Pearls
Problem ID: pearls

Nikoli's Jewelry Store in Puzzletown sells a line of necklaces consisting of black and white pearls. The pearls in the necklace are firmly glued to a cord of length k, where each unit of cord length either holds a pearl or is empty. Each necklace is displayed on a rectangular velvet-lined surface divided into a grid, where each cell of the grid is either a round depression (to hold a pearl) or a flat region (which may be empty or may contain a section of empty cord). A properly-displayed necklace corresponds to a closed, non-self-intersecting path joining some of the cells of the display.

Because this is, after all, Puzzletown, there are some tricky rules governing how the necklace is to be displayed, namely, the rules of a puzzle called "Masyu." When the the necklace is set down along the path (the spacing units on the string match the spacing of the cells on the display surface), the pearls satisfy the constraints of the Masyu puzzle, i.e.,

- A white pearl may not be set down on a cell containing a path corner; however, at least one of the two adjacent cells that extend the path through the pearl must contain a corner.

- A black pearl must be placed in a cell containing a path corner; however, neither of the two cells extending the path through the black pearl may contain a corner.

An example of a necklace correctly displayed is shown in Figure 1 (this also corresponds to sample input 1).
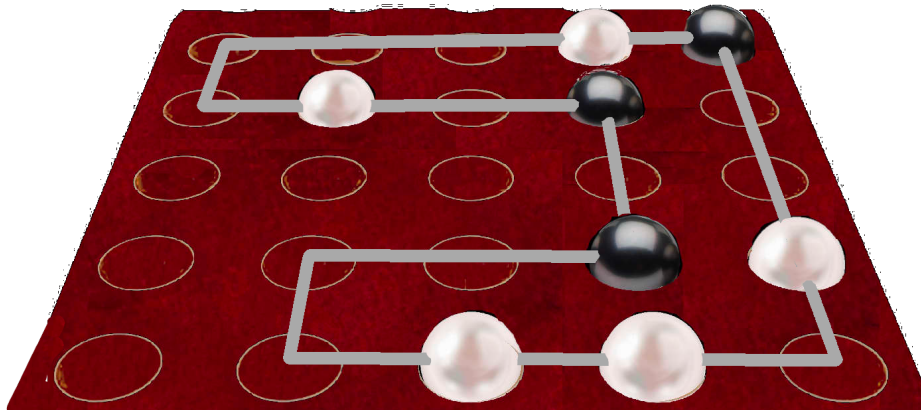

Figure 1: A necklace of length 20 on its display platform

Input

The first line contains 3 integers k, n, and m, where $k \geq 6$ is the length of the string (pearls and empty slots), n is the number of rows of the velvet grid, and m is the number of columns. The upper-left cell is row 1, column 1. The second line describes a string of length k consisting only of the characters B, W, and . (for black pearl, white pearl, and empty space). The first character will always be a pearl—either B or W. The third line contains two integers r and c, the row and column of the grid that contains the first pearl in the string.

Output

If there exists a proper way to display the necklace within the given grid boundaries, print a path description of the necklace layout, assuming the first pearl in the string is located at row r, column c of the grid and the path describes the pearls in the same sequence as the input string. The path description should consist of the letters N,S,E, and W, indicating whether the path proceeds north, south, east, or west from the current cell. The path should be closed and

should not intersect itself. If there is more than one such path, output the one whose description is alphabetically the smallest.

If there is no possible path, output `impossible`

Comments

I found this site useful for generating my sample inputs/outputs: `https://www.kakuro-online.com/masyu/` However, it does not choose the lexicographically smallest path as described in the output section.

I can think of variants, none of which I've given any thought to whatsoever, e.g., find the smallest sized grid into which you can fit the necklace, or omit the spaces in the string and assume beads can slide, then find the smallest length string that can be used to meet the requirements, or . . . .

This is a pure off-the-top-of-my-head idea. I haven't written a solution and I don't even know if it's a hard or easy problem. [Added later–it's NP-complete: `https://web.archive.org/web/20120204150006/http://www2.stetson.edu/~efriedma/papers/pearl/pearl.html`]

I assumed (without actually searching) that somebody, somewhere has written a problem asking for a solution to a Masyu puzzle, but I thought perhaps this variant might be new. Here, we are given a clockwise ordering of the black and white beads rather than their locations, and we're given the length of the path and the spacing between the beads. I'm not sure how many different solutions there could be for a given input, or if it's easy to guarantee a unique solution with an appropriate choice of inputs. (For starters, I fixed the location of the first bead.)

Sample Input 1	Sample Output 1

`20 5 5`
`W.B.B...WW.W..BW....`
`2 2`
	`EESSWWSEEENNNNWWWWSE`

Sample Input 2	Sample Output 2

`6 5 5`
`W..W.B`
`3 3`
	`impossible`