https://course.acciojob.com/idle?question=8c4f7d7e-b6bd-48ec-87c3-c29387675987

- MEDIUM
- Max Score: 40 Points

Box Rotation

You are given an m x n matrix of characters box representing a side view of a box. Each cell of the box is one of the following:

A stone #

A stationary obstacle *

Empty .

The box is rotated 90 degrees clockwise, causing some of the stones to fall due to gravity. Each stone falls down until it lands on an obstacle, another stone, or the bottom of the box. Gravity does not affect the obstacles' positions, and the inertia from the box's rotation does not affect the stones' horizontal positions.

It is guaranteed that each stone in the box rests on an obstacle, another stone, or the bottom of the box.

Return an n x m matrix representing the box after the rotation described above.

Input Format

Input consists of integer M and N followed by M lines each containing N characters

Output Format

Print the new matrix.

EXAMPLE 1

Input

```
1 3
# . #
```

Output:

```
.
#
#
```

Explanation

EXAMPLE 2

Input

```
2 4
# . * .
# # * .
```

Output:

```
# .
# #
* *
. .
```

Constraints

m == box.length

n == box[i].length

1 <= m, n <= 400

box[i][j] is either '#', '*', or '.'

Topic Tags

- 2-Pointers
- 2D-Arrays

My code

```
// n java
import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int m = sc.nextInt(), n = sc.nextInt();
        char[][] mat = new char[m][n];
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++)
                mat[i][j] = sc.next().charAt(0);
        }
        sc.close();
        Solution ob = new Solution();
        char[][] new_matrix = ob.rotateTheBox(mat);
        for (int i = 0; i < new_matrix.length; i++) {
            for (int j = 0; j < new_matrix[i].length; j++)
                System.out.print(new_matrix[i][j] + " ");
            System.out.println();
        }
    }
```

```
}
class Solution{
    public static char[][] rotateTheBox(char[][] box) {
        //Write Code here
           // Aprotch
           //1 sr rotate 90 digree
           //2 fore eatch element,if element is #
           //then searth boton "." and swap
            int m = box.length, n = box[0].length;
         char[][] ans = new char[n][m];
         // rotate first, then drop
         for (int i = 0; i < m; i++) {
             for (int j = 0; j < n; j++) {
                 ans[j][i] = box[m-1-i][j];
             }
         }
        //now rotation is compleate;
          //and now matrix is n X m; not m X n;
              for (int i = n - 1; i >= 0; i--) {
             for (int j = 0; j < m; j++) {
                 if (ans[i][j] == '#') {
                     int curRow = i;
                     while (curRow+1 < n && ans[curRow+1][j] == '.') {
                         curRow++;
                     }
                     if (curRow != i) {
                             //here swaping
                         ans[curRow][j] = '#';
                         ans[i][j] = '.';
                     }
```

```
                }
            }
        }
        return ans;
    }
}
```