Nowruz (DNK)

It is few days to Nowruz (Persian new year), and grandpa has invited his family to his garden. There are k kids among the guests. To make the gathering more fun for the kids, grandpa is going to run a game of hide-and-seek.

The garden can be represented by an $m \times n$ grid of unit cells. Some (possibly zero) cells are blocked by rocks, and the remaining cells are called *free*. Two cells are called *neighbors* if they share an edge. That is, each cell has up to 4 neighbors: two in the horizontal direction, and two in the vertical direction. Grandpa wants to turn his garden into a maze. For this purpose, he can block some free cells by planting bushes in them. The cells where he plants the bushes are no longer free.

A maze must have the following property. For each pair a and b of free cells in the maze there must be exactly one *simple path* between them. A simple path between cells a and b is a sequence of free cells in which the first cell is a, the last cell is b, all cells are distinct, and each two consecutive cells are neighbors.

A kid can hide in a cell if and only if that cell is free and has *exactly* one free neighbor. No two kids can hide in the same cell.

You are given the map of the garden as input. Your task is to help grandpa create a maze in which many kids can hide.

Implementation details

This is an output-only task with partial scoring. You are given 10 input files, each describing grandpa's garden. For each input file you should submit an output file with a map of a maze. For each output file you will get points based on the number of kids that can hide in your maze.

You are not supposed to submit any source code for this task.

Input format

Each input file describes one grid representing a garden and gives the number of kids k invited by grandpa. The format is as follows:

- line 1: m n k
- line $1 + i$ (for $1 \leq i \leq m$): row i of the grid, which is a string of length n, consisting of the following characters (without any whitespace):

'.': a free cell,
'#': a rock.

Output format

- line i (for $1 \leq i \leq m$): row i of the maze (the garden, after bushes are planted). It is a string of length n, consisting of the following characters (without any whitespace):

  '.': a free cell,

  '#': a rock,

  'X': a bush. (Note that the letter X must be in uppercase.)

Constraints

- $1 \leq m, n \leq 1024$

Scoring

An output file is considered to be *valid* if all the following conditions are met:

- The output map must match the input map with the only exception that arbitrarily many '.' characters in the input map can be changed to 'X' characters (cells blocked by bushes).
- The output map must have the property of a maze, as defined in the problem statement.

If your output for a test case is not valid, your score for that test case will be 0. Otherwise, the score will be $\min(10, 10 \cdot l/k)$ points, rounded down to two digits after the decimal point. Here, l is the number of kids that can hide in your output maze, and k is the number provided in the input. You will score 10 points for a test case if and only if your output is a maze in which k or more kids can hide. For each test case there exists a solution that scores 10 points.

Note that if your solution is valid but still scores 0 points according to the above formula, the grading verdict you will see in the CMS will be 'Wrong Answer'.

Example

Consider the following input:

```
4 5 5
....#
.#..#
...#.
....#
```

Below is a possible valid output:

```
.X.X#
.#..#
...#X
XX..#
```

Since $l = 4$ kids can hide in this maze, this solution will receive $10 \cdot 4/5 = 8$ points. The cells in which kids can hide are marked with `O` below:

```
OXOX#
.#.O#
...#X
XX.O#
```

The following three outputs are not valid:

```
.XXX#        ...X#        XXXX#
.#XX#        .#.X#        X#XX#
...#.        ...#X        ..X#X
XX..#        XXXX#        ..XX#
```

In the left output there is no simple path between the free cell in the top left corner and the free cell in the rightmost column. In the two other outputs, for each pair of distinct free cells there are exactly two distinct simple paths between them.