

## Problem C. Gravity

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

You are given an  $N \times M$  binary matrix. Cell  $(i, j)$  contains the character “.” if it is free and the character “#” if the cell is occupied by a piece. Each maximal 4-connected component of “#” characters forms an indivisible piece. During the process described below, the pieces do not merge or split in any way. All cells that are part of the same piece will move in exactly the same way.

The pieces start falling towards the floor (the last row of the matrix) with equal speeds. The pieces move down without any rotations. Every second, all pieces try to move down one row. If this motion results in a piece crossing the lower boundary of the matrix, that piece stops in place instead. Similarly, if this motion results in a piece overlapping with another piece (note that this can only happen if the latter piece is not moving), then the former piece also stops in place. In other words, pieces stop falling when they hit the floor or when they hit another piece.

Output the final state of the pieces after all of them stop falling.

### Input

The first line of input contains two integers  $N$  and  $M$  ( $1 \leq N, M \leq 2000$ ).

The next  $N$  lines describe the matrix. Each of them contains  $M$  characters which are either “.” or “#”. The characters denoting cells on the same line are **not** separated by any whitespace.

### Output

Print the resulting matrix after all pieces have finished falling. The matrix must be printed in the same format as given in the input, except for the line containing the matrix dimensions.

### Example

standard input	standard output
10 10	.....
.....	.....
..#####..	..#####..
..#...#..	..#...#..
..#.#.#..	..#...#..
..#.#.#..	..#...#..
..#...#..	..#.#.#..
..#####..	..#####..
.....	.....#..
..#...#..	..#...#..
.....#..	