Problem H
Sorting Machine

Adrian knows that Morgan the robot is capable of sorting texts, but he is uncertain about Morgan's efficiency in doing the task. Adrian decides to give Morgan a test on his efficiency.

First, Adrian gives Morgan a list of N equal-length texts, numbered from 1 to N. Each text is a string S_i that contains M characters, indexed from 1 to M. S_{ij} represents character j in string S_i.

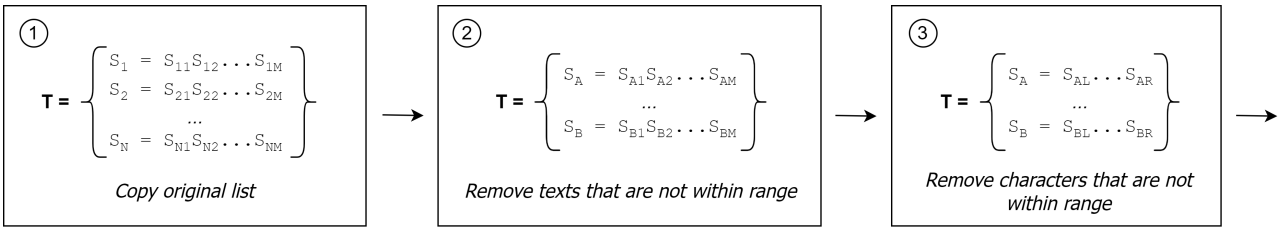
Adrian will give Morgan Q tasks. Each task is represented by a tuple $\langle A, B, L, R, X \rangle$ satisfying the following.

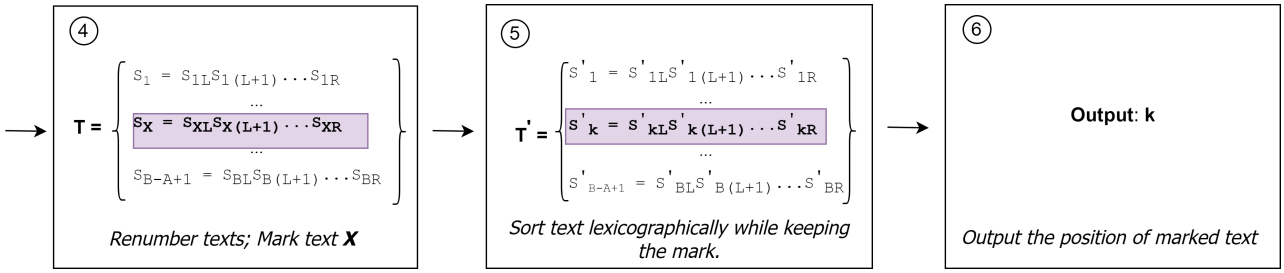
- $1 \leq A \leq B \leq N$
- $1 \leq L \leq R \leq M$
- $1 \leq X \leq B - A + 1$

For each task, Morgan should perform the following procedures.

1. Copy the original list of texts; let T be the copied list. This list will be updated throughout the task.
2. Remove all texts i from T that are **not** within the range of $A \leq i \leq B$.
3. For all remaining texts in T, remove all characters at index j that are **not** within the range of $L \leq j \leq R$;
4. The remaining texts in T is renumbered from 1 to $B - A + 1$. Mark text X in T.
5. Sort T lexicographically; let the result be T'. Note that the performed sort is a stable sort, meaning that if two texts are equal, then they maintain their order in the sorted list.
6. Output the position of the marked text in T'. The lexicographically smallest text will be at position 1 (one-based).

The image below are the ilustrations how the procedure works.





It turns out that it takes Morgan a lot of time to solve those tasks. Therefore, Adrian asks for your help to improve Morgan's program so that he can solve those tasks quickly and accurately.

A string s of length n is lexicographically smaller than string t with the same length if there exists an integer $1 \leq i \leq n$ such that $s_j = t_j$ for all $1 \leq j < i$, and $s_i < t_i$.

Input

Input begins with two integers N M ($1 \leq N, M \leq 100\,000$; $1 \leq N \times M \leq 100\,000$) representing the number of texts and the length of each text, respectively. Each of the next N lines contains a string S_i representing text i. Each text contains M lower-case characters.

The next line contains an integer Q ($1 \leq Q \leq 100\,000$) representing the number of tasks. Each of the next Q lines contains five integers A B L R X ($1 \leq A \leq B \leq N$; $1 \leq L \leq R \leq M$; $1 \leq X \leq B - A + 1$) representing a task.

Output

For each task, output an integer in a single line representing the answer of that task.

Sample Input #1

```
5 6
adrian
morgan
george
undine
stella
5
1 5 1 6 1
1 5 1 6 2
1 2 3 6 1
2 4 3 5 3
1 2 5 6 2
```

Sample Output #1

```
1
3
```

```
2
1
2
```

Explanation for the sample input/output #1

For tasks 1 and 2, the final T is all of the given texts. Task 1 marks text 1 of T, which is `adrian`; while task 2 marks text 2 of T, which is `morgan`. After sorted, the list T' becomes [`adrian`, `george`, `morgan`, `stella`, `undine`]. The marked text in task 1 is at position 1 in T'. Similarly, the marked text 2 is at posiition 3 in T'.

For task 3, the final T is [`rian`, `rgan`]. Task 3 marks text 1 of T, which is `rian`. After sorted, the list T' becomes [`rgan`, `rian`]. The marked text in task 3 is at position 2 in T'.

For task 4, the sorted list T' contains [`din`, `org`, `rga`]. Task 4 marks text 3 of T, `din`, which is at position 1 in T'.

For task 5, the sorted list T' contains [`an`, `an`]. Note that both `an` are different from each other; the first one is taken from text 1 of T, while the second one is taken from text 2 of T. Task 5 marks text 2 of T, `an` taken from text 2 of T, which is at position 2 of T'.

Sample Input #2

```
3 9
indonesia
nationaln
contestco
3
2 2 1 9 1
1 2 3 7 2
1 3 2 5 2
```

Sample Output #2

```
1
2
1
```