The Forbidden Grid

Safou

**Problem Statement:**

Sudoku is a popular puzzle game where the goal is to fill a 9x9 grid with numbers from 1 to 9, such that each number appears exactly once in every row, column, and 3x3 subgrid. However, in this variation of the Sudoku challenge, there's an added complexity: **Forbidden Numbers**.

In this challenge, certain cells of the Sudoku grid come with restrictions, i.e., certain numbers are "forbidden" in specific cells. Your task is to fill the Sudoku grid such that all the standard Sudoku rules are followed **and** no forbidden numbers appear in their respective restricted cells.

**Objective:**

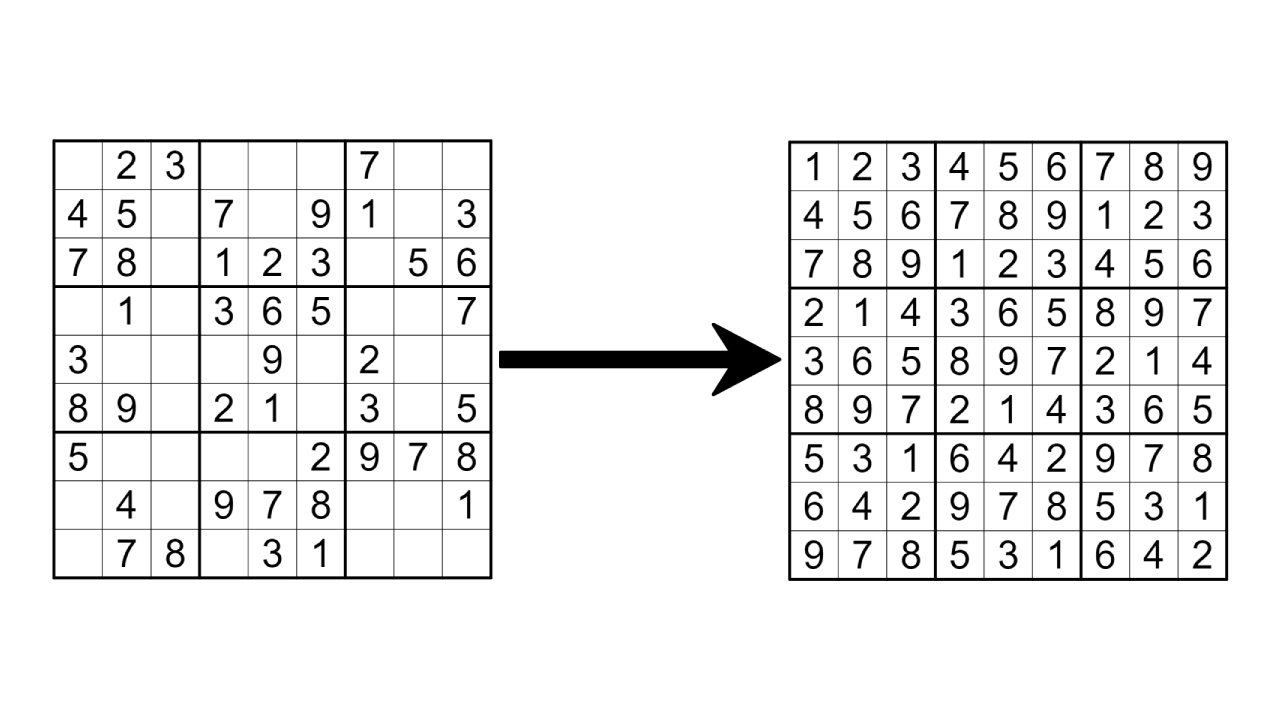
1. You are given a 9x9 Sudoku board with some pre-filled values.
2. Additionally, you're given a set of "forbidden numbers" for specific cells. These forbidden numbers cannot appear in the specified cells.
3. Your task is to solve the puzzle while respecting the forbidden constraints.

**Input:** You are given an T test cases, each one has:

* The first 9 lines represent the Sudoku grid. Each line consists of 9 integers, where 0 represents an empty cell and the non-zero numbers are the pre-filled values.
* The next line contains an integer m, which represents the number of forbidden constraints.
* The next m lines contain three integers: r, c, and num, where r and c represent the row and column (both 0-indexed) of a cell, and num is the forbidden number for that cell.

**Output:**

* You need to print the solved Sudoku grid, if a solution exists, while adhering to the forbidden numbers. If no solution is possible, print "No solution.".

****

**Example 1:**

Input :

1 2 3 0 0 6 7 8 9

4 5 0 0 8 0 1 2 3

7 8 0 0 2 3 4 0 0

2 0 4 0 6 5 8 9 0

0 6 5 8 0 7 2 1 4

8 9 7 2 0 4 3 6 5

5 0 1 6 4 2 0 7 8

6 0 2 9 7 8 0 3 1

0 7 8 5 3 1 6 4 0

9

1 4 7

5 2 8

6 5 2

3 6 3

5 4 1

2 0 7

6 2 2

4 4 5

8 3 1

Output :

No solution.

**Flag:**

For each grid, if it is solvable, represent it with "1"; if unsolvable, represent it with "0". After each test case, concatenate the result (“0” or “1”) in a binary string. Then, convert this binary string into its hexadecimal form (All caps), which will reveal the flag.

**Example 2:**

Input:

0 2 3 0 0 0 7 0 0

4 5 0 7 0 9 1 0 3

7 8 0 1 2 3 0 5 6

0 1 0 3 6 5 0 0 7

3 0 0 0 9 0 2 0 0

8 9 0 2 1 0 3 0 5

5 0 0 0 0 2 9 7 8

0 4 0 9 7 8 0 0 1

0 7 8 0 3 1 0 0 0

8

8 7 9

3 2 7

3 7 7

2 6 9

8 4 9

0 3 2

6 0 6

8 7 1

Output:

1 2 3 4 5 6 7 8 9

4 5 6 7 8 9 1 2 3

7 8 9 1 2 3 4 5 6

2 1 4 3 6 5 8 9 7

3 6 5 8 9 7 2 1 4

8 9 7 2 1 4 3 6 5

5 3 1 6 4 2 9 7 8

6 4 2 9 7 8 5 3 1

9 7 8 5 3 1 6 4 2