Nurikabe Checker

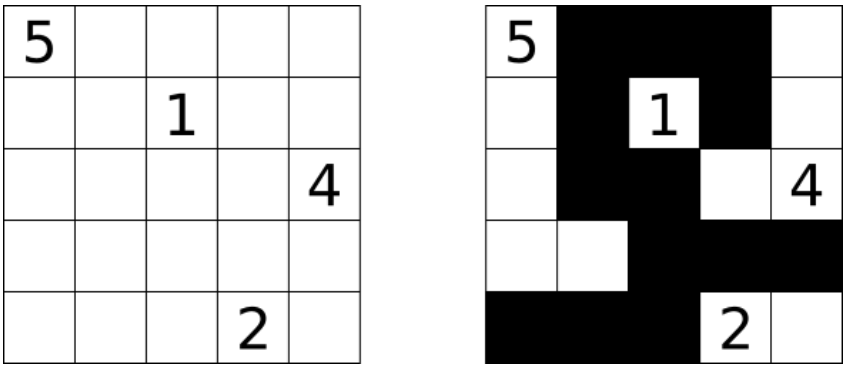
Filename: `nurikabe`

Evan loves solving logic puzzles. One type of puzzle that he finds particularly interesting is called Nurikabe.

In a Nurikabe puzzle, you are initially given a grid where the cells are either empty or contain a positive integer. To solve the puzzle, you must fill in some of the cells. Cells that are filled in are considered black, while cells that aren't filled in are considered white. Two cells of the same color are connected if they are directly or indirectly connected, that is, if they share a side horizontally or vertically, or if there is a path of directly connected cells from one to the other. Each group of connected white cells is called an *island*, and each group of connected black cells is a *stream*. Furthermore, a correctly solved puzzle must satisfy the following constraints:

- Cells with numbers cannot be filled in.
- Each island must contain exactly one cell with a number, and that number must be equal to the number of cells in the island.
- Every black cell must be connected directly or indirectly with every other black cell (i.e. there can only be one stream).
- There can be no 2x2 squares of black cells (called *pools*) in the stream.

The figure below shows the Nurikabe puzzle given in the first sample case (left) and its correct solution (right).



Evan worked on lots of Nurikabe puzzles before he realized that he didn't completely understand the rules! He needs help deciding if his completed puzzles are correct, or if he needs to try again.

The Problem:

Given a completed Nurikabe puzzle, determine if it is solved correctly.

The Input:

The first line of input contains a single, positive integer, p, representing the number of puzzles. Following this line are p descriptions of completed puzzles. Each puzzle description begins with a line with three integers, n, m and k ($1 \leq n \leq 300$; $1 \leq m \leq 300$; $1 \leq k \leq n * m$), representing the number of rows, the number of columns, and the number of numbered cells, respectively. Rows are numbered beginning from top to bottom, and columns are numbered from left to right. Following this are n lines, each containing m characters. These lines represent the completed puzzle, where the characters are one of the following:

- "`.`": A white cell
- "#": A black cell

The following k lines describe the white cells with numbers. Each of these lines contains three integers, r_i, c_i and v_i ($1 \leq r_i \leq n$; $1 \leq c_i \leq m$; $1 \leq v_i \leq n * m$), representing the row number, column number, and value of the i^{th} numbered cell, respectively. No two numbered cells will be given at the same position in the puzzle, and all numbered cells will be associated with a white cell. As a consequence of these constraints, the input will contain at least k white cells.

The Output:

For each puzzle, output "`Puzzle #`*i*`: `" where i is the puzzle number, starting at 1. Then output "`Correct`" if the puzzle is correctly solved, or "`Incorrect`" otherwise.

(Sample Input and Sample Output are on following page)

Sample Input:

```
2
5 5 4
.###.
.#.#.
.##..
..###
###..
2 3 1
1 1 5
3 5 4
5 4 2
5 6 4
#####.
.##...
.#.##.
.#..##
.#.##.
1 6 6
4 1 4
4 3 4
5 6 1
```

Sample Output:

```
Puzzle #1: Correct
Puzzle #2: Incorrect
```