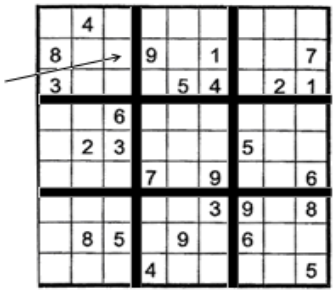
# Problem 26: Do You Su Doku?[1]




Source filename: `sudoku.(cpp|java)`
Input filename: `sudoku.in`
Output filename: `sudoku.out`

Su Doku is popular all over the world. Although it first became popular in Japan; its popularity has spread to encompass the entire globe. Most bookstores have several Su Doku puzzle books on their shelves. Your challenge for this problem is to write a program that will solve Su Doku puzzles.

There is really only 1 rule for solving a Su Doku puzzle: Fill in the grid so that every column, every row, and every 3x3 box contains the digits 1 through 9.

Since each column, row, and 3x3 box contains 9 squares, the digits must appear exactly once in each column, row, and 3x3 box. Consequently, one can deduce the contents of all 81 squares logically. Sometimes, it is easy to deduce the correct digit for an empty square. For example, in the puzzle shown above, the blank square in row 2 column 3 (indicated by the small arrow) must contain the digit '2'. We can rule out digits 1, 7, 8, & 9, because those are already used elsewhere in row 2. Likewise, we can eliminate digits 3, 5, & 6; because they are used elsewhere in column 3. Finally, we can eliminate the digits 3, 4, & 8, because they are used in the same 3x3 box. Therefore, by elimination, we can deduce that the square must contain the digit '2'.

Although in the general case a Su Doku puzzle may have more than 1 solution, for the purposes of this problem the test cases are such that every puzzle has at most 1 solution. For example, the puzzle shown above has only 1 solution. However, there are test cases that have no solution.

Write a program named **`sudoku.cpp`** that reads input from a file named **`sudoku.in`** and writes output to a file named **`sudoku.out`**. All output to the screen will be ignored.

## Input File (`sudoku.in`)

The first line in the input file contains a single positive integer that represents a test case number.

The next 9 lines contain a test case. Each of these lines will contain 9 digits – all separated by a single space. The digit '0' is used to indicate an empty box in a Su Doku puzzle. There are no leading or trailing spaces on any input line. Thus, the only spaces in the file are those that separate the 9 digits on each line of the test cases.

[1] This problem appeared on the March 31, 2006 CCSC Student Programming Contest.

The next line after a test case will contain another test case number. If the test case number is greater than 0, the next 9 lines will contain the digits of the next Su Doku puzzle. A test case number equal to 0 indicates the end of the input file.

For example, the puzzle displayed at the top of the previous page is represented by the first test case shown below. There are no blank lines in the input file.

## Output File (`sudoku.out`)

For each test case write a line to the file that states: "`Test Case #n:`", where *n* is the case number. If the puzzle represented by the given test case has a solution, the next 9 lines of output should contain that solution. Each digit should be separated from the others by a single space. There must not be any leading or trailing spaces.

If the puzzle represented by the given test case has no solution, then the next line should contain the text:
"`No solution`".

One blank line must be written to the output file between every pair of test cases; however, there must not be a blank line before the first test case nor after the last.

**Sample Input File**

```
1
0 4 0 0 0 0 0 0 0
8 0 0 9 0 1 0 0 7
3 0 0 0 5 4 0 2 1
0 0 6 0 0 0 0 0 0
0 2 3 0 0 0 5 0 0
0 0 0 7 0 9 0 0 6
0 0 0 0 0 3 9 0 8
0 8 5 0 9 0 6 0 0
0 0 0 4 0 0 0 0 5
2
0 0 0 7 0 1 0 0 9
0 3 2 4 0 0 0 0 0
0 7 1 3 0 9 0 4 5
3 2 0 6 5 0 0 0 0
0 0 0 0 7 0 0 5 6
6 0 7 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 8 7 3
0 0 5 0 4 0 0 9 1
3
7 0 6 0 0 1 0 8 0
8 0 0 7 0 0 0 4 9
0 1 3 0 5 4 2 0 7
0 0 4 3 0 7 0 5 0
6 2 0 9 0 5 0 0 1
0 3 0 6 0 2 8 0 0
2 0 5 4 3 0 9 1 0
3 0 0 0 0 9 0 0 8
4 8 0 1 2 0 7 0 0
0
```

**Sample Output File**

```
Test Case #1:
5 4 1 8 7 2 3 6 9
8 6 2 9 3 1 4 5 7
3 9 7 6 5 4 8 2 1
9 1 6 3 4 5 7 8 2
7 2 3 1 8 6 5 9 4
4 5 8 7 2 9 1 3 6
2 7 4 5 6 3 9 1 8
1 8 5 2 9 7 6 4 3
6 3 9 4 1 8 2 7 5

Test Case #2:
No solution

Test Case #3:
7 4 6 2 9 1 5 8 3
8 5 2 7 6 3 1 4 9
9 1 3 8 5 4 2 6 7
1 9 4 3 8 7 6 5 2
6 2 8 9 4 5 3 7 1
5 3 7 6 1 2 8 9 4
2 7 5 4 3 8 9 1 6
3 6 1 5 7 9 4 2 8
4 8 9 1 2 6 7 3 5
```