Find X

Your friend gave you a treasure map. As you might know, in treasure maps, “X marks the spot”. However, there are a lot of X’s in this treasure map! Your friend told you that the spot is marked by a ‘super X’, which is multiple X that are arranged in an X shape.

The map is divided into RxC squares, arranged in R rows and C columns. In each square, the map can be either empty (denoted by a dot ‘.’) or have an X (denoted by the letter ‘X’). A ‘super X’ is defined as a square of size at least 3 that is composed entirely by X on both diagonals.

For example,

```
X...X  XX..X  X.X.X  XXXXX
.X.X.  X.X.X  .X.X.  X.XX.X
..X..  ..X..  X.X.X  XX.XXX
.X.X.  X.XX.  .X.X.  X.XX.X
X...X  X.X.X  X.X.X  XXXXX
```

All four of those are considered ‘super X’, since the diagonals are composed entirely of the letter X.

Meanwhile,

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

The first matrix is not considered a ‘super X’, because one X is missing from one of the diagonals.
The second and third matrix is not a ‘super X’, because the size of the squares are less than 3.

You want to count, how many ‘super X’ are there in the treasure map?

Input

The first line contains two numbers R and C, that signifies the number of rows and columns in the treasure map, respectively.

The next R line contains C characters each. Each line represents one row in the treasure map. The characters are either '.' or 'X'.

Output

The number of 'super X' in the map.

Sample Input

```
7 9
X...X.X..
.X...X...
..X.X.X..
...X.....
..X.X...X
.X...X...
X.....X..
```

Sample Output

4

Explanation

X				X		X		
	X				X			
		X		X		X		
			X					
		X		X				X
	X				X			
X						X		

Skeleton program

```
/**
 * Name        :
 * Matric no.  :
```

```
 * Plab account:
 */

import java.util.*;

class Findx {
      public static void main(String[] args) {
            // declare the necessary variables
            // declare a Scanner object to read input
            // read input and process them accordingly
      }
}
```

Note:

1. You should develop your program in the subdirectory, ex1 and use the java file provided. You should not create new file or rename the file provided.
2. You don't have to use OOP in this sit in lab. You are allowed to add more methods inside each file.
3. Please be reminded that the marking scheme is
   Input:10%, Output:10%, Programming Style:30% and Correctness: 50%