Problem G
Rafting Trip
Time Limit: 1 Second(s)

You are planning a rafting trip. The terrain can be viewed as a grid. Each cell is either land, or has part of a river flowing through it in one of the four directions: north, south, east, or west. Some land cells contain a sightseeing spot.

You can choose any river cell as the starting point of your rafting trip. Once your raft reaches a river cell (including the starting cell), it will follow the water direction of that cell and move to an adjacent cell or exit the grid.

You can visit a nearby sightseeing spot if your raft reaches a river cell that is adjacent to it, including the starting cell. (Cell adjacency includes horizontal and vertical neighbors, but not diagonal neighbors.) Each sightseeing spot can be visited at most once.



Illustration of the first sample case. The optimal rafting trip starts at the cell with the raft and visits 4 sightseeing spots (marked by binoculars). The river cells reached along the trip are highlighted in dark blue.

Your rafting trip stops when your raft moves onto a land cell, exits the grid, or enters a river cell that it has reached before. Note that if the raft ends at a land cell, you cannot visit the sightseeing spots adjacent to that land cell.

What is the maximum number of sightseeing spots you can visit in a single rafting trip if you choose your starting cell optimally?

Input

The first line of input contains two integers r and c ($2 \le r, c \le 500$); the terrain grid has r rows and c columns.

Each of the next r lines contains c characters describing one row of the terrain grid. A dot `'.'` denotes a land cell without a sightseeing spot. A hash `'#'` denotes a land cell that contains a sightseeing spot. River cells are denoted by `'^'` (north), `'v'` (south), `'>'` (east), or `'<'` (west). There is at least one river cell in the grid.

Output

Output a single line with a single integer, which is the maximum number of sightseeing spots you can visit in a single rafting trip.

Sample Input 1	Sample Output 1
5 6 v<<<#v v#v<.> >>v<<< ..v##^ #<<<.^	4

Sample Input 2	Sample Output 2
4 5 >v<<. ^<..# #...# .#>^#	2