CSE141 Introduction to Programming
Fall'16

Assignment #3
Due: Oct 28, 2016

Dr. Imran Rauf
TA: Yusra Haider

Instructions

- Upload a single `.java` file and write your name and ERP id on the first line as comment.
- Late submission: There will be 20% penalty for up to *one* day late submissions, 50% for *two* days late submissions. *No submission will be accepted after two days past the due date.*
- Plagiarism: Students are expected to perform their work individually unless otherwise specified by the instructor. Assignments may be discussed in general terms with other students and the students may receive assistance from the instructor, TA, or classmates. Assistance does not mean obtaining solutions and modifying them; this is considered plagiarism.

1. **Background**: Sudoku is a logic puzzle in which the player fills a 9×9 grid with numbers in such a way that every row, column, and 3×3 "box" contains each of the numbers 1 through 9 exactly once. An instance of the puzzle consists of a partially filled 9×9 grid, and it is the player's task to fill in the remaining cells using logical deduction. In the following figure, a Sudoku puzzle is shown on left and a solution (marked in red) is shows on the right.

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

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

Task: Your task is to write a program that verifies whether a given Sudoku puzzle is a valid Sudoku solution or not. Given a Sudoku puzzle as input, your program should give, as output, whether the given puzzle is a valid solution or not. So, for example, the left puzzle in the above figure is not valid Sudoku solution while the one on right would be a valid Sudoku solution.

Represent *blank* or *empty* cells with the number 0.

You may use the following code snippet for generating a valid Sudoku solution, and for testing purposes:

```
int[][] matrix = new int[9][9];
for (int i = 0; i < 9; i++) {
    for (int j = 0, n = (i * 3 + i / 3); j < 9; j++, n++) {
        matrix[i][j] = (n % 9) + 1;
    }
}
```