Loop around Lake

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Lesya is working as a landscape designer. Her task is to create a road around the lake in the park using as few of construction materials as possible. She can only create the road on the lawn cells.

THe park is a grid of $h \times w$ cells, each cell can be either ground, or lawn, or lake. You are given the park map where lawn cells are shown. It is guaranteed that:

1) the set of lawn cells is 4-connected, that is, it is possible to get from each cell to any other cell by walking between cells that have a common side;
2) the set of lawn cells has only one internal area which is 4-connected — the lake;
3) it is possible to start from some lawn cell, walk around the lake, and get back to the same cell, and the lake will be inside the walk loop. Let us call such set of cells *4-looping* the lake.

Lesya must design the road by replacing some lawn cells with road cells, so that the road was 4-looping the lake, and the number of cells in the road was as small as possible.

Help her to come up with the optimal road plan.

Input

The first line contains two integers h and w ($3 \le h, w \le 1000$).

Each of the next h lines contains w characters "." and "#" denoting land/lake and lawn, respectively.

It is guaranteed that the lawn in 4-connected and contains exactly one inner 4-connected area.

Output

Output h lines with w characters each — the optimal plan of the road. The character '`#`' denotes the road cell, the character '`.`' denotes any other cell. Note that the road can only be built on lawn cells.

If there are several optimal plans to build the road, you can output any of them.

Example

standard input	standard output
4 6 ###### #.#.## #...#. #####.	#####. #...#. #...#. #####.