

C. Ice Cave

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

You play a computer game. Your character stands on some level of a multilevel ice cave. In order to move on forward, you need to descend one level lower and the only way to do this is to fall through the ice.

The level of the cave where you are is a rectangular square grid of n rows and m columns. Each cell consists either from intact or from cracked ice. From each cell you can move to cells that are side-adjacent with yours (due to some limitations of the game engine you cannot make jumps on the same place, i.e. jump from a cell to itself). If you move to the cell with cracked ice, then your character falls down through it and if you move to the cell with intact ice, then the ice on this cell becomes cracked.

Let's number the rows with integers from 1 to n from top to bottom and the columns with integers from 1 to m from left to right. Let's denote a cell on the intersection of the r -th row and the c -th column as (r, c) .

You are staying in the cell (r_1, c_1) and this cell is cracked because you've just fallen here from a higher level. You need to fall down through the cell (r_2, c_2) since the exit to the next level is there. Can you do this?

Input

The first line contains two integers, n and m ($1 \leq n, m \leq 500$) — the number of rows and columns in the cave description.

Each of the next n lines describes the initial state of the level of the cave, each line consists of m characters "." (that is, intact ice) and "X" (cracked ice).

The next line contains two integers, r_1 and c_1 ($1 \leq r_1 \leq n, 1 \leq c_1 \leq m$) — your initial coordinates. It is guaranteed that the description of the cave contains character 'X' in cell (r_1, c_1) , that is, the ice on the starting cell is initially cracked.

The next line contains two integers r_2 and c_2 ($1 \leq r_2 \leq n, 1 \leq c_2 \leq m$) — the coordinates of the cell through which you need to fall. The final cell may coincide with the starting one.

Output

If you can reach the destination, print 'YES', otherwise print 'NO'.

Examples

input
4 6 X...XX ...XX. .X..X. 1 6 2 2
output
YES

input
5 4 .X.. ...X X.X.XX.

5 3 1 1
output
NO

input
4 7 ..X.XX. .XX..X. X...X.. X..... 2 2 1 6
output
YES

Note

In the first sample test one possible path is:

After the first visit of cell (2, 2) the ice on it cracks and when you step there for the second time, your character falls through the ice as intended.