

2360 Robots

The Robots game is a one-player game played on a 31×31 board. The board is partitioned into 1×1 cells arranged in 31 rows and 31 columns. Each cell is indexed by (r, c) where r is the row and c is the column (starting from 1), and it may be empty, occupied by you, occupied by a robot, or occupied by debris. The object of this game is to move in such a way to destroy all the robots before the robots destroy you.

Initially, you occupy the cell at $(15,15)$, and there are R robots ($1 \leq R \leq 50$) located in R different cells other than $(15,15)$. All other cells are empty. You are also given a list of T ($0 \leq T \leq 20$) cells that are potential teleport locations. You make the first move, and then the robots and you alternate moves. At your move, you are allowed to walk to an adjacent cell in any one of the eight compass directions, teleport to one of the specified teleport locations, or remain stationary. You may walk to an adjacent cell if it is empty. In addition, you may walk to an adjacent cell which contains debris by pushing the debris to the next adjacent cell along the line of movement, provided that the next cell does not already contain debris. If a cell that debris is pushed to contains a robot, that robot is destroyed. If you choose to teleport, the destination must be an empty cell. You may not make any move that will leave you or any debris that you push outside of the board.

When the robots move, each robot walks to the adjacent cell (even if it is not empty) in the eight compass directions such that the destination cell is closest to your current position (i.e. after your last move). The distance between two cells (r_1, c_1) and (r_2, c_2) is defined to be $|r_1 - r_2| + |c_1 - c_2|$. All robots walk at the same time during a move. If two or more robots walk to the same cell, or if a robot walks to a cell containing debris, all of these robots are destroyed. Destroyed robots become debris.

You lose the game if any robot walks to your current position, even if multiple robots do so and destroy each other. You win the game if all robots are destroyed and none has moved to your current position.

In order to stay in the game as long as possible, you will only consider moves that do not lead to an immediate loss (a loss before your next move). A plausible strategy is to always walk to a cell (or remain stationary) such that the number of robots remaining after your move and the robots' move (i.e. just before your next move) is minimized. In case of a tie, choose the move that maximizes the minimum distance to the remaining robots just before your next move. If there are still ties, choose the move that also minimizes the row index of the destination cell, and finally, break remaining ties by also minimizing the column index.

If it is not possible to make a move by walking or by remaining stationary without leading to an immediate loss, you should teleport to the first unused legal destination chosen from a list of locations given to you, as long as it does not lead to an immediate loss. When you search for a teleport site, you should always start the search at the beginning of the list. If no such teleport destination is available, you should remain stationary, leading to an immediate loss.

In this problem, you will implement this strategy and see how well it works.

Input

The input consists of a number of instances. The first line of each instance contains the integers R and T separated by a space. This is followed by R lines containing two integers separated by a space, indicating the row and column of the initial positions of the R robots. You may assume that each robot initially occupies a cell which is not $(15,15)$ and the locations of the robots are distinct. The next T lines give the list of teleport destinations available. Each line is given by the row and column of the destination cell, separated by a space. The input is terminated by a case with $R = T = 0$.

Output

For each case, print the case number (starting from 1), in the format shown in the Sample Output, on its own line. For each teleport taken, print one line of the form:

Move m : teleport to (r,c)

where m is the number of moves you have made (including this one), and (r,c) is the destination of the teleport. This is followed by three lines containing the result of the game. If you win the game, print

Won game after making m moves.

Final position: (r,c)

Number of cells with debris: d

where m is the number of moves you have made when you won the game, (r,c) is your final position, and d is the number of cells with debris (use the word “moves” even if $m = 1$). If you lose the game, print

Lost game after making m moves.

Final position: (r,c)

Number of cells with debris: d

Number of robots remaining: n

where m is the number of moves you have made when you lost the game, (r,c) is the location at which you are destroyed, d is the number of cells with debris, and n is the number of robots remaining when you lost the game (use the word “moves” even if $m = 1$).

Separate the output for each case by a blank line.

Sample Input

```
4 0
17 18
13 18
8 12
10 12
4 0
17 17
13 17
13 13
17 13
3 3
17 18
13 18
5 31
15 16
16 15
3 7
0 0
```

Sample Output

Case 1:

Won game after making 5 moves.

Final position: (14,16)

Number of cells with debris: 1

Case 2:

Lost game after making 2 moves.

Final position: (15,15)

Number of cells with debris: 1

Number of robots remaining: 0

Case 3:

Move 30: teleport to (16,15)

Move 58: teleport to (15,16)

Move 86: teleport to (3,7)

Lost game after making 114 moves.

Final position: (1,29)

Number of cells with debris: 1

Number of robots remaining: 1