Formulate both of them as a CSP problems, i.e. define variables, their domains and constraints in a formal or at least semi-formal way.

• Implement searching + constraint propagation algorithm for solving the chosen CSPs , e.g. backtracking algorithm enhanced with selected heuristics and combined with the forward checking.

• Test the implementations with problems of different sizes or different difficulty, i.e. for different sizes of the boards (n x m), number of letters to be used, and the number words to be used on the board, whenever it is relevant to the given problem.

• Present most interesting results: solutions and numerical results (e.g. execution times, also for problems that cannot be solved).

• A explanation doc should contain all the above points (implementation should be delivered in a form of the source code files).

Sudoku :

perform experiments on the following number of files:

• 1 file from the group of easy problems (the levels 1 and 2),

• 1 file from the group of medium problems (the levels 3 - 5),

• 1 file from the group of difficult problems (6 and 7),

• 1 file from the group of mingled problems (the level: 8),

• 1 file from the group of mingled problems (the level: 9)

Fill in Puzzle :

Instances of Fill-in puzzle are to be found in the folder (5 instances).

Each instance is described by two files: puzzle and wordsX.

The first file includes the description of the board, the second one provides words. The instance puzzle0 is for testing code and its solution is given in the file filled0.

No datasets required for SCRABBLE obviously 😊)