

Problem 2 – Expedition

You are on an expedition in the mountains but the signs are pointing the wrong way and no one knows how to find you – so you're lost. It is getting dark already and you remember that you have an old map of the surrounding terrain in your backpack. To read it, first you've got to decode it. If you manage to figure out the way back to the rest house, you'll get there safely. Otherwise, you'll have to spend the night in the mountains and, as you know, the night is dark and full of terrors(bears).

You have an **encrypted** map represented by a **rectangular** matrix (**primary**) with dimensions **N x M** and **random** values of **0** and **1**. An element of the matrix with **value 0** corresponds to a **path** and an element with **value 1** – to an **obstacle** that **cannot be crossed**. In addition to the primary matrix, there is another one – **secondary** rectangular matrix, with dimensions **P x Q**. Its elements are also of **random** values (**0** and **1**). To be able to decode the map, you've got to overlay the secondary matrix on the primary matrix **n-times**. The coordinates of the element from the **primary** matrix, corresponding to the **upper-left (0, 0)** element of the **secondary** matrix, are received from the input. For example, if the received input is [1, 1] you have to place the upper-left (0,0) corner of the secondary matrix at 1, 1 of the primary matrix. Use the following modification criteria for **altering the primary** matrix:

- Element with **value 1** from the **secondary** matrix **inverts** the value of the corresponding element in the **primary** matrix;
- Element with **value 0** from the **secondary** matrix **doesn't change** the value of the corresponding element in the **primary** matrix;

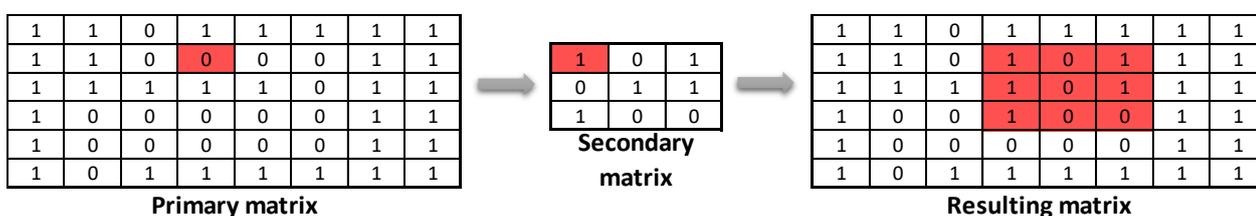
When you're done with **all of the alterations** of the primary matrix you will get the **final** matrix. It represents the **deciphered** map of the terrain and you can use it to find your way back to the rest house. Your current location (**start**) is **always** going to be **on one of the four sides** of the matrix (**excluding the corners**) and the end will not be with the same coordinates as the start. You can only move **up, down, left** or **right**. At every moment, there will be **only one possible direction** to take, **if any**. In case you've come to a **dead end**, there is no way back and you've got to spend the night in the mountains.

Input

You will receive four arguments – **primary** matrix, **secondary** matrix, **overlay** coordinates and **starting** coordinates. The overlay coordinates are an array of arrays, holding (x, y) coordinate pairs. The starting coordinates are also an (x, y) coordinate pair.

Output

The first line of the output contains the number of steps made from the beginning to the end of the path. The starting point is the first step. Each cell of the matrix, holding value 0, corresponds to a single step. In case you find a way out, on the second line you must print "**Top**"/"**Bottom**"/"**Right**"/"**Left**", depending on that on which side of the matrix lies the exit (there won't be an exit in the corners of the matrix). If you've come to a dead end, you must print "**Dead end**", followed by a **quadrant number**, in which the end of the path is. See the examples below to understand how it works.



Start
↓

0	1	0	1	1	1	1	1	0
1	1	0	0	0	0	0	0	1
1	1	1	1	1	1	1	0	1
1	0	0	0	0	0	0	0	1
1	0	1	1	1	1	1	1	1
1	0	1	1	0	1	1	1	0

↓
End

Start
↓

1	1	0	1	1	1	1	1	1
0	1	2	0	0	0	0	1	0
1	0	1	1	1	1	1	0	1
1	1	1	1	0	0	0	0	1
0	0	3	1	1	1	1	4	1
1	1	1	1	0	1	1	1	0

Constraints

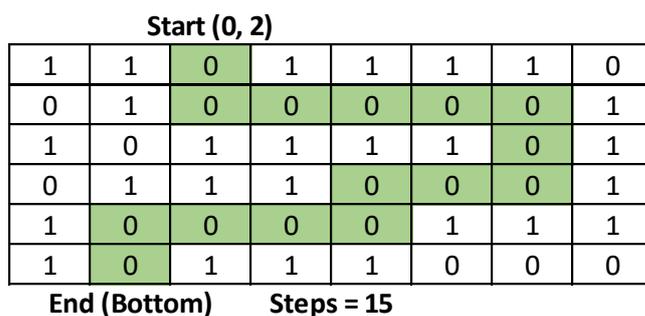
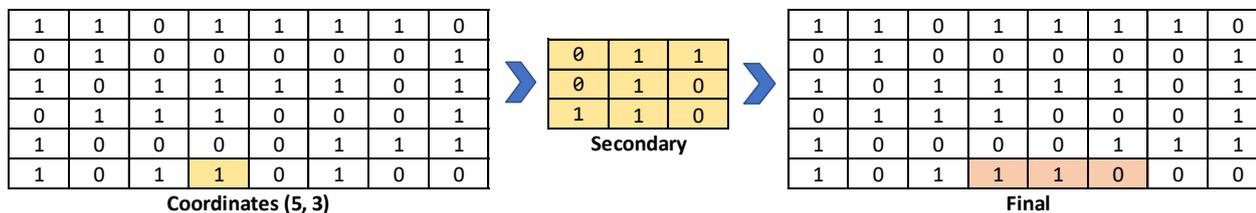
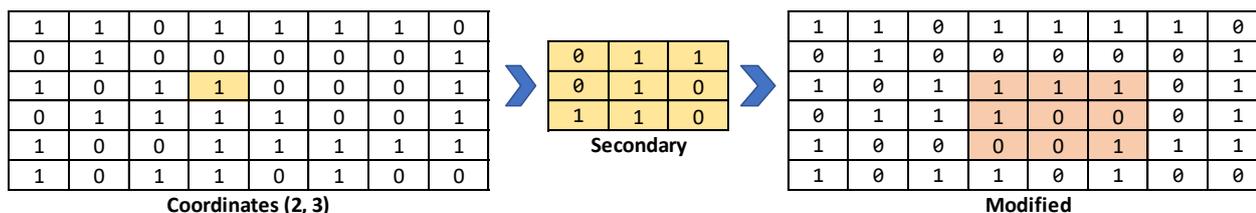
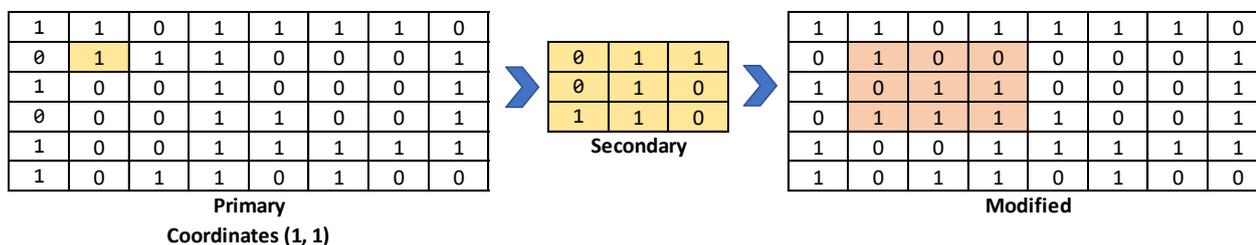
- The number of rows and columns of the primary matrix will always be an even number.

Scroll down for examples.

Examples

Input	Output
<pre>[[1, 1, 0, 1, 1, 1, 1, 0], [0, 1, 1, 1, 0, 0, 0, 1], [1, 0, 0, 1, 0, 0, 0, 1], [0, 0, 0, 1, 1, 0, 0, 1], [1, 0, 0, 1, 1, 1, 1, 1], [1, 0, 1, 1, 0, 1, 0, 0]], [[0, 1, 1], [0, 1, 0], [1, 1, 0]], [[1, 1], [2, 3], [5, 3]], [0, 2]</pre>	15 Bottom

Explanation



Input	Output
<pre> [[1, 1, 0, 1], [0, 1, 1, 0], [0, 0, 1, 0], [1, 0, 1, 0]], [[0, 0, 1, 0, 1], [1, 0, 0, 1, 1], [1, 0, 1, 1, 1], [1, 0, 1, 0, 1]], [[0, 0], [2, 1], [1, 0]], [2, 0] </pre>	<p>4</p> <p>Dead end 1</p>

Explanation

