

Mirrors Revisited

Dickie is playing with mirrors again, but now, he brought some friends to the mirror room, to make every friend different, Dickie gave them a flashlight to each friend with a unique color (red, blue, yellow and green), Dickie has the white flashlight and they started playing, after having fun for hours, they all thought the same idea, they want to unite the beams of light for each flashlight as long as they can, of course, they cannot stand at the same place and point to the same mirror, so each friend must be in a different square from any other, he can choose to point his flashlight to any of the 4 diagonals (south-west, south-east, north-west, north-east). Your task is to find the number of beams that intersect any other in the room, as it is the speed of light, the time is irrelevant and the direction that each friend has to point to make it.

There are certain rules, by instance, if a friend or Dickie choose to stay in a borderline ([0,X] [X,0], [0,M], [N,0]) he cannot point his flashlight to the direction corresponding to the collide. That is, if Dickie or some friend is in position (1,0), representing the row 1 column 0, he cannot point the flashlight to north-east nor north-west, equivalent things happen for each scenario.

We say that a light intersect any other if and only if the square (I,J) has received more than one flash beams. That is, a single flash beam doesn't count as an intersection.

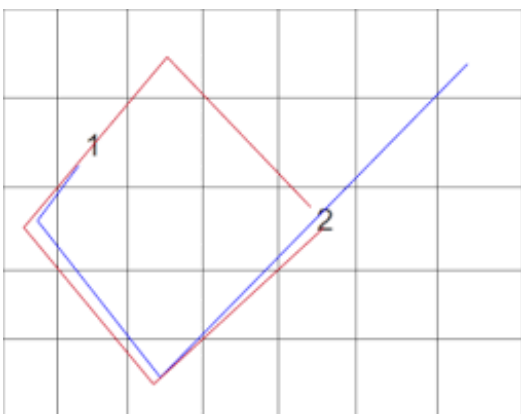
If the same result comes from different settings directions, you should print the one immediately smallest lexicographically, by instance,

if friend 1 points the flashlight to south-west and friend 2 points the flashlight to north-east and make it to 10 intersections, but the same happens if friend 1 points the flashlight no north-east and friend 2 points the flashlight to north-east, choose the second option as "north-east" is lexicographically smallest than south-west.

If friend 1 from scenario 1 and friend 1 from scenario 2 shares the same direction, you should keep until you find the best answer.

You may use string comparison to find the answer (string A < string B)

For simplicity, we are only interested on the light beam until it reaches one of the visited points, when crashing to a diagonal, it is safe to say that it will stop immediately.



The figure above shows the maximum path (6 square intersection) between two flashlights

INPUT DETAILS:

Each input file starts with three integers N,M,F, denoting respectively the size (NxM) of the matrix and the number of flashlights given. After that, F lines will come each denoting the position of every friend in the matrix

OUTPUT DETAILS:

Print in a single line the sum of intersections of flash beams in the room, then, print F lines in the format "S. |DIRECTION|" where S denotes the number of the friend you are showing, followed by a period '.' and a space, then print the direction that the friend holds to make the beam (north-west, north-east, south-east, south-west are accepted as a valid direction).

INPUT	OUTPUT
4 5 3	6
1 1	1. north-east
1 2	2. north-east
2 2	3. south-east
INPUT2	OUTPUT2
4 5 2	5
2 4	1. north-west
0 0	2. south-east

CONSTRAINTS:

$1 \leq N, M \leq 200$

$1 \leq F \leq 5$