Overview: Test cases for the sudoku problem.Test cases are divided into mainly 4 parts- Input Validation, Boundary Cases, Control Flow and Data Flow. In all that section test cases are divided by methods.

Input Validation

- Sudoku constructor
  - Send a 0 size.
  - Send a null value.
  - Send negative number.
- setPossibleValues
  - Send a null string.
  - Send an empty string.
- setCellValue
  - Pass an integer negative in x and positive in y.
  - Pass an integer positive integer in x and negative in y.
  - Pass an integer negative in x and y both.
- toPrintString
  - Pass an emptyCellLetter which is already a Symbol of a grid.
  - Pass null value in emptyCellLetter.
- Solve
  - Sudoku grid with no empty cell.
  - sudoku grid with all empty cells.

Boundary Cases

- Sudoku constructor.
  - Create sudoku of size 1
  - Create sudoku of large size
- SetPossibleValues
  - Pass a large string as a value.
  - Pass string of length 1.
- setCellValue.
  - Pass letter which is not part of possible value.
- toPrintString
  - Create a String for grid size 1.
  - Create a String for a large grid size.
- Solve
  - Sudoku grid with single value remaining to fill.
  - Sudoku grid with a large number of values remaining to fill in cells.

Control Flow

Sudoku:

- Create sudoku of size 3 (9 rows and 9 columns).
- Create sudoku of size 10 (100 rows and 100 columns).

SetPossibleValues:

- Pass a string containing characters.
- Pass a string that contains whitespace.
- Pass a string containing integers.
- Pass a string containing a combination of characters and integers.
- String could be comma separated or space separated.
- The length of the passed string is less than the size of the mini grid.
- The length of the passed string is more than the size of the single grid.
- String value contains the same character with different case (Capital/Small).
- String value contains a single or multiple space.
- The string contains duplicated characters in it.
- The number of possible symbols is less than the required symbols.
- The number of possible symbols is greater than the required symbols.

setCellValue:

- Pass letter which is not part of possible symbols.
- Pass a letter which might be duplicate
    - Pass a value that is already present in the same row.
    - Pass a letter that is already present in the same column.
    - Pass a letter which is already present in that mini-grid of specified rows and column.
- For the value of x and y
    - Pass row number x which does not exist.
    - Pass column number y which does not exist.
- Set value for Cell which is already set before.
- Set values for all cells.
- Set value for only a single cell.

toPrintString:

- Create a printable string for all empty cells in the grid.
- Create a printable string for no empty cell in a grid.
- Create a printable string for the grid size 4 of sudoku.
- Pass a special character as a letter.

Solve:

- Multiple solutions possible for sudoku.
- Invoke this method with a grid contains a single cell value.
- Invoke this method with all cell valued is filled.
- Invoke this method in a case where no solution possible.

Data Flow

- Call SetPossibleValues method before creating sudoku using constructor.
- Call SetPossibleValues twice for the same object.
- Call SetPossibleValues twice for a new object.
- Call setCellValue before setting possible symbols of sudoku using SetPossibleValues.
- Call solve method without invoking sudoku.
- Call solve method without setting initial cell value using setCellValue method.
- Call solve method, then set some value of cell using setCellValue, then again call the solve method.
- Call toPrintString method before invoking SetPossibleValues method.