Problem A: **Rotating Display**

Wendy is finishing her summer job in the lab testing a new 3D printed robot which is being taught to manipulate small objects.

To test the robot's abilities, a simple device called the display is used. It is a thin transparent square array of $N \times N$ square slots. Each slot contains a token, in the shape of an ASCII character (for better recognition), which is held in place in the slot by small magnets. The display can be rotated by 90° around the axis perpendicular to its surface or flipped by 180° around one of its four axes parallel to the display surface.

The robot simulates display rotations and flips by the following process. It removes the tokens from their slots and puts them back into different slots so that the contents of the display looks exactly as if the whole display was rotated or flipped. If, for achieving the desired effect, it is necessary to rotate or to flip particular tokens in their new positions the robot does it as well. The display remains stationary during the whole process.

For example, if the upper left corner of the display contains a token which looks like symbol “<” (less than) then after flipping the display around the vertical axis this token is moved to the upper right corner where it looks like the symbol “>” (greater than). Then, after left rotation the same token is moved back to the upper left corner where it looks like the symbol “^” (caret).

Wendy has programmed the robot to perform a long sequence of successive flips and rotations. To check the correctness of the robot's algorithms, she needs to know in advance how the display should look when the robot finishes its work.

Input Specification

We suppose that a token on the display can be shaped as any of the following so called symmetric characters: “<”, “>”, “^”, “v”, “o”, “x”, “|”, “-”, “/”, “\”. When a symmetric character is rotated or flipped it either remains the same or it becomes another symmetric character whose shape is the most similar to the rotated/flipped one.

There are more test cases. Each case starts with a line containing one integer N ($1 \le N \le 100$). Next, there are N lines representing the initial state of the display. Each line contains a string which consists of exactly N symmetric characters. Each character represents one slot on the display and the order of symbols in the input corresponds to the order of the tokens on the display. No slot on the display is empty. After N lines, there is a line with a command string specifying the flips and rotations that has to be carried out. A command string consists of command characters, each command character specifies one rotation or flip as follows: “<” (rotation left), “>” (rotation right), “-” (flip around horizontal axis), “|” (flip around vertical axis), “\” (flip around main diagonal), “/” (flip around anti-diagonal). Two successive command characters are separated by single space. The robot has to respect the order of commands in the command string. The number of commands is always positive and at most 10^6.

The decimal ASCII codes of the characters relevant to this problem are 45 (“-”), 47 (“/”), 60 (“<”), 62 (“>”), 92 (“\”), 94 (“^”), 111 (“o”), 118 (“v”), 120 (“x”), 124 (“|”).

Output Specification

For each test case, print N lines specifying the final position and orientation of the tokens on the display. The output format of the display representation is identical to the input format except for the size of the display which should not be printed.

Sample Input

```
3
o^-
/v|
vx^
< |
5
x>-o\
voooo
|ooo/
ooo/v
\o/vv
| \ |
```

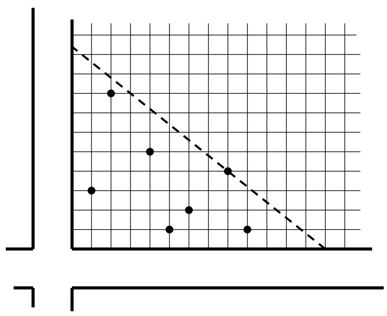
Output for Sample Input

```
>-|
x<>
</o
<</o\
</ooo
/ooo|
oooo^
\o-<x
```

Problem B: Cable Connection

Two straight roads A and B perpendicular to each other start at the common crossing. Road A runs eastward and road B runs northward.

The roads are part of a large industrial system being built in the area and they should be connected by a special high frequency cable. Unfortunately, the connection cannot be built directly at the intersection. Additionally, there are some buildings standing in the corner of the field bordered by the roads. The buildings represent obstacles to the cable.



After some negotiations and considering various technical limitations, the analysts of the cable company constructed a set of critical points, determined by the obstacles. Then they suggested that the cable should connect the roads in a way that satisfies the following criteria:

- The cable should run in a straight line.
- The cable should not run between any critical point and the corner of the field at the roads crossing.
- The length of the cable should be the minimum possible.

Now, your task is to determine the length of the cable.

Input Specification

There are more test cases. Each case starts with a line containing one integer N $(1 \leq N \leq 10^6)$ which specifies the number of critical points. Next, there are N lines representing the points, each line describes one of them. A point P is represented by two integers a, b $(1 \leq a, b \leq 10\,000)$ separated by space. Integer a is the distance from P to road A and integer b is the distance from P to road B. All distances are in meters. You may suppose that the lengths of the roads and also the size of the field are not limited. All coordinate pairs (a, b) in one test case are unique.

Output Specification

For each test case, print a single line with one floating point number L denoting the minimum possible length of the cable expressed in meters. L should be printed with the maximum allowed error of 10^{-3}.

Sample Input

```
7
5 1
9 1
6 2
1 3
8 4
4 5
2 8
```

Output for Sample Input

```
16.648
```

Problem C: **Orchard Division**

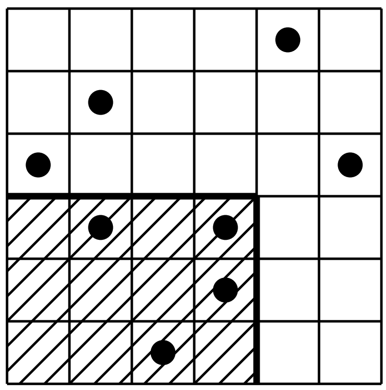
Uncle Oliver is going to sell a significant part of his famous dwarf plum tree orchard. He is going to divide the orchard into two parts, sell the first one and keep the other one.

The trees were originally planted in regular rows and columns forming a rectangular grid with the same number of rows and columns. As years went by, Oliver removed many trees which were weak or plagued by bugs so nowadays there is also a lot of free squares unoccupied by any tree.

Oliver has decided that he will keep exactly half of all the trees in the orchard. Moreover, he has few additional demands which, in his opinion, will ensure easy maintenance of his part in the future.

- The part Oliver is going to keep should be in the shape of a rectangle.
- A least one corner of the rectangle should coincide with a corner of the orchard.
- The rectangle area should be as small as possible.

Originally, each tree was planted in the center of an imaginary square whose area was exactly one square meter. Thus, the position of each tree can be described by the coordinates of the square on which it is standing. The dividing fence between the two parts of the orchard will run along the borders of the squares.



Input Specification

There are more test cases. Each case starts with a line containing two integers M $(1 \leq M \leq 10^9)$ and N $(1 \leq N \leq 10^6)$ separated by space. The orchard side length in meters is expressed by M and the number of trees in the orchard is expressed by N. Next, there are N lines, each line specifies x and y coordinates of one tree in the orchard. The coordinates are separated by space. For simplicity reasons, we assume that the coordinates are zero based, so the coordinates of the squares in the corners of the orchard are $(0,0), (0,M-1), (M-1,M-1), (M-1,0)$. All coordinate pairs (x,y) in one test case are unique.

Output Specification

For each test case, print a single line with one whole number A denoting the minimum possible area in square meters of uncle Oliver's part of the orchard. If it is not possible to divide the orchard according to Oliver's demands print "-1". Note that the output value might not fit into 32-bit integer type.

Sample Input

```
6 8
4 5
1 4
0 3
5 3
1 2
3 2
3 1
2 0
3 3
2 0
1 1
0 2
2 2
0 0
1 1
```

Output for Sample Input

```
12
-1
1
```

Problem D: Aerial Archeology

Andrew is on a summer vacation job with a group of aerial archeologists. The group is internationally known for their advances in using nuclear imaging spectroscopy to investigate the underground remains of prehistoric cultures. Today, Andrew's job is to find a route for a helicopter which will carry the spectrometer over the area of archeological interest in the nearby lowlands. The spectrometer is a very sensitive and vulnerable device and the helicopter carrying it has to fly at constant speed in a perfectly straight line to minimize the measurement noise.

Hidden under the surface in the lowlands, there are more prehistoric settlements whose location and boundaries have been previously established by other techniques. All settlement boundaries are drawn on a special map which is at Andrew's disposal. The goal of the flight is to fly over as many settlements as possible and measure the soil composition in and around them. Thus, all Andrew has to do is to draw such straight line on the map that intersects the maximum number of settlements drawn there.

The shapes of the settlements are complicated and the settlements overlap, often chaotically. So it is not immediately obvious where to draw the line.

Input Specification

The input describes the shapes and the positions of settlements on the map. Each settlement is represented as a simple polygon (no two of its non-adjacent boundary segments touch or intersect each other). The polygons may overlap one another.

There are more test cases in the input. Each case starts with a line containing one positive integer N which specifies the number of polygons on the map. Then there is the description of N polygons. Each polygon description starts with one text line containing single integer M ($M \geq 3$) which denotes the number of vertices of the polygon. The next M lines specify the vertices of the polygon. Each of these lines specifies one vertex by its two coordinates x, y separated by space. The vertices are listed in the clockwise direction along the polygon boundary. All coordinates are integers with an absolute value at most 10 000. The total number of vertices of all polygons on the map does not exceed 1 000.

Output Specification

For each test case, print a single line with integer P denoting the maximum number of polygons on the map which can be intersected by a straight line. Note that only the intersections of the line with the *interior* of the polygons are considered.

Sample Input

```
3
4
0 0
0 1
1 1
1 0
4
1 2
1 3
2 3
2 2
5
2 1
2 2
9 2
10 3
10 1
```

Output for Sample Input

```
2
```

Problem E: Tree Stands

Tree stands are elevated wooden platforms attached to trees. Typically, huntsmen use tree stands to watch or to shoot their prey.

In our county, huntsmen have built a remarkable system of tree stands. The tree stands are connected by narrow straight paths which form a kind of maze on the hunting grounds. The builders wanted to minimize the impact on the environment and so they built the minimum possible number of paths which ensure that there is a connection between any two tree stands. Also, a tree stand is visible from another stand if and only if the two are connected by a path.

A group of local huntsmen wants to find out which particular tree stands will serve the best their hunting interests. Each day they climb a different set of tree stands and watch the wildlife.

There are a few more important circumstances to consider:

- Security rules dictate that any occupied tree stand must be visible from at least one other occupied tree stand so that in case of an emergency the huntsman in the neighbour tree stand can come to help the colleague.
- A tree stand is always occupied by at most one huntsman.
- It does not matter which huntsman is in which tree stand. It only matters which tree stands are occupied and which are not.
- The size of the group does not change.

How many days will the group spend in the tree stands before they investigate all possible choices of tree stands available to them?

Input Specification

There are more test cases. Each case starts with a line containing two integers N and K ($2 \leq K \leq N \leq 200$) separated by space. N is the number of tree stands, K is the size of the group of huntsmen. The tree stands are labeled $1, 2, 3, \ldots, N$.

Next, there are $N - 1$ lines, each line specifies one path between two tree stands. The line contains the labels of the stands separated by a space. The order of the labels on a line and the order of the paths in the input is arbitrary.

Output Specification

For each test case, print on a separate line the number of days which the group will spend in the tree stands. Express the result modulo 1 000 000 007.

Sample Input

```
4 3
1 2
1 4
1 3
5 4
1 2
2 3
3 4
4 5
```

Output for Sample Input

```
3
3
```