

Problem 4. Park trails

Input file: `input.txt`
Output file: `output.txt`
Time limit: 3 seconds
 6 seconds (for Java)
Memory limit: 256 megabytes

The Forest Frontiers theme park was recently inspected by the Federal Emergency Management Agency. The most serious problem revealed was the layout of trails in the park. Help the park management fix this problem.

The park map can be easily represented on the coordinate plane. Each trail in the park is a line segment parallel to one of the coordinate axes. Any two trails don't have common points, with one exception: end points of segments may coincide. An evacuation point is located somewhere in the park. This point is guaranteed either to lie off all the trails or to coincide with an end point of a trail.

In case of an emergency, an alarm is switched on, and visitors are told to proceed to the evacuation point. According to the FEMA rules, any visitor must be able to reach the evacuation point moving only along trails, with an additional constraint that the distance to the evacuation point must decrease monotonously during movement. It is assumed that park visitors never leave the trails, so each visitor can be located at arbitrary point of arbitrary trail when the alarm is triggered.

Obviously, no one took this silly regulation into account when planning the park layout.

At the moment, the above-ground part of the park is finished, and it's impossible to build new trails above ground. However, it's possible to build some trails below ground, in tunnels. Underground trails are treated the same way as regular trails in FEMA regulations. The difference in elevation is **not** taken into account when calculating the distance to the evacuation point.

We can safely assume that each tunnel consists of a set of underground trails. Additional FEMA rules apply to tunnels:

1. A tunnel cannot be forked: there are strictly two entrances and a linear path between them.
2. Each tunnel entrance must be located on an above-ground trail or at the evacuation point.

Tunnels can be built at varying depths, so they can intersect on planar park map.

Builders calculate the cost of tunnel construction based on its length on the park map. The vertical component (the depth of the tunnel) is not taken into account. Suggest how to build tunnels in order to comply with the FEMA requirements, so that the total length of all tunnels is minimal.

Input

The first line of the input file contains an integer N — the number of trails in the park ($1 \leq N \leq 100\,000$). The second line contains two numbers x_e and y_e — the evacuation point coordinates.

Each of the remaining N lines describes a single trail with four numbers x_1, y_1, x_2, y_2 , where x_1, y_1 are the coordinates of one end of the trail and x_2, y_2 are the coordinates of the other end. It is guaranteed that $x_1 = x_2$ and $y_1 \neq y_2$ or vice versa.

All coordinates are integer. Absolute value of any coordinate does not exceed 10^8 .

Output

The first line of the output file must contain two integers: K — the number of tunnels to be built and A — the total length of these tunnels. Each of the following K lines must contain a description of a single tunnel.

Each tunnel can be represented as a polyline with segments parallel to the coordinate axes. The description of a tunnel must begin with the number of vertices in the polyline (at least two), followed by x and y coordinates of the polyline vertices in order (from one end to another). All numbers must be integers. Any two adjacent vertices of the polyline must have one coordinate equal, and the other one different.

The total number of vertices in all tunnels you're planning to build must not exceed 10^6 . It is guaranteed that there exists an optimal plan complying with the limitations of the output format. If several solutions are possible, print any of them. If the park already complies with the FEMA rules as is, and no tunnels are necessary, the output file must contain two numbers $K = A = 0$.

Example

input.txt	output.txt
12	5 16
0 0	3 0 0 1 0 1 3
1 3 1 5	3 2 2 0 2 0 0
3 1 5 1	4 0 0 1 0 1 1 3 1
1 3 2 3	3 2 3 3 3 3 4
2 3 2 2	3 3 2 3 3 4 3
2 2 3 2	
3 1 3 2	
5 1 5 3	
5 3 4 3	
4 3 4 4	
4 4 3 4	
3 4 3 5	
3 5 1 5	