Pattern Matching

You have a big piece of paper. Written on the paper are a RxC matrix of letters, where each cell has a single uppercase letter on it.

From this big piece of paper, you want to cut a smaller, rectangular piece of size R_p x C_p with a certain pattern on it. However, you haven't decided the complete pattern: you might not care about some cells in the pattern. For these cells, any letter can be used.

You want to know, in how many ways can you get this smaller piece?

Input

The first line contains R and C, the number of rows and columns in the original piece of paper
The next R lines contains the matrix in the original piece of paper.
The next line contains R_p and C_p, the size of the smaller piece that you want.
The next R_p lines contains the pattern. An asterisk '*' is used if any letter can be used for that cell.

Output

The number of ways that you can cut this smaller piece.

Sample input

```
6 2
AAAXAA
CCDXCX
2 2
AA
C*
```

Sample Output

```
3
```

Explanation

There are three ways to get the pattern:

```
AA----
CC----
```

```
-AA---
-CD---

----AA
----CX
```

Constraint

R, C, R_p, and C_p will be between 1 and 50 inclusive.

R_p will be less than R and C_p will be less than C.

Skeleton program

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

import java.util.*;

class Matching {
     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%