Mirko has been learning to drive, but he still **cannot make a U-turn in a narrow street**. That's why he has decided to go practice in a town where U-turns are forbidden everywhere. This prohibition can be marked by the following sign:

Mirko has soon figured out that his ideal town must not contain dead-end streets, since it is impossible to exit such a street without a U-turn (let us assume that Mirko cannot drive in reverse either). Write a program to analyse a town map and determine whether the town is suitable for Mirko (i.e. whether the town has any dead-end streets).

The town map is a table with **R** x **C** cells, where each cell is a building segment (denoted by X) or a road surface (denoted by a dot). From a road surface cell, Mirko can move to any of the surrounding four cells (up, down, left, or right), provided that it is also a road surface (i.e. not a building).

Formally, we will determine that a town is free of dead-end streets if, starting from any road surface cell and going in any of the possible directions, we can return to the starting cell without making a 180 degrees turn (changing our direction to the opposite one).

INPUT

The first line of input contains the positive integers **R** and **C** ($3 \leq$ **R**, **C** ≤ 10), the dimensions of the town.

Each of the next **R** lines contains **C** characters, with each character either "X" or ".". These **R** x **C** characters represent the town map as described in the text above. At least two cells will be road surfaces, and all road surfaces will be connected (i.e. mutually reachable).

OUTPUT

The first and only line of output must contain 0 if the town is free of dead-end streets, otherwise it must contain 1.

SAMPLE TESTS

input	input	input
4 3 XXX X.X X.X XXX	5 5 XX.XX X...X ..... X...X XX.XX	3 9 ...XXX... .X.....X. ...XXX...
output	output	output
1	1	0