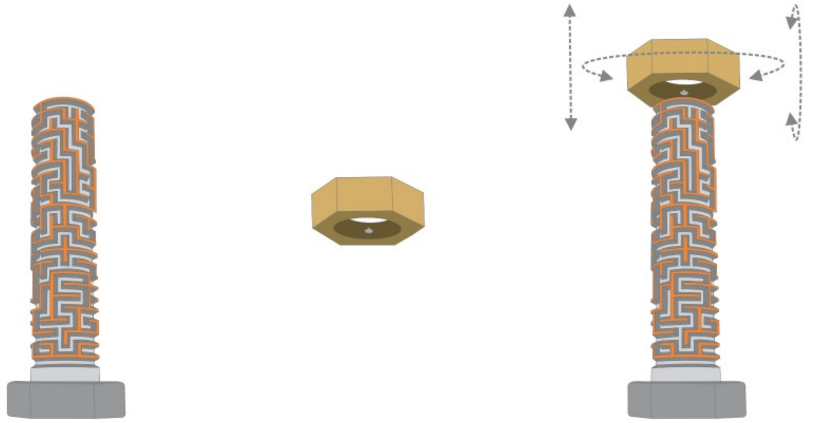
Problem M – Maze in Bolt

Author: Welton Cardoso, Brasil

Usain has an online store specializing in selling 3D puzzles made by 3D printers. One of the best-selling puzzles these days is the Maze in Bolt. This puzzle is composed of two parts: a screw-shaped piece with an embossed labyrinth engraved on it, and a nut. The inner part of the nut may contain tips. These tips make the nut slide only through the corridors of the labyrinth.

Initially, the two parts of the puzzle are separated. The challenge is to slide the nut all the way through the maze until it reaches the head of the bolt. The nut can be moved clockwise, counterclockwise, down (towards the head of the bolt), and up (away from the head). Each of these movements is only possible when every tip in the inner part of the nut is not prevented from sliding to the new position due to some wall of the maze. Besides the mentioned movements, another one is allowed: when the bolt and the nut are separated, the nut can be flipped. The illustration below shows both parts of the puzzle as well as all the allowed moves.



(a) Screw-shaped piece (b) Nut piece (c) Movements of the nut

A customer placed an order for a large quantity of the Maze in Bolt. Each puzzle is designed by Usain himself in a random and unique way but, due to the size of the order and the tight deadline, he believes he will not be able to check whether the created puzzles have a solution or not. Usain asked for your help in devising an algorithm that quickly checks any given pair of nut and bolt. For doing so, the inner part of the nut will be modeled as a binary circular string. Regarding the bolt, each row of the maze will be modeled the same way.

Input

The first line contains two integers R ($1 \leq R \leq 1000$) and C ($3 \leq C \leq 100$), indicating respectively the number of rows and columns of the maze. The second line contains a circular string S of length C, representing the inner part of the nut. Each character of S is "`1`" if the nut has a tip in the corresponding position, while an empty space is indicated by character "`0`". Each of the next R lines contains a circular string describing a row of the maze. In this case, each character of the string is "`1`" if the maze has a wall in the corresponding position, while an empty space is indicated by character "`0`". Rows are given from top (the tip of the bolt) to bottom (the head).

Output

Output a single line with the uppercase letter "`Y`" if the puzzle has a solution, and the uppercase letter "`N`" otherwise.

Sample input 1	**Sample output 1**

8 13
0110010000000
1100101110100
1001101000100
1100101110100
1000100010000
1010111011001
0000001010000
1001101111101
0001001100100
	Y

Sample input 2	**Sample output 2**

1 3
100
101
	Y

Sample input 3	**Sample output 3**

2 3
100
101
010
	N

Sample input 4	**Sample output 4**

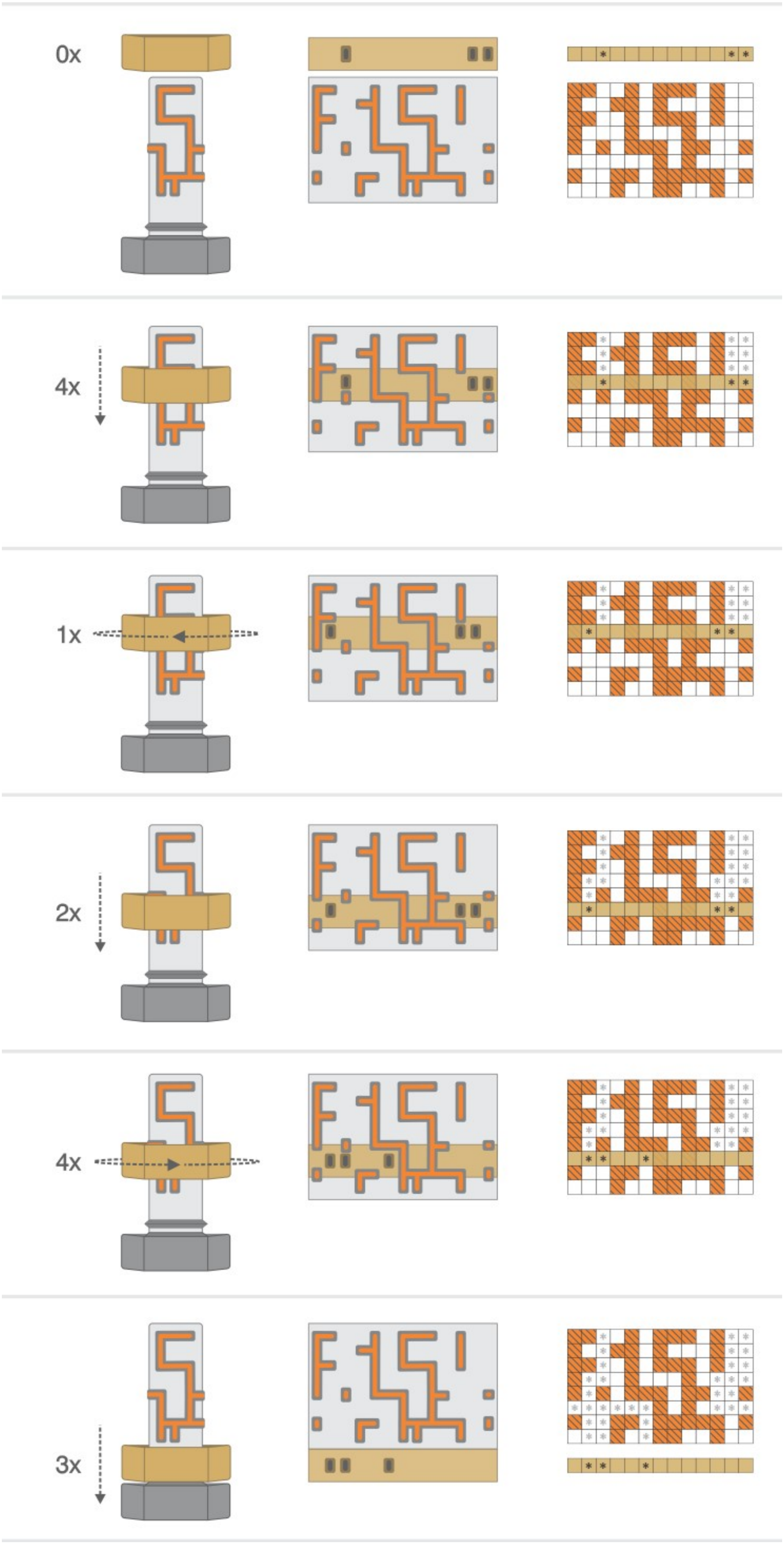
4 6
001000
011111
010001
010100
110111
	Y

Sample input 5	**Sample output 5**

1 6
001011
001011
	Y

The picture below explains the first sample. Images on the left represent 3D views of different stages of the game. In the middle, the images are flattened 2D versions of the corresponding 3D views. Finally, images on the right represent the stages of the game according to the described model (although for the sake of clarity, each character “1” has been replaced by a given symbol, and each character “0” is shown as an empty space).

It can be seen in the top three images that the nut (having three pins) and the bolt start separated. The second group of three images shows the situation of the game after the nut has been moved four rows down (towards the head of the bolt). Then the nut is rotated one position, moved down two more rows, rotated four positions in the opposite direction, and finally moved down three rows, which solves the puzzle. Note that in this case the nut has not been flipped, nor moved up (away from the head of the bolt).


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