Sudoku

Given a partially filled 9 x 9 sudoku board, you need to output a solution to it.

Input

The first line contains **N** which is the number of Sudoku puzzles. **N** Sudoku puzzles follow. Each one contains 9 lines with each line containing 9 space separated integers. A 0 indicates an unfilled square, and an integer between 1 and 9 denotes a filled square with that value.

Output

Output to STDOUT the completely filled Sudoku board for each Sudoku. If there are multiple solutions, you can output any one.

Scoring

If your code output does not represent a valid filled Sudoku board, or if it is inconsistent with the given input, your solution will get a "Wrong Answer" judgement.

Otherwise, your score is the ratio (4000 - characters in the source code)/40. Any source code which solves all the testcases and has greater than 2000 characters will get a nominal score of 0.1. Your goal is to reach as close to 100 as possible.

Your code will be tested on multiple testcases. Each testcase will have N sudokus and your code should obey all the limits present in the environment to clear the testcase.

Your code will be judged as a wrong answer even if it fails any one of the testcase. Each sudoku is a different variant and a code that's optimal for one configuration needn't be for another.

Test Case Generation

You are guaranteed that there is at least one solution satisfying the input. All inputs are randomly generated, by filling in up to 25 random squares. If the grid generated isn't solvable, it is discarded and the procedure is repeated.

Sample Input

```
2
0 0 0 0 0 0 0 0 0
0 0 8 0 0 0 0 4 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 6 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0
0 0 0 0 0 0 2 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 8 0 0 0 0 4 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 6 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```

```
2 0 0 0 0 0 0 0 0
0 0 0 0 0 0 2 0 0
0 0 0 0 0 0 0 0 0
```

Sample Output

```
3 4 5 6 7 8 9 1 2
6 7 8 9 1 2 3 4 5
9 1 2 3 4 5 6 7 8
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 3 4 5 6 7 8 9 1
5 6 7 8 9 1 2 3 4
8 9 1 2 3 4 5 6 7
3 4 5 6 7 8 9 1 2
6 7 8 9 1 2 3 4 5
9 1 2 3 4 5 6 7 8
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 3 4 5 6 7 8 9 1
5 6 7 8 9 1 2 3 4
8 9 1 2 3 4 5 6 7
```

Task

Complete the function sudoku_solve that takes a 2-D array of sudoku grid as input.