F. Guards In The Storehouse

time limit per test: 1.5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Polycarp owns a shop in the capital of Berland. Recently the criminal activity in the capital increased, so Polycarp is thinking about establishing some better security in the storehouse of his shop.

The storehouse can be represented as a matrix with n rows and m columns. Each element of the matrix is either `.` (an empty space) or `x` (a wall).

Polycarp wants to hire some guards (possibly zero) to watch for the storehouse. Each guard will be in some cell of matrix and will protect every cell to the right of his own cell and every cell to the bottom of his own cell, until the nearest wall. More formally, if the guard is standing in the cell (x_0, y_0), then he protects cell (x_1, y_1) if all these conditions are met:

- (x_1, y_1) is an empty cell;
- either $x_0 = x_1$ and $y_0 \le y_1$, or $x_0 \le x_1$ and $y_0 = y_1$;
- there are no walls between cells (x_0, y_0) and (x_1, y_1). **There can be a guard between these cells, guards can look through each other.**

Guards can be placed only in empty cells (and can protect only empty cells). The *plan* of placing the guards is some set of cells where guards will be placed (of course, two plans are different if there exists at least one cell that is included in the first plan, but not included in the second plan, or vice versa). Polycarp calls a plan *suitable* if there is **not more than one** empty cell that is not protected.

Polycarp wants to know the number of suitable plans. Since it can be very large, you have to output it modulo $10^9 + 7$.

Input

The first line contains two numbers n and m — the length and the width of the storehouse ($1 \le n, m \le 250$, $1 \le nm \le 250$).

Then n lines follow, ith line contains a string consisting of m characters — ith row of the matrix representing the storehouse. Each character is either `.` or `x`.

Output

Output the number of suitable plans modulo $10^9 + 7$.

Examples

input
1 3 .x.
output
3

input
2 2 xx xx
output
1

input
2 2 .. ..

output
10

input
3 1 x . x

output
2

Note

In the first example you have to put at least one guard, so there are three possible arrangements: one guard in the cell $(1, 1)$, one guard in the cell $(1, 3)$, and two guards in both these cells.