Overview

Sudoku is a very popular **number placement puzzle** where the person attempting to solve it must fill a n x n grid with numbers such that every row, column and subset have unique elements.

Files and external data

Input has to be provided using the following means:

1. Number of rows/columns
2. List of allowed symbols that can be used to fill the cells
3. A pre-set combination of cells and their values

Output:

1. Solved Sudoku matrix for the given combination

Data structures and their relations to each other

Data Structures or Classes used in this implementation can be listed as follows

1. Lists
2. Arrays
3. **isValid -** used to determine if the cell can hold a unique value and yet preserve the uniqueness of the entire matrix
4. **isSolved -** Returns true if solved, else returns false
5. **findEmptyCell -** Used to find the next empty cell to fill

Assumptions

No Additional Assumptions

Key Test and Corner Cases

1. When the user tries to enter a matrix with lesser number of rows than the size of the matrix specified in the beginning, it raises an exception
2. When the user tries to create a list of valid symbols, the programming logic will ensure the default character chosen to fill empty spots does not have a conflict with it

References

1. **https://www.geeksforgeeks.org/sudoku-backtracking-7/**
2. **https://en.wikipedia.org/wiki/Sudoku**
3. **https://hackernoon.com/sudoku-and-backtracking-6613d33229af**