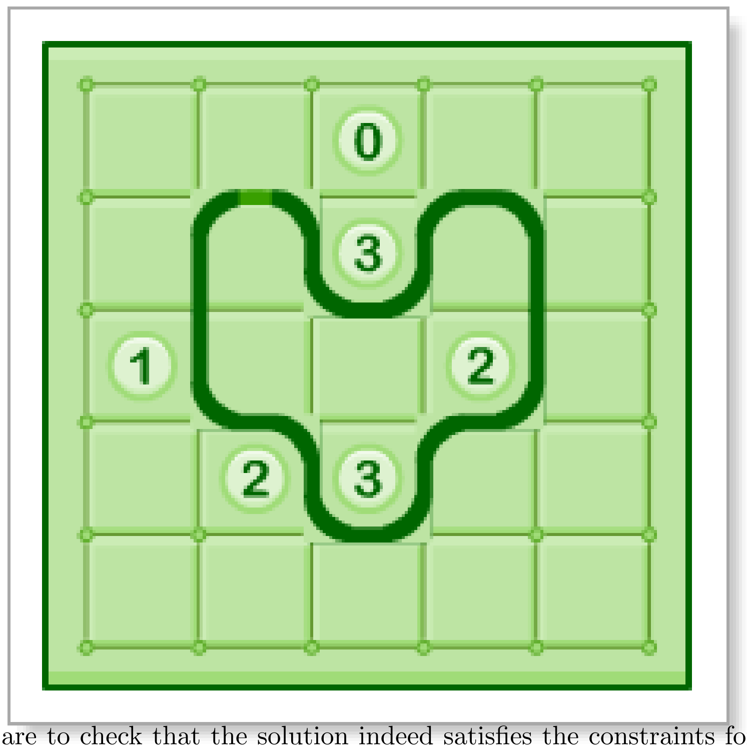
The puzzle game Slitherlink is played on a grid. Each cell of the grid contains either a space or an integer from 0 to 3. The goal is to draw in some grid lines connecting vertically or horizonally adjacent points on the grid subject to the following constraints:

1. If a grid cell is numbered k then exactly k out of the 4 adjacent grid lines are drawn.
2. The grid lines form a single loop that does not cross itself.



Determining if a given instance of a Slitherlink puzzle can be completed or not is NP-complete which means nobody knows of an efficient algorithm to solve the problem for large grids. Your task is simpler. Given an instance of a Slitherlink puzzle and a proposed solution, you are to check that the solution indeed satisfies the constraints for that puzzle.

For example, the first sample input is not valid since there are 2 grid lines drawn adjacent to a cell numbered 1. The second, third, and fourth samples are invalid since the grid lines do not form a single loop. However, the fifth sample is valid.

Input

The first integer denotes the number of test cases to follow. Each test case is specified by a line containing two positive integers R and C (both at most 50) followed by a grid of $2R + 1$ rows of $2C + 1$ characters. The contents of row r and column c (counting from 1) are as follows:

1. if both r and c are odd then the character is a '+' indicating a grid point
2. if both r and c are even then the character is either a space or one of 0, 1, 2, or 3 indicating the contents of the corresponding grid space
3. if r is odd and c is even then the character is either a space or '-' indicating a drawn horizontal grid line
4. if r is even and c is odd then the character is either a space or '|' indicating a drawn vertical grid line

You can assume all inputs are well-formed descriptions of a Slitherlink puzzle and a proposed solution. A blank line also precedes each test case.

Output

The output for each test case is a single line containing either 'Valid' or 'Invalid', depending on whether the proposed solution is valid or not.

Sample Input

```
5

3 3
+ +-+ +
 1| |2
+-+ +-+
|  1  |
+-+-+ +
 1  |3|
+ + +-+

2 3
+-+ +-+
|3| |3|
+ + + +
|3| |3|
+-+ +-+

2 2
+-+-+
|3|3|
+ + +
|3|3|
+-+-+

2 2
+ +-+
 2| |
+-+-+
| |2
+-+ +

3 4
+ +-+-+ +
  |   |2
+-+ + +-+
|3 0   3|
+-+ +-+-+
  | |  1
+ +-+ + +
```

Sample Output

```
Invalid
Invalid
Invalid
Invalid
Valid
```