9. Platform

Program Name: Platform.java   Input File: platform.dat

You've been placed in a maze. You really dislike mazes. Thankfully, you've been given a layout of the maze to make your life a bit easier, but looking at it, you're not sure if it's actually possible to reach an exit.

The layout is given as an *n* x *m* grid. You are suddenly very glad you have it, as it marks the locations of the floor tiles that will begin a headfirst slide across what is turning out to be an incredibly slippery maze, which are represented as follows:

- ^ makes you slide upwards

- > makes you slide to the right

- < makes you slide to the left

- v makes you slide downwards

Slippery tiles, which will only help you along your runaway slide, are represented by a '.'. Thankfully, there are some platforms that prevent you from sliding further. Those platforms are denoted by an 'x'. You may move from one platform to any adjacent sliding floor piece or platform within the maze, but not a slippery tile. You may not change your sliding direction while sliding, unless you slide over another floor piece that makes you slide another direction.

You always start in the top left corner of the grid. In order to escape, you must land on the bottom right corner. Both locations are always guaranteed to be platforms. You can never slide off the grid, and there won't be a situation where you slide forever.

Input
The first line of input consists of a single integer, *t*, indicating the number of test cases that follow.
For each test case, there will be two integers, *n* and *m*. The following *n* lines have *m* characters each, NOT separated by a space. These *n* lines represent the maze layout.

Output
For each test case, print YES if it's possible to reach the bottom right platform, and NO if it's not possible.

Constraints
```
1 <= t <= 10
1 <= n, m <= 100
```

Example Input File
```
3
1 5
x>..x
3 3
x>v
xv<
^<x
7 7
x>....v
xv.>.v.
...^...
...x..<
...>..v
...^.<.
.>..x.x
```

9. Platform (cont.)

Example Output to Screen

```
YES
NO
YES
```