COCI '10 Contest 3 #6 Mono

Time limit: 5.0s **Memory limit:** 64M

Mirko soon realised that number sequences are not the best career choice, and went right back to letter table business.

Mirko's table has R rows and C columns and consists of lowercase letters.

Each cell of the table is a square of equal size. We assign coordinates to vertices of those squares, so that the upper-left corner of the table has coordinates $(0, 0)$, the upper-right $(C, 0)$, the lower-left $(0, R)$, and the lower-right (C, R).

We say that a polygon within the table is **monoliteral** if the following holds:

1. its vertices are from the described set of cell-square vertices,
2. its edges are parallel to the coordinate axes,
3. all letters inside the polygon are equal.

A simple polygon for which the first two conditions are true (the third one may or may not be true) is given. Mirko would like to know the number of **monoliteral** polygons that can be obtained by **moving** the given one up, down, left, or right or any combination thereof, but **not rotating**.

Input Specification

The first line of input contains two space separated integers R and C $(1 \leq R, C \leq 500)$.

Each of the next R lines contains exactly C lowercase letters, this is Mirko's table.

The following line contains integer V $(4 \leq V \leq 500)$, number of vertices of the given polygon.

Each of the next V lines contains two integers X, Y $(0 \leq X \leq C, 0 \leq Y \leq R)$. These are the coordinates of the vertices of the given polygon. Vertices are given in clockwise order.

The given polygon will satisfy conditions 1 and 2 from above.

Output Specification

In the first and only line of output, print the expected number of polygons.

Scoring

In test cases worth 40% of total points, R, C and V will not exceed 20.

In test cases worth 70% of total points, V will not exceed 20.

Sample Input 1

```
3 3
aaa
aaa
aaa
4
2 0
2 2
0 2
0 0
```

Sample Output 1

```
4
```

Sample Input 2

```
3 3
aaa
aba
aaa
4
2 0
2 2
0 2
0 0
```

Sample Output 2

```
0
```

Sample Input 3

```
5 4
xyyx
xyyy
xxyy
xxxx
xxxx
8
1 3
1 2
0 2
0 0
2 0
2 1
3 1
3 3
```

Sample Output 3

```
2
```