Counting Contours

Please refer to the judging system for the colored illustrations.

A recursive island, is an island that lies within a lake (which may itself be inside an island). A recursive lake, is a lake that lies within an island (which may itself be inside a lake).

Bash is fascinated with this idea. He wants to create some drawings with islands and lakes, where some can potentially be recursive.

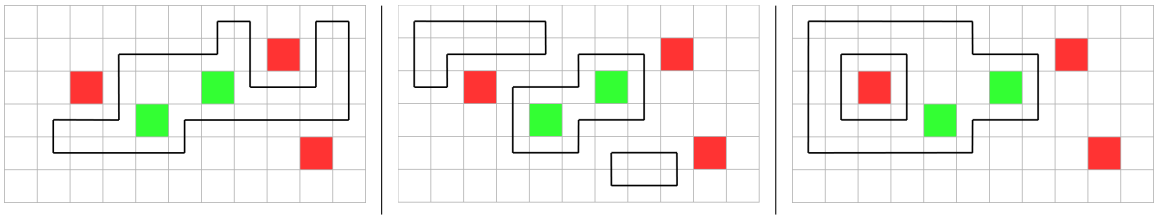
Bash has a grid with R rows and C columns, divided into $R \times C$ cells. Bash then colors some cells red, and some cells green.

Now bash wants to draw some non-intersecting *contours* on this grid. A contour is a border between land and water, and must satisfy the following conditions:

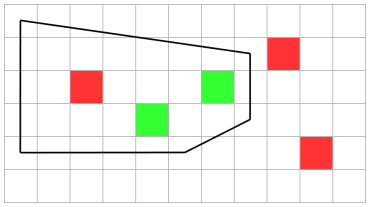
- It must be a polygon, with all vertices being centers of some non-colored grid cells.
- All edges are axis-aligned.
- It doesn't self intersect.
- No edge intersects with any colored cells.

Note that it is possible for one contour to be completely inside another. However, no point can lie on the borders of more than one contours.

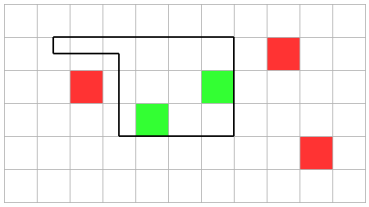
Here are some examples of **valid** way to draw contours:



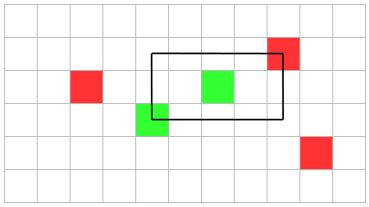
Here are some examples of **invalid** way to draw contours:



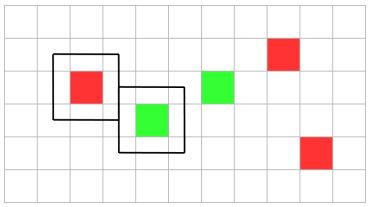
Some edges are not axis-aligned.



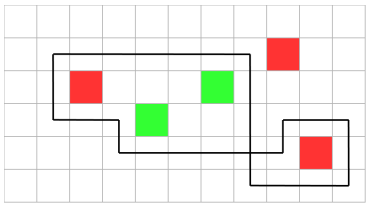
Some vertices are not grid cells' centers.



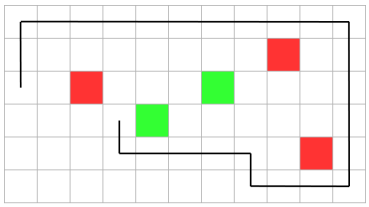
A contour touches colored cells.



Two contours intersect each others.



A contour intersects itself.



This is not a polygon.

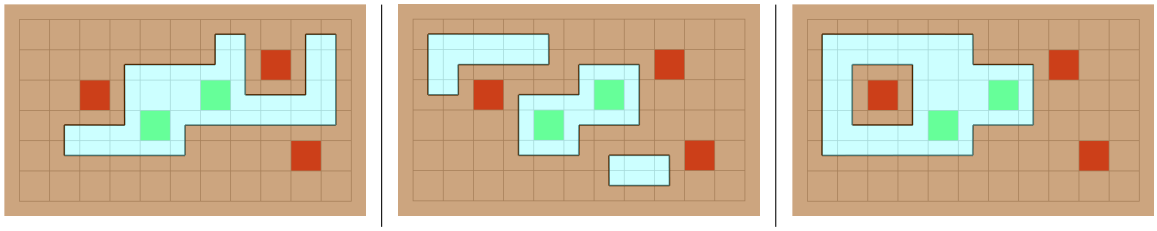
After drawing all contours, Bash wants to draw islands and lakes in between the contours such that:

- For each contour, the parts inside and outside of it must be different (one must be water and the other must be land).
- The parts touching the border of the grid must be land.

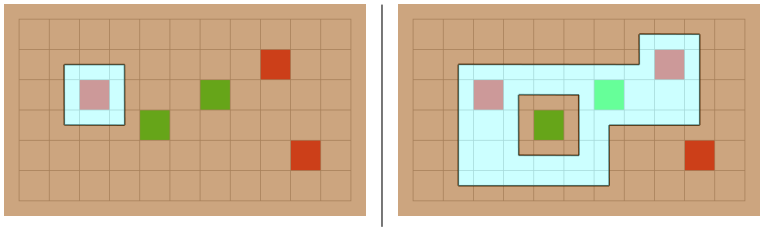
A drawing is considered **valid** iff:

- All red cells are land.
- All green cells are water.

Here are some **valid** drawings after filling in land and water:



Here are some **invalid** drawings after filling in land and water:



Please count the number of **valid** drawings. Two drawings are considered different iff there exists some part of the grid that is land in one way, and is water in the other.

Input

Each test contains multiple test cases. The first line contains the number of test cases τ ($1 \leq \tau \leq 10^4$). The description of the test cases follows.

- The first line contains three integers: R, C and k ($1 \leq R \leq 12$; $1 \leq C \leq 100$; $0 \leq k \leq R \cdot C$), where k is the number of colored cells.
- In the next k lines, each line contains two integers r, c and a string s ($1 \leq r \leq R$; $1 \leq c \leq C$, s is either `red` or `green`) – meaning that the cell (r, c) is colored by s. It is guaranteed that all k cells are distinct.

It is guaranteed that the sum of $R \cdot C$ over all test cases does not exceed 1200.

Output

For each test case, print one line containing the number of valid drawings, modulo $10^9 + 7$.

Sample Input 1	Sample Output 1
2 3 3 1 2 2 green 3 3 1 2 2 red	1 1