Flow

Ivancho's real passion are the video games.

During the last geography lesson at his school they discussed the soil of the Sahara (or maybe its absence; Ivancho was too absent-minded at the time to understand what the teacher was talking about). However, Ivancho was inspired by the lesson and wanted to apply what he learnt in his next computer project. He decided that the map in the video game on which he was working on will be initially desert-based. In order to make the map more interesting he thought it would be cool to include as many as possible rivers, flows and lakes. As Ivancho likes the water more than the sand, it is not impossible to turn the desert into an ocean in the end. He asks you as good programmers to help him with the generation of the maps.

- Ivancho's map is a square table with size **N** x **N**.
- Ivancho wants to include not more than **P** flows in his map. Each flow is determined by the coordinates of its two ends **(row1, col1)** and **(row2, col2)**. The first number in the coordinates is for the number of the row and the second one is the number of the column in the table for the respective cell.
- One flow is nothing more but a path of consecutive adjacent cells starting from the one end of the flow and finishing at its other end. Two squares are adjacent if they have a common side. The flow must not go out of the table. All cells that are used must be inside. No cell can be used to build more than one flow.
- Ivancho knows that there must be **B** team bases on the map. The bases are squares in the table with coordinates **(row, col)** - the first number is the number of row and the second number - the number of column of the square in the table. There must be no flow passing through any base.

Ivancho's wish is to have as many as possible valid flows and meanwhile as much as possible water in his map.

Input: On the first line of the input file **flow.in** are the numbers **N** and **P** determining respectively the size of the table and the number of potential flows thath could be made. On each of the next **P** lines are written the coordinates of the two ends of the flow, **row1 col1 row2 col2**, separated by a space interval. After that is the information about the location of the team bases. Firstly, the number of the teams **B** is given followed by **B** lines with their coordinates, **row col**. No cell from the table will appear twice in the input file.

Output: On the output file **flow.out** print **P** lines - one line with the required data for each flow from the input file written in the same order. This means that line **i** of the output gives data about flow **i** of the input. Each line begins with one number **K** - the length of the path describing the flow, followed by **K** pairs of numbers - **row col** - the coordinates of the cells

from which the path is built. The start and the end cells of the path must coincide with the two ends of the flow given in the input. Each cell from the path must be adjacent to the previous one. If you have decided not to create the respective flow (i.e. this flow not to be in the map) print only one number 0 on its line.

Constraints:
6 <= N <= 100
1 <= P <= 250
0 <= B <= 500
For all of the coordinates in the input and the output must be valid that 0 <= row, col <= N - 1

Additional information about the tests is included in the following table.

Group	%	N	P	B
1	10	N <= 10	P <= 5	B = 0
2	15	N <= 25	P <= 50	B <= 50
3	15	N <= 50	P <= 100	B <= 100
4	20	N <= 75	P <= 150	B <= 250
5	20	N <= 100	P <= 200	B = 0
6	20	N <= 100	P <= 250	B <= 500

Time limit: 5 sec

Evaluation: For each test if the output is invalid, the program will receive 0 points on this test case. Otherwise, we calculate the following formula

(number of flows created) * (number of used cells in the table)

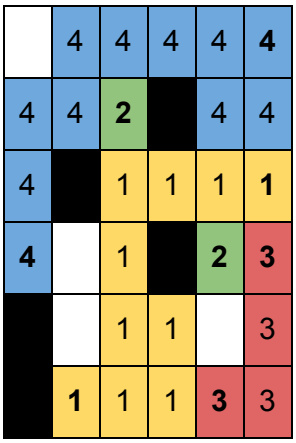
One cell is counted as used whenever there is a flow passing through it. The cells on which a team base is placed are not counted as used. Your program will get **((yours + 1) / (best + 1))^2** percent of the points predicted for the specific test case.

Visualization: To help the participants, we have prepared a visualisation for this round so that they can see how their strategies work. The documentation for the visualisation can be found with the other materials of the round.

Example input	Example output
6 4 2 5 5 1 1 2 3 4 5 4 3 5 3 0 0 5 5 4 0 3 3 5 0 1 3 2 1	10 5 1 5 2 5 3 4 3 4 2 3 2 2 2 2 3 2 4 2 5 0 4 5 4 5 5 4 5 3 5 11 3 0 2 0 1 0 1 1 0 1 0 2 0 3 0 4 1 4 1 5 0 5

Explanation of the example:

The map looks like that in the end



There are 3 valid built flows and the number of used squares is 25.
The total value of this solution is 3 x 25 = 75.