

Harrier's Forbidden Dungeon Expedition

Input file: `stdin`
Output file: `stdout`



Figure 1: The Forbidden Dungeon. Image courtesy of Freepik.com.

In the depths of an enigmatic and ancient dungeon, a fabled treasure was believed to be hidden. The dungeon was an intricate labyrinth of dark corridors, chambers, and secret passages, and at the heart of it all lay the elusive treasure.

Harrier, an intrepid adventurer known for their unwavering courage, had come into possession of a mysterious map. This map revealed the layout of the dungeon, with intricate pathways and chambers to explore. However, the map posed a unique challenge—Harrier could only move rightwards or downwards through the dungeon's passages. The dungeon's forbidding nature meant that making a wrong turn could lead into a death trap.

Complicating the quest, some chambers within the dungeon were shrouded in peril. These chambers were marked as forbidden, and entering one of them could spell doom for Harrier. Navigating the dungeon and finding the treasure required not only keen observation but also precise decision-making.

Harrier, being both an experienced explorer and a skilled programmer, decided to develop a program to solve this enigma. They knew that efficiency was paramount, as the dungeon was known for its complexity. Harrier needed a solution to guide them through the labyrinth, avoiding forbidden chambers, and leading them to the coveted treasure. He had obtained a map of the dungeon. The map represented the dungeon as a grid with “.” and “x” symbols. A dot represented a safe chamber, while a cross denoted a forbidden chamber. His mission was to list the number of paths for the reachable chambers in the dungeon, starting from the top-left corner and reaching the treasure at the bottom-right corner while avoiding the forbidden chambers.

Harrier needed a solution that would navigate the dungeon while avoiding forbidden chambers and determining the number of paths to each of the reachable chambers.

Input

The first line contains two integers, m and n ($1 \leq m, n \leq 100$), the number of rows and columns in the grid respectively.

m lines follows and each line containing n characters, being either a . or a x.

Output

For the first line, print k , the number of following lines. k lines follows, containing three integers x , y and c_{xy} . Hereby c_{xy} is the number of paths, modulo $10^9 + 7$, of the reachable chamber at (x, y) .

The k lines must be sorted with ascending xs , and with ascending ys when the xs are the same. Output with a different permutation will not be accepted.

Sample input and output

stdin	stdout
2 3 ... x..	5 0 0 1 0 1 1 0 2 1 1 1 1 1 2 2
3 5x.x.	13 0 0 1 0 1 1 0 2 1 0 3 1 0 4 1 1 0 1 1 2 1 1 4 1 2 0 1 2 1 1 2 2 2 2 3 2 2 4 3

The below figure reflects the first input dungeon. We will label the forbidden chambers with slimes. In this example, there are two ways to reach $(1, 2)$:

- $(0, 0) \rightarrow (0, 1) \rightarrow (0, 2) \rightarrow (1, 2)$
- $(0, 0) \rightarrow (0, 1) \rightarrow (1, 1) \rightarrow (1, 2)$

Therefore $c_{12} = 2$. Also, it is obvious that the paths from $(0, 0)$ to $(0, 0)$, $(0, 1)$, $(0, 2)$ and $(1, 1)$ are all unique. Thus $c_{00} = c_{01} = c_{02} = c_{11} = 1$.

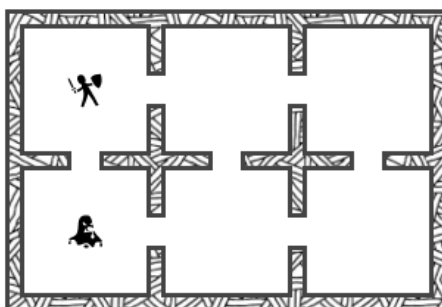


Figure 2: The dungeon specified in the first input.