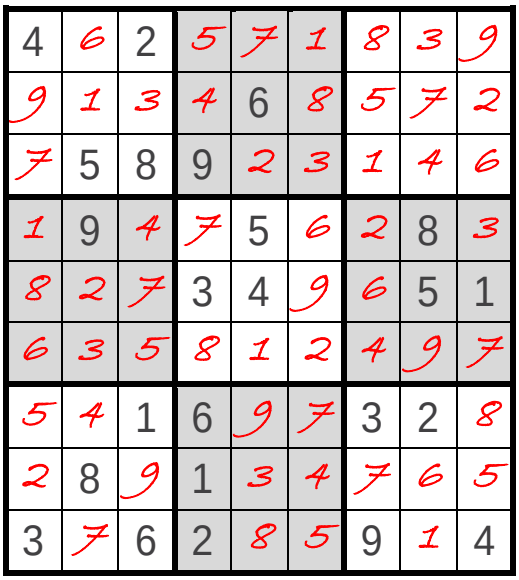
Problem 112: Sudoku

Difficulty: Hard

Author: Brett Reynolds, Orlando, Florida, United States

Originally Published: Code Quest 2019

Problem Background



Sudoku is a popular logic puzzle commonly found in newspapers, magazines, and online. Most likely originating in Indiana in 1979, the puzzle format found great popularity in Japan in the 1980s and became a worldwide phenomenon in the new millennium. Newspapers in particular contributed to the puzzle's establishment as a household name due to the puzzle's similarities with crossword puzzles.

Sudoku is played on a 9-by-9 grid of squares divided into 3-by-3 subgrids. Each square is filled in with one of the numbers from 1 to 9 inclusive, such that in the final solution any given digit appears exactly once within its row, column, and subgrid. The original puzzle is mostly blank, with only some numbers pre-filled as hints. The player must use these hints to determine how to fill in the remaining squares through process of elimination, logical deduction, and trial and error. In the image above, the bold black numbers are the original hints given by the puzzle; the italic red numbers are those filled in by the player to produce the solution. In order to be a "proper" Sudoku puzzle, a given set of hints must have one unique solution.

The properties of Sudoku puzzles have lent it to a great deal of study by mathematicians and computer scientists. Considerable research has been put into finding the minimum number of clues that can be given while still producing a unique solution (17), and into finding puzzles that follow certain patterns. Solving Sudoku puzzles efficiently is a somewhat difficult task in computer science; it falls into the category of problems known as "NP-complete." This means that it is believed that no algorithm exists that can solve a Sudoku puzzle in less than polynominal time (without having loops nested at least two deep).

Problem Description

You will need to write a program that can read a Sudoku puzzle and find its solution. Remember, to solve a Sudoku puzzle, you must fill in all the blank squares with a number between 1 and 9 inclusive, such that each number appears exactly once in each row, column, and 3-by-3 subgrid.

All of the puzzles your program will be given will be "proper" Sudoku puzzles; as stated above, this means that each puzzle will have exactly one valid solution.

Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include nine lines of text. Each line will contain only the digits from 1 through 9 inclusive and underscores (_). Underscores represent blank spaces in the puzzle that must be filled. Digits represent hints that should remain in place in the final solution.

```
2
4_2_____
____6____
_589_____
_9__5__8_
___34__51
_________
__16__32_
_8_1_____
3_62__9_4
_________
```

Sample Output

For each test case, your program must output the solved Sudoku puzzle, printing nine lines with nine digits per line.

```
___16__52
_7_5_4__6
39_______
62____39_
_______6_
9___3____
_5_71_94_
2__6__5_7
```

462571839
913468572
758923146
194756283
827349651
635812497
541697328
289134765
376285914
568327419
439168752
172594836
394286175
625471398
781953264
947835621
856712943
213649587