Su-Do-Kode

Dave has become addicted to Sudoku, the latest puzzle craze in all the newspapers and bookstands. In case you don’t know, a Sudoku is a simple number puzzle played on a 3\times3 grid of 3\times3 subgrids. Below is an example:

\begin{tabular}{|c|c|c|}
\hline
5 & 7 & 4 \hline
8 & 5 & 9 \hline
6 & 2 & 3 \hline
\end{tabular} \hspace{1cm} \begin{tabular}{|c|c|c|}
\hline
3 & 5 & 7 \hline
2 & 1 & 6 \hline
5 & 2 & 1 \hline
\end{tabular}

This is the Initial Puzzle Solution. The object of Sudoku is to place numbers 1 through 9 in the empty spaces such that no row, column, or 3\times3 subgrid has any number more than once. An interesting property of Sudoku puzzles is that there is always only one possible solution, and it can always be determined using logic, without the need for guessing. Although Dave is wild about Sudoku, he still comes up with incorrect solutions. Dave is tired of being made fun of by his more Sudoku-savvy friends, so he’s asked you to write a program to check his solutions for him.

The Problem:

Given a set of Dave’s Sudoku puzzle solutions, determine which ones are correct, and which are invalid. For a Sudoku solution to be correct, every row, column, and 3\times3 subgrid of the puzzle must have each digit (1 through 9) exactly once.

The Input:

Dave will give you multiple solutions to check. Input will begin with a single integer, \( n \), on a line by itself. Following this will be \( n \) sets of 9 lines, each containing 9 digits. Each digit will be in the range 1 through 9, inclusive. Each set of 9 lines of 9 digits represents one of Dave’s potential Sudoku puzzle solutions.

The Output:

For each Sudoku solution, print “Sudoku \( \#d:\)”, where \( d \) represents the number of the puzzle (beginning at 1). Follow this with two spaces and either “Dave’s the man!”, if the solution is correct, or “Try again, Dave!”, if it is invalid. Leave a blank line after the output for each solution.

Sample Input:
Sample Output:

Sudoku #1:  Dave’s the man!

Sudoku #2:  Try again, Dave!