

## Ass 5

### Instructions

Finish `src/sudoku/Sudoku`, `src/sudoku/SudokuTest`

### DELIVERABLES:

1. Submit the full source.
2. Sample output
3. Provide a text document title README that tells me for each of the methods, if they: work, or do not work.

The classic **Sudoku** game involves a grid of 81 squares. The grid is divided into nine blocks, each containing nine squares. The numbers 1-9 must appear once in each row, column and one of the nine blocks has to contain all the numbers 1-9 within its squares. Each number can only appear once in a row, column or block.

We generalize to a version of dimension  $n$ , which yields puzzles on squares of size  $n^2$ . The rules of the game are that each row, column and one of the  $n$  blocks of size  $n \times n$  have all numbers  $1..n^2$  in the squares. We show examples for  $n=1$ ,  $n=2$ , and  $n=3$  (which is the classic Sudoku).

### Deliverables.

1. Submit the full `src` directory, so that I can compile and run.
2. Add a README text file, that documents for each of the following methods, if they work correctly or not.
  - (i) `public Sudoku(int dim)`
  - (ii) `public Sudoku(int dim, int[][] square)`
  - (iii) `public static Sudoku fromFile(int dim, String filename)`
  - (iv) `public static class ParseException extends Exception`
  - (v) `public String toString()`
  - (vi) `public Formula getProblem() :`
    - \\ page 32 of lec 13 from MIT notes
  - (vii) `public Sudoku interpretSolutionN()`
    - \\ page 33 of lec 13 from MIT notes
3. Your tests should include the generation of formulas for the following configurations:

1x1: \*  
1x1: 1

2x2: 1 \* \* \*  
      \* 2 \* \*  
      \* \* 3 \*  
      \* \* \* 4

3x3:

```
4 . . | . . . | 8 . 5
. 3 . | . . . | . . .
. . . | 7 . . | . . .
-----+-----+-----
. 2 . | . . . | . 6 .
. . . | . 8 . | 4 . .
. . . | . 1 . | . . .
-----+-----+-----
. . . | 6 . 3 | . 7 .
5 . . | 2 . . | . . .
1 . 4 | . . . | . . .
```

Please include screen shots of your program producing output for these examples