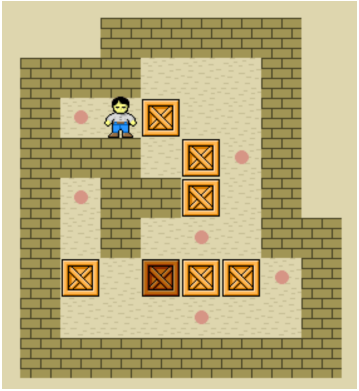
D Distribution Center

Time limit: 4s

Sokoban is the best employee at your town's biggest distribution center. He likes his job because it is simple, albeit a bit physically intensive. Everyday he is pushing crates to some possbile destinations from where they will be loaded in trucks. Unfortunately, Sokoban's youth is behind him, so he starts to feel the effects of the physical labour. Therfore, he came up with an algorithm to help him move the crates optimally: the Fast Pushing Crates (FPC) algorithm. For the algorithm to be complete, he needs a bit of help from you.



Sokoban pushing crates in the distribution center. CC-BY 3.0 By Carloseow at English Wikipedia.

His algorithm receives as input the layout of the distribution center, as a grid, and returns the steps he needs to take to efficiently push the crates. In his algorithm, he needs to find out which squares in the grid are *dead squares*. We call a square a dead square if it is a wall or if it is impossible to push a crate from that square to any of the destinations (even if Sokoban could teleport to any location).

Given the layout of the distribution center, help Sokoban find out which squares are dead squares.

Input

The input consists of:

- A line with two integers r and c ($1 \leq r, c \leq 10^3$), the number of rows and columns of the grid.
- r lines, each containing c characters, where:
  - A '`#`' represents a wall. It is guaranteed that the grid is surrounded by walls.
  - A '`D`' represents a destination. There can be any number of them, including 0.
  - A '`.`' represents an empty square.

Output

Output r lines, each containing c characters, where the ith character of the jth line is:

- '`X`' if the square at position (i, j) is a dead square.
- '`O`' otherwise.

Sample Input 1

```
7 7
#######
#.....#
#.....#
#..D..#
#.....#
#.....#
#######
```

Sample Output 1

```
XXXXXXX
XXXXXXX
XXOOOXX
XXOOOXX
XXOOOXX
XXXXXXX
XXXXXXX
```

Sample Input 2

```
8 8
########
#..#D###
#..#.###
#..#.###
#..#...#
#..#.###
#D.#.###
########
```

Sample Output 2

```
XXXXXXXX
XXXXOXXX
XOXXOXXX
XOXXOXXX
XOXXOOXX
XOXXOXXX
XOXXXXXX
XXXXXXXX
```