My Beautiful Garden

You have just bought a beautiful house in the countryside. You really love your new house and plan to decorate your house so that it will be the most beautiful house people have ever seen. As a start, you want to have a beautiful garden, in which you want to have as many beautiful plants and accessories as possible.

You already have a large rectangular plot of land with R rows and C columns located at the back of your house. However, the plot of land is now empty without any plants, let alone accessories. As you gaze upon the empty plot of land, you envision a beautiful garden with a beautiful water fountain in the middle, with a lot of colorful flowers surrounding it, complete with a paved walkway along it.

Now, it is time to head back to reality. You realize that the plot of land contains many mines, which may explode if you decide to plant something on it. You have been given a map by the previous owner of the land, with the map denoting the place where the mines are located. You are interested in figuring out how many sub-plot of lands in your backyard that you can use without the danger of any mines exploding. A sub-plot of land is defined as a square of any size, as long as it fits inside your backyard.

Input

The first line contains two integers R (1 <= R <= 500) and C (1 <= C <= 500), separated by a single space, denoting the size (in row by column) of your plot of land located at your backyard. R lines follow. In each line, there are C characters with no space separating them. Each of the C characters can be either "." or "o", denoting an empty plot of land and a plot of land containing a mine respectively.

Output

Print the number of sub-plot of land that does not contain any mines in your backyard. Your output should contain a newline character.

Sample Input 1

```
5 5
ooooo
ooooo
ooooo
ooooo
ooooo
```

Sample Output 1

```
0
```

Sample Input 2

```
3 4
o.o.
..o.
..o.
```

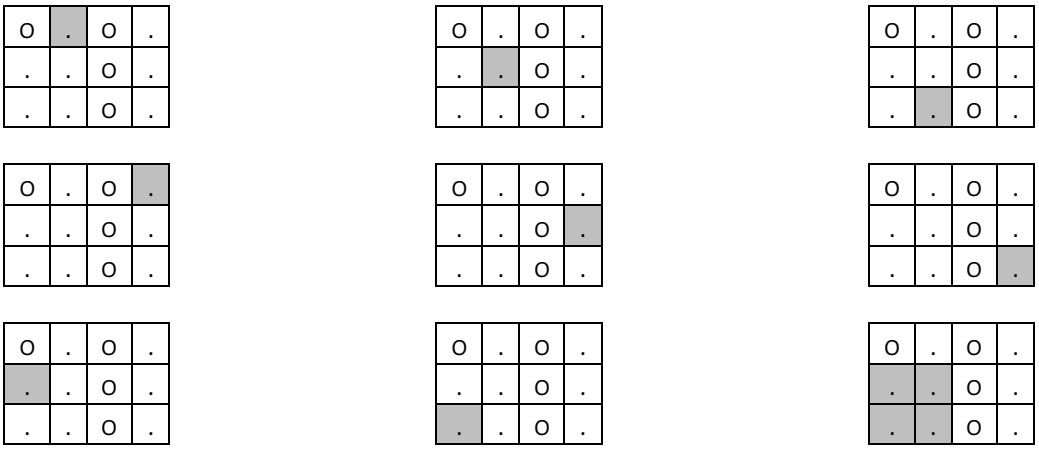
Sample Output 2

```
9
```

Explanation

In the first sample input, every plot of land contains a mine. Therefore, there are 0 square sub-plot of land that you can plant on.

In the second sample input, there are nine possible sub-plot of lands illustrated in the following series of diagrams:



Skeleton

You are given the skeleton file **`Garden.java`**. You should see the contents below when you open the file, otherwise you might be working in the wrong directory.

```
/**
 * Name          :
 * Matric No.    :
 * PLab Acct.    :
 */

public class Garden {

    public void run() {
        // treat this as your "main" method
    }

    public static void main(String[] args) {
        Garden myBeautifulGarden = new Garden();
        myBeautifulGarden.run();
    }
}
```

Notes

1. You should develop your program in the subdirectory **ex1** and use the skeleton file provided.
2. You are free to use anything to solve this problem.
3. You will get:
   - **90%** of the total marks you receive if your algorithm runs in **$O(N^5)$.**
   - **100%** of the total marks you receive if your algorithm runs in **$O(N^4)$.**
   - **110%** of the total marks you receive if your algorithm runs in **$O(N^3)$.**

   with N being the maximum of R and C.
4. Please be reminded that the marking scheme is:

   Input : 10%
   Output : 10%
   Correctness : 50%
   Programming Style : 30% (awarded if you score **at least 20% from the above**):
   - Meaningful comments (pre- and post- conditions, comments inside the code): 10%
   - Modularity (modular programming, proper modifiers [public / private]): 10%
   - Proper Indentation: 5%
   - Meaningful Identifiers (for both method and variable names): 5%

   Compilation Error : Deduction of **50% of the total marks obtained**.