Problem D
Deceptive Directions
Time limit: 2 seconds

You find yourself on a remote island, searching for a legendary lost treasure. However, despite having gotten your hands on directions leading straight to the treasure, you have a problem. It turns out you have a saboteur in your expedition, and that at some point they edited the precious directions so they no longer lead to the treasure.

The island can be viewed as a rectangular grid, and the instructions are a sequence of east/west/north/south steps to take in this grid, from a given starting position. These instructions lead straight to the treasure (but may involve walking around obstacles) in the sense that there is no shorter way of reaching the treasure. However, the saboteur has arbitrarily replaced each step of the instructions by a step in one of the other three directions. In other words, any "west" step has been replaced by "east", "north" or "south". This replacement has been done independently for each step, so one "west" may have been replaced by "north" and another by "south", and so on.

Because of this sabotage, the instructions seem pretty useless. But maybe they can still be used to narrow down the search. Write a program to find all possible locations of the treasure.

Input

The first line of input consists of two integers w and h ($3 \leq w, h \leq 1000$), the width and height of the map. Then follow h lines, each containing w characters, describing the map. Each such character is either a '`.`' symbolizing a walkable space, '`#`' symbolizing an obstacle such as a body of water, dense forest, or a mountain, or '`S`' symbolizing the starting point of the directions.

Finally, there is a line containing a string I ($1 \leq |I| \leq 10^5$) consisting only of the characters '`NWSE`', giving the faulty instruction sequence.

The map has exactly one '`S`' and its boundary consists only of obstacle cells. The faulty instruction sequence is such that there is at least one possible location of the treasure.

Output

Output the map in the same format as the input (without the first line specifying the dimensions), with all possible locations of the treasure indicated by exclamation marks ('`!`').

Sample Input 1	Sample Output 1
5 5 ##### #...# #.S.# #...# ##### N	##### #...# #!S!# #.!.# #####

Sample Input 2	Sample Output 2
7 5 ####### #..#..# #..S..# #..#..# ####### ESS	####### #!.#..# #..S..# #..#..# #######