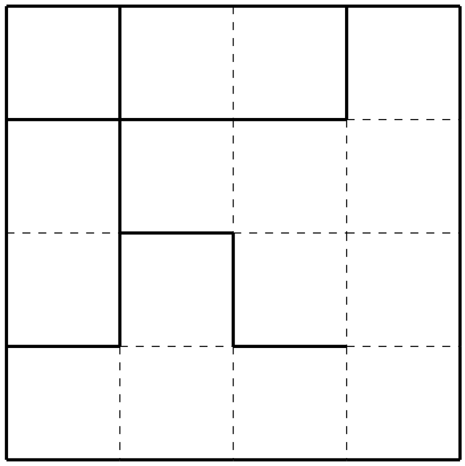
Board Bicoloring

You are given a board with r rows and c columns, divided into $r \cdot c$ cells. The rows are numbered from 1 to r and the columns are numbered from 1 to c. We denote by (i, j) the cell at the i-th row and j-th column. Two cells are neighbors if and only if they share an edge.

We paint some edges between neighboring cells.



We define a path from a cell (i_1, j_1) to cell (i_k, j_k) as a sequence of cells $(i_1, j_1), (i_2, j_2), \ldots (i_k, j_k)$ where two consecutive cells are neighbors and their common edge is not painted.

We define a connected component as a set of cells where there is a path between any pair of cells in the set, and it is not possible to add any cell to this set without losing this property.

Now we want to color the board using two colors: Red and Green, so that two adjacent cells on two different connected components have different colors. We call such coloring **Bicoloring**. Note that a **Bicoloring** may not exist.

You need to process q queries of the following three types:

1. `H i j` if the common edge between cells (i, j) and $(i + 1, j)$ is not painted, we paint it. Otherwise, we erase the paint.
2. `V i j` if the common edge between cells (i, j) and $(i, j + 1)$ is not painted, we paint it. Otherwise, we erase the paint.
3. `? i j` we consider three cases:
   - if there exists a pair of neighboring cells anywhere in the board that are in the same connected component, where their common edge is painted, print one line containing the string `Oh no!`,
   - otherwise, if there is no **Bicoloring** for the board, print one line containing the string `Cannot!`,
   - otherwise, consider all possible **Bicoloring**s where the cell $(1, 1)$ is Red, find all possible colors of cell (i, j). If this cell can be painted by either color, print `GR`. If this cell can only be painted red, print `R`, and if this cell can only be painted green, print `G`.

Input

The first line of the input contains two positive integers r and c ($1 \le r \cdot c \le 3 \cdot 10^5$). The next $r + 1$ lines describe the initial board according to the following rules:

- Each line contains $2 \cdot c + 1$ characters, numbered starting from 1. In each line:
  - All even-indexed characters can be either a space (`' '`) or an underscore (`'_'`).
  - All odd-indexed characters can be either a space (`' '`) or a pipe (`'|'`).
- All even-indexed characters in the first line and the last line are underscores, which demonstrate the top and the bottom borders of the board.
- Every underscore in the input, except the ones mentioned above, represents a horizontal painted edge between two consecutive cells in the same column.
- The first and the last characters of all lines (except the first one) are pipes, which demonstrate the left and the right borders of the board.
- Every pipe in the input, except the ones mentioned above, represents a vertical painted edge between two consecutive cells in the same row.

The next line contains a single integer q ($1 \le q \le 3 \cdot 10^5$) — the number of queries. The next q lines describe the queries, each line can be one of the three types, as described above:

1. `H i j` ($1 \le i < r$, $1 \le j \le c$),
2. `V i j` ($1 \le i \le r$, $1 \le j < c$),
3. `? i j` ($1 \le i \le r$, $1 \le j \le c$).

Output

For each query of type 3, print a single line as described above.

Sample Input 1

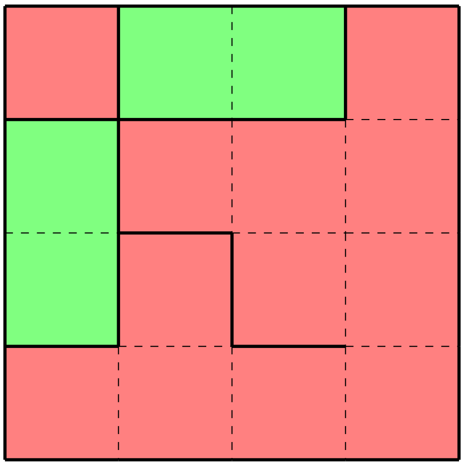
```
4 4
 _ _ _ _ 
|_|_ _| |
| |_    |
|_| |_  |
|_ _ _ _|
8
? 4 4
H 3 4
? 4 4
H 2 2
V 3 2
H 3 3
H 3 4
? 4 4
```

Sample Output 1

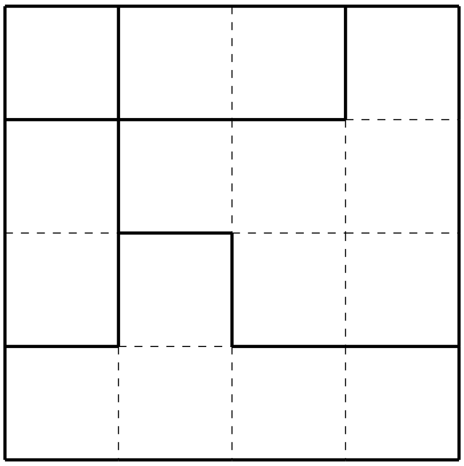
```
Oh no!
Cannot!
R
```

Sample Explanation

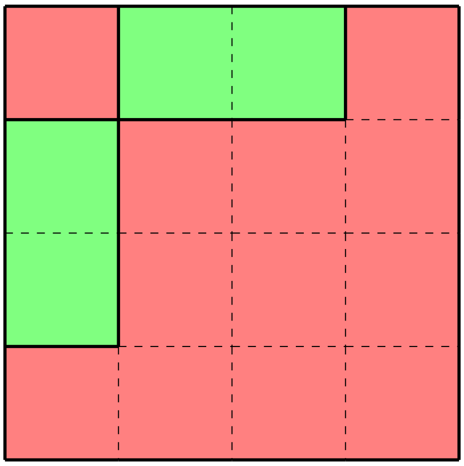
For the first query of type 3 (the 1-st query), the cells $(3,3)$ and $(4,3)$ belong to the same connected component, but their common edge is painted.



Before the second query of type 3 (the 3-rd query), the board is as follows. It can be shown that there exists no **Bicoloring**.



For the third query of type 3 (the 8-th query), the only possible **Bicoloring** is as follows:



This page is intentionally left blank.