# CS 241 Data Organization using C
# Lab 5: Sudoku Solver

February 18 2020 (Due on March 5 2020)

## 1 Sudoku

Sudoku is a logic-based, combinatorial number placement puzzle. The objective is to fill a 9x9 grid with digits so that each column, each row, and each of the nine 3x3 sub-grids that compose the grid (also called "boxes", "blocks", "regions", or "sub-squares") contains all of the digits from 1 to 9. The puzzle setter provides a partially completed grid, which for a well-posed puzzle has a unique solution.[1]

## 2 Requirements

Write a C program that reads a bunch of sudoku puzzles from standard input and solves them. Each puzzle is given as a line of 81 characters, with '.' characters representing unknown digits.

- If a solution exists that uses the given digits in the given order, find it and print it. Specifically, echo the input on one line and print the solution on a new line.
- If not, echo the input and print "No Solution" on a new line.
- If the input does not follow the specified format, print "Error" on a new line.
- After each test case, output a blank line to standard output. So, each line of input will produce 3 lines of output: echo of input, output line, and blank line.
- Whether or not the test was a valid sudoku puzzle, your program must be prepared to handle the next test. Make sure that you do not leave your program in an invalid state as the result of an error test case. This is a particularly common issue when dealing with lines that were not of the expected length. Make sure you reset all relevant bookkeeping before beginning the next line of input.

### 2.1 Input Format

- A record is a sequence of characters followed by the newline character: '\n'.
- The input will consist of some number of records (lines).
- Each valid record will consist of 81 characters, representing the 9 rows of a sudoku puzzle, each consisting of 9 characters.

[1] https://en.wikipedia.org/wiki/Sudoku

- The only valid characters in a record are the nine possible digits (1-9) and ‘.’.
- A valid record does not have the same number appear twice in the same row, column, or box.

## 3 Turning in your assignment

Attach your program file `sudoku.c` to the Sudoku assignment in UNM Learn.

## 4 Grading Rubric (total of 100 points)

[-5 points]: The program does not start with a comment stating the students first and last name and/or the source file is not named correctly.

[-10 points]: Program compiles with warnings on `moons.cs.unm.edu` machine using `/usr/bin/gcc` with the **-Wall -ansi -pedantic** options

[-10 points]: Program takes more than 5 minutes to complete all tests in `testSudoku.in`, regardless of correctness of output.

[20 points]: Code follows the CS-241 Coding Standard.

[60 points]: 2 points for each passed test record of `testSudoku.in` using:

```
./yourprogram < testSudoku.in > youroutputfile
diff youroutputfile testSudoku.out
```

[20 points]: 1 point for each passed test record of an unknown set.

[5 bonus points]: Correctly solve a file of 100 unknown test cases in less than 2 seconds (when run on `moons.cs.unm.edu`). (Hint: one of the unknowns may be `hardPuzzle.in`) No extra credit possible if you do not pass all the tests in `testSudoku.in` (Let’s not waste the graders’ time timing incorrect programs.)
You program will be timed using the `time` command as follows:

```
time ./yourprogram < inputfile > outputfile
```