations. But the Smart Beaver is ready to face the worst and is asking you to count the minimum number of operations he needs to reduce the magic number to zero.

Input

The single line contains the magic integer n, $0 \le n$.

- to get 20 points, you need to solve the problem with constraints: $n \le 10^6$ (subproblem C1);
- to get 40 points, you need to solve the problem with constraints: $n \le 10^{12}$ (subproblems C1+C2);
- to get 100 points, you need to solve the problem with constraints: $n \le 10^{18}$ (subproblems C1+C2+C3).

Output

Print a single integer — the minimum number of subtractions that turns the magic number to a zero.

Sample test(s)

input
24
output
5

Note

In the first test sample the minimum number of operations can be reached by the following sequence of subtractions:

$$24 \to 20 \to 18 \to 10 \to 9 \to 0$$

C2. The Great Julya Calendar

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Yet another Armageddon is coming! This time the culprit is the Julya tribe calendar.

The beavers in this tribe knew math very well. Smart Beaver, an archaeologist, got a sacred plate with a magic integer on it. The translation from Old Beaverish is as follows:

"*May the Great Beaver bless you! May your chacres open and may your third eye never turn blind from beholding the Truth! Take the magic number, subtract a digit from it (the digit must occur in the number) and get a new magic number. Repeat this operation until a magic number equals zero. The Earth will stand on Three Beavers for the time, equal to the number of subtractions you perform!*"

Distinct subtraction sequences can obviously get you different number of operations. But the Smart Beaver is ready to face the worst and is asking you to count the minimum number of operations he needs to reduce the magic number to zero.

Input

The single line contains the magic integer n, $0 \le n$.

- to get 20 points, you need to solve the problem with constraints: $n \le 10^6$ (subproblem C1);
- to get 40 points, you need to solve the problem with constraints: $n \le 10^{12}$ (subproblems C1+C2);
- to get 100 points, you need to solve the problem with constraints: $n \le 10^{18}$ (subproblems C1+C2+C3).

Output

Print a single integer — the minimum number of subtractions that turns the magic number to a zero.

Sample test(s)

input
24
output
5

Note

In the first test sample the minimum number of operations can be reached by the following sequence of subtractions:

$$24 \to 20 \to 18 \to 10 \to 9 \to 0$$

C3. The Great Julya Calendar

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Yet another Armageddon is coming! This time the culprit is the Julya tribe calendar.

The beavers in this tribe knew math very well. Smart Beaver, an archaeologist, got a sacred plate with a magic integer on it. The translation from Old Beaverish is as follows:

"*May the Great Beaver bless you! May your chacres open and may your third eye never turn blind from beholding the Truth! Take the magic number, subtract a digit from it (the digit must occur in the number) and get a new magic number. Repeat this operation until a magic number equals zero. The Earth will stand on Three Beavers for the time, equal to the number of subtractions you perform!*"

Distinct subtraction sequences can obviously get you different number of operations. But the Smart Beaver is ready to face the worst and is asking you to count the minimum number of operations he needs to reduce the magic number to zero.

Input

The single line contains the magic integer n, $0 \le n$.

- to get 20 points, you need to solve the problem with constraints: $n \le 10^6$ (subproblem C1);
- to get 40 points, you need to solve the problem with constraints: $n \le 10^{12}$ (subproblems C1+C2);
- to get 100 points, you need to solve the problem with constraints: $n \le 10^{18}$ (subproblems C1+C2+C3).

Output

Print a single integer — the minimum number of subtractions that turns the magic number to a zero.

Sample test(s)

input
24
output
5

Note

In the first test sample the minimum number of operations can be reached by the following sequence of subtractions:

$$24 \to 20 \to 18 \to 10 \to 9 \to 0$$

D1. Escaping on Beaveractor

time limit per test: 6 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Don't put up with what you're sick of! The Smart Beaver decided to escape from the campus of Beaver Science Academy (BSA). BSA is a $b \times b$ square on a plane. Each point x, y $(0 \le x, y \le b)$ belongs to BSA. To make the path quick and funny, the Beaver constructed a Beaveractor, an effective and comfortable types of transport.

The campus obeys traffic rules: there are n arrows, parallel to the coordinate axes. The arrows do not intersect and do not touch each other. When the Beaveractor reaches some arrow, it turns in the arrow's direction and moves on until it either reaches the next arrow or gets outside the campus. The Beaveractor covers exactly one unit of space per one unit of time. You can assume that there are no obstacles to the Beaveractor.

The BSA scientists want to transport the brand new Beaveractor to the "Academic Tractor" research institute and send the Smart Beaver to do his postgraduate studies and sharpen pencils. They have q plans, representing the Beaveractor's initial position (x_i, y_i), the initial motion vector w_i and the time t_i that have passed after the escape started.

Your task is for each of the q plans to determine the Smart Beaver's position after the given time.

Input

The first line contains two integers: the number of traffic rules n and the size of the campus b, $0 \le n$, $1 \le b$. Next n lines contain the rules. Each line of the rules contains four space-separated integers x_0, y_0, x_1, y_1 — the beginning and the end of the arrow. It is guaranteed that all arrows are parallel to the coordinate axes and have no common points. All arrows are located inside the campus, that is, $0 \le x_0, y_0, x_1, y_1 \le b$ holds.

Next line contains integer q — the number of plans the scientists have, $1 \le q \le 10^5$. The i-th plan is represented by two integers, x_i, y_i are the Beaveractor's coordinates at the initial time, $0 \le x_i, y_i \le b$, character w_i, that takes value `U`, `D`, `L`, `R` and sets the initial direction up, down, to the left or to the right correspondingly (the Y axis is directed upwards), and t_i — the time passed after the escape started, $0 \le t_i \le 10^{15}$.

- to get 30 points you need to solve the problem with constraints $n, b \le 30$ (subproblem D1);
- to get 60 points you need to solve the problem with constraints $n, b \le 1000$ (subproblems D1+D2);
- to get 100 points you need to solve the problem with constraints $n, b \le 10^5$ (subproblems D1+D2+D3).

Output

Print q lines. Each line should contain two integers — the Beaveractor's coordinates at the final moment of time for each plan. If the Smart Beaver manages to leave the campus in time t_i, print the coordinates of the last point in the campus he visited.

Sample test(s)

input

```
3 3
0 0 0 1
0 2 2 2
3 3 2 3
12
0 0 L 0
0 0 L 1
0 0 L 2
0 0 L 3
0 0 L 4
0 0 L 5
0 0 L 6
2 0 U 2
2 0 U 3
3 0 U 5
1 3 D 2
1 3 R 2
```

output

```
0 0
0 1
0 2
1 2
2 2
3 2
3 2
2 2
3 2
1 3
2 2
1 3
```

D2. Escaping on Beaveractor

time limit per test: 6 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Don't put up with what you're sick of! The Smart Beaver decided to escape from the campus of Beaver Science Academy (BSA). BSA is a $b \times b$ square on a plane. Each point x, y $(0 \le x, y \le b)$ belongs to BSA. To make the path quick and funny, the Beaver constructed a Beaveractor, an effective and comfortable types of transport.

The campus obeys traffic rules: there are n arrows, parallel to the coordinate axes. The arrows do not intersect and do not touch each other. When the Beaveractor reaches some arrow, it turns in the arrow's direction and moves on until it either reaches the next arrow or gets outside the campus. The Beaveractor covers exactly one unit of space per one unit of time. You can assume that there are no obstacles to the Beaveractor.

The BSA scientists want to transport the brand new Beaveractor to the "Academic Tractor" research institute and send the Smart Beaver to do his postgraduate studies and sharpen pencils. They have q plans, representing the Beaveractor's initial position (x_i, y_i), the initial motion vector w_i and the time t_i that have passed after the escape started.

Your task is for each of the q plans to determine the Smart Beaver's position after the given time.

Input

The first line contains two integers: the number of traffic rules n and the size of the campus b, $0 \le n$, $1 \le b$. Next n lines contain the rules. Each line of the rules contains four space-separated integers x_0, y_0, x_1, y_1 — the beginning and the end of the arrow. It is guaranteed that all arrows are parallel to the coordinate axes and have no common points. All arrows are located inside the campus, that is, $0 \le x_0, y_0, x_1, y_1 \le b$ holds.

Next line contains integer q — the number of plans the scientists have, $1 \le q \le 10^5$. The i-th plan is represented by two integers, x_i, y_i are the Beaveractor's coordinates at the initial time, $0 \le x_i, y_i \le b$, character w_i, that takes value `U`, `D`, `L`, `R` and sets the initial direction up, down, to the left or to the right correspondingly (the Y axis is directed upwards), and t_i — the time passed after the escape started, $0 \le t_i \le 10^{15}$.

- to get 30 points you need to solve the problem with constraints $n, b \le 30$ (subproblem D1);
- to get 60 points you need to solve the problem with constraints $n, b \le 1000$ (subproblems D1+D2);
- to get 100 points you need to solve the problem with constraints $n, b \le 10^5$ (subproblems D1+D2+D3).

Output

Print q lines. Each line should contain two integers — the Beaveractor's coordinates at the final moment of time for each plan. If the Smart Beaver manages to leave the campus in time t_i, print the coordinates of the last point in the campus he visited.

Sample test(s)

input

```
3 3
0 0 0 1
0 2 2 2
3 3 2 3
12
0 0 L 0
0 0 L 1
0 0 L 2
0 0 L 3
0 0 L 4
0 0 L 5
0 0 L 6
2 0 U 2
2 0 U 3
3 0 U 5
1 3 D 2
1 3 R 2
```

output

```
0 0
0 1
0 2
1 2
2 2
3 2
3 2
2 2
3 2
1 3
2 2
1 3
```

D3. Escaping on Beaveractor

time limit per test: 6 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Don't put up with what you're sick of! The Smart Beaver decided to escape from the campus of Beaver Science Academy (BSA). BSA is a $b \times b$ square on a plane. Each point x, y $(0 \le x, y \le b)$ belongs to BSA. To make the path quick and funny, the Beaver constructed a Beaveractor, an effective and comfortable types of transport.

The campus obeys traffic rules: there are n arrows, parallel to the coordinate axes. The arrows do not intersect and do not touch each other. When the Beaveractor reaches some arrow, it turns in the arrow's direction and moves on until it either reaches the next arrow or gets outside the campus. The Beaveractor covers exactly one unit of space per one unit of time. You can assume that there are no obstacles to the Beaveractor.

The BSA scientists want to transport the brand new Beaveractor to the "Academic Tractor" research institute and send the Smart Beaver to do his postgraduate studies and sharpen pencils. They have q plans, representing the Beaveractor's initial position (x_i, y_i), the initial motion vector w_i and the time t_i that have passed after the escape started.

Your task is for each of the q plans to determine the Smart Beaver's position after the given time.

Input

The first line contains two integers: the number of traffic rules n and the size of the campus b, $0 \le n$, $1 \le b$. Next n lines contain the rules. Each line of the rules contains four space-separated integers x_0, y_0, x_1, y_1 — the beginning and the end of the arrow. It is guaranteed that all arrows are parallel to the coordinate axes and have no common points. All arrows are located inside the campus, that is, $0 \le x_0, y_0, x_1, y_1 \le b$ holds.

Next line contains integer q — the number of plans the scientists have, $1 \le q \le 10^5$. The i-th plan is represented by two integers, x_i, y_i are the Beaveractor's coordinates at the initial time, $0 \le x_i, y_i \le b$, character w_i, that takes value `U`, `D`, `L`, `R` and sets the initial direction up, down, to the left or to the right correspondingly (the Y axis is directed upwards), and t_i — the time passed after the escape started, $0 \le t_i \le 10^{15}$.

- to get 30 points you need to solve the problem with constraints $n, b \le 30$ (subproblem D1);
- to get 60 points you need to solve the problem with constraints $n, b \le 1000$ (subproblems D1+D2);
- to get 100 points you need to solve the problem with constraints $n, b \le 10^5$ (subproblems D1+D2+D3).

Output

Print q lines. Each line should contain two integers — the Beaveractor's coordinates at the final moment of time for each plan. If the Smart Beaver manages to leave the campus in time t_i, print the coordinates of the last point in the campus he visited.

Sample test(s)

input

```
3 3
0 0 0 1
0 2 2 2
3 3 2 3
12
0 0 L 0
0 0 L 1
0 0 L 2
0 0 L 3
0 0 L 4
0 0 L 5
0 0 L 6
2 0 U 2
2 0 U 3
3 0 U 5
1 3 D 2
1 3 R 2
```

output

```
0 0
0 1
0 2
1 2
2 2
3 2
3 2
2 2
3 2
1 3
2 2
1 3
```

E1. Deja Vu

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Everybody knows that we have been living in the Matrix for a long time. And in the new seventh Matrix the world is ruled by beavers.

So let's take beaver Neo. Neo has so-called "deja vu" outbursts when he gets visions of events in some places he's been at or is going to be at. Let's examine the phenomenon in more detail.

We can say that Neo's city is represented by a directed graph, consisting of n shops and m streets that connect the shops. No two streets connect the same pair of shops (besides, there can't be one street from A to B and one street from B to A). No street connects a shop with itself. As Neo passes some streets, he gets visions. No matter how many times he passes street k, every time he will get the same visions in the same order. A vision is a sequence of shops.

We know that Neo is going to get really shocked if he passes the way from some shop a to some shop b, possible coinciding with a, such that the list of visited shops in the real life and in the visions coincide.

Suggest beaver Neo such path of non-zero length. Or maybe you can even count the number of such paths modulo 1000000007 $(10^9 + 7)$?..

Input

The first line contains integers n and m — the number of shops and the number of streets, correspondingly, $1 \le n \le 50$, $0 \le m \le \frac{n(n-1)}{2}$. Next m lines contain the descriptions of the streets in the following format: x_i y_i k_i v_1 v_2 ... v_k, where x_i and y_i $(1 \le x_i, y_i \le n, x_i \ne y_i)$ are indices of shops connected by a street, k_i $(0 \le k_i \le n)$ is the number of visions on the way from x_i to y_i; v_1, v_2, ..., v_k $(1 \le v_i \le n)$ describe the visions: the numbers of the shops Neo saw. Note that the order of the visions matters.

It is guaranteed that the total number of visions on all streets doesn't exceed 10^5.

- to get 50 points, you need to find any (not necessarily simple) path of length at most $2 \cdot n$, that meets the attributes described above (subproblem E1);
- to get 50 more points, you need to count for each length from 1 to $2 \cdot n$ the number of paths that have the attribute described above (subproblem E2).

Output

Subproblem E1. In the first line print an integer k $(1 \le k \le 2 \cdot n)$ — the numbers of shops on Neo's path. In the next line print k integers — the number of shops in the order Neo passes them. If the graph doesn't have such paths or the length of the shortest path includes more than $2 \cdot n$ shops, print on a single line 0.

Subproblem E2. Print $2 \cdot n$ lines. The i-th line must contain a single integer — the number of required paths of length i modulo 1000000007 $(10^9 + 7)$.

Sample test(s)

input

```
6 6
1 2 2 1 2
2 3 1 3
3 4 2 4 5
4 5 0
5 3 1 3
6 1 1 6
```

output

```
4
6 1 2 3
```

input

```
6 6
1 2 2 1 2
2 3 1 3
3 4 2 4 5
4 5 0
5 3 1 3
6 1 1 6
```

output

```
1
2
1
1
2
1
1
2
1
1
2
```

```
1
```

Note

The input in both samples are the same. The first sample contains the answer to the first subproblem, the second sample contains the answer to the second subproblem.

E2. Deja Vu

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Everybody knows that we have been living in the Matrix for a long time. And in the new seventh Matrix the world is ruled by beavers.

So let's take beaver Neo. Neo has so-called "deja vu" outbursts when he gets visions of events in some places he's been at or is going to be at. Let's examine the phenomenon in more detail.

We can say that Neo's city is represented by a directed graph, consisting of n shops and m streets that connect the shops. No two streets connect the same pair of shops (besides, there can't be one street from A to B and one street from B to A). No street connects a shop with itself. As Neo passes some streets, he gets visions. No matter how many times he passes street k, every time he will get the same visions in the same order. A vision is a sequence of shops.

We know that Neo is going to get really shocked if he passes the way from some shop a to some shop b, possible coinciding with a, such that the list of visited shops in the real life and in the visions coincide.

Suggest beaver Neo such path of non-zero length. Or maybe you can even count the number of such paths modulo 1000000007 $(10^9 + 7)$?..

Input

The first line contains integers n and m — the number of shops and the number of streets, correspondingly, $1 \le n \le 50$, $0 \le m \le \frac{n(n-1)}{2}$. Next m lines contain the descriptions of the streets in the following format: x_i y_i k_i v_1 v_2 ... v_k, where x_i and y_i $(1 \le x_i, y_i \le n, x_i \ne y_i)$ are numbers of shops connected by a street, k_i $(0 \le k_i \le n)$ is the number of visions on the way from x_i to y_i; v_1, v_2, ..., v_k $(1 \le v_i \le n)$ describe the visions: the numbers of the shops Neo saw. Note that the order of the visions matters.

It is guaranteed that the total number of visions on all streets doesn't exceed 10^5.

- to get 50 points, you need to find any (not necessarily simple) path of length at most $2 \cdot n$, that meets the attributes described above (subproblem E1);
- to get 50 more points, you need to count for each length from 1 to $2 \cdot n$ the number of paths that have the attribute described above (subproblem E2).

Output

Subproblem E1. In the first line print an integer k $(1 \le k \le 2 \cdot n)$ — the numbers of shops on Neo's path. In the next line print k integers — the number of shops in the order Neo passes them. If the graph doesn't have such paths or the length of the shortest path includes more than $2 \cdot n$ shops, print on a single line 0.

Subproblem E2. Print $2 \cdot n$ lines. The i-th line must contain a single integer — the number of required paths of length i modulo 1000000007 $(10^9 + 7)$.

Sample test(s)

input

```
6 6
1 2 2 1 2
2 3 1 3
3 4 2 4 5
4 5 0
5 3 1 3
6 1 1 6
```

output

```
4
6 1 2 3
```

input

```
6 6
1 2 2 1 2
2 3 1 3
3 4 2 4 5
4 5 0
5 3 1 3
6 1 1 6
```

output

```
1
2
1
1
2
1
1
2
1
1
2
```

```
1
```

Note

The input in both samples are the same. The first sample contains the answer to the first subproblem, the second sample contains the answer to the second subproblem.
