

C. New Year and Domino

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

They say "*years are like dominoes, tumbling one after the other*". But would a year fit into a grid? I don't think so.

Limak is a little polar bear who loves to play. He has recently got a rectangular grid with h rows and w columns. Each cell is a square, either empty (denoted by '.') or forbidden (denoted by '#'). Rows are numbered 1 through h from top to bottom. Columns are numbered 1 through w from left to right.

Also, Limak has a single domino. He wants to put it somewhere in a grid. A domino will occupy exactly two adjacent cells, located either in one row or in one column. Both adjacent cells must be empty and must be inside a grid.

Limak needs more fun and thus he is going to consider some queries. In each query he chooses some rectangle and wonders, how many ways are there to put a single domino inside of the chosen rectangle?

Input

The first line of the input contains two integers h and w ($1 \leq h, w \leq 500$) — the number of rows and the number of columns, respectively.

The next h lines describe a grid. Each line contains a string of the length w . Each character is either '.' or '#' — denoting an empty or forbidden cell, respectively.

The next line contains a single integer q ($1 \leq q \leq 100\,000$) — the number of queries.

Each of the next q lines contains four integers $r1_i, c1_i, r2_i, c2_i$ ($1 \leq r1_i \leq r2_i \leq h, 1 \leq c1_i \leq c2_i \leq w$) — the i -th query. Numbers $r1_i$ and $c1_i$ denote the row and the column (respectively) of the upper left cell of the rectangle. Numbers $r2_i$ and $c2_i$ denote the row and the column (respectively) of the bottom right cell of the rectangle.

Output

Print q integers, i -th should be equal to the number of ways to put a single domino inside the i -th rectangle.

Examples

input
5 8 ...#...# .#..... ##.#.... ##..#.# 4 1 1 2 3 4 1 4 1 1 2 4 5 2 5 5 8
output
4 0 10 15

input
7 39###...###...###...###...###...

```
...#.#.#.#.#.....#.#.#.#.#...
.###.#.#.#.#.###.....###.#.#.#.###.
.#.....#.#.#.#.#.....#.....#.#.#.#.#.
.###.###.#.###.....###.###.#.###.
.....
6
1 1 3 20
2 10 6 30
2 10 7 30
2 2 7 7
1 7 7 7
1 8 7 8
```

output

```
53
89
120
23
0
2
```

Note
A red frame below corresponds to the first query of the first sample. A domino can be placed in 4 possible ways.