# Sudoku Solver

## Features

- Solves standard 9x9 Sudoku puzzles using the \*\*backtracking algorithm\*\*.  
- Handles Sudoku puzzles with \*\*multiple solutions\*\*.  
- Allows \*\*custom puzzle input\*\*.  
- Prints each solution and the \*\*total number of solutions\*\* found.  
- Validates user input for correctness.

## How It Works

The program uses a recursive backtracking approach:  
1. \*\*Row-Column Traversal:\*\* It traverses the Sudoku grid row by row, column by column.  
2. \*\*Constraint Validation:\*\* For each empty cell, it tries digits `1-9` and checks whether the placement is valid using:  
 - Row check  
 - Column check  
 - 3x3 subgrid check  
3. \*\*Backtracking:\*\* If no valid digit is found, it backtracks to the previous step and tries the next option.

## Input Format

- The Sudoku grid is entered as a \(9 imes 9\) grid of integers.  
- Use `0` to represent empty cells.

\*\*Example Input:\*\*

0 0 8 0 0 0 0 0 0  
4 9 0 1 5 7 0 0 2  
0 0 3 0 0 4 1 9 0  
1 8 5 0 6 0 0 2 0  
0 0 0 0 2 0 0 6 0  
9 6 0 4 0 5 3 0 0  
0 3 0 0 7 2 0 0 4  
0 4 9 0 3 0 0 5 7  
8 2 7 0 0 9 0 1 3

## Output

- Each solution is printed row by row.  
- The total number of solutions found is displayed.

\*\*Example Output:\*\*

--------Solution is : ---------  
5 1 8 6 9 3 2 4 7  
4 9 6 1 5 7 8 3 2  
7 2 3 8 2 4 1 9 5  
1 8 5 7 6 9 4 2 3  
3 7 4 5 2 1 9 6 8  
9 6 2 4 8 5 3 7 1  
2 3 1 9 7 2 5 8 4  
6 4 9 2 3 8 7 5 7  
8 2 7 3 4 9 6 1 3  
  
Total solutions found: 1

## How to Use

1. Clone this repository:  
 ```bash  
 git clone https://github.com/your-username/sudoku-solver.git  
 ```  
2. Compile the program:  
 ```bash  
 javac SudokuSolver.java  
 ```  
3. Run the program:  
 ```bash  
 java SudokuSolver  
 ```  
4. Enter your Sudoku grid when prompted. For example:  
 ```  
 Enter the Sudoku grid row by row (use 0 for empty cells):  
 0 0 8 0 0 0 0 0 0  
 4 9 0 1 5 7 0 0 2  
 ...  
 ```  
5. View the solutions and total count.

## Constraints

- The Sudoku grid must be exactly \(9 imes 9\).  
- All input values must be integers between `0` and `9`.  
- Puzzles with no solutions will output:  
 ```  
 No solution exists for the given Sudoku.  
 ```

## Future Improvements

- Add a GUI interface for easier puzzle input.  
- Optimize the backtracking algorithm for performance.  
- Expand support for larger Sudoku grids (e.g., \(16 imes 16\)).

## License

This project is licensed under the MIT License. See the `LICENSE` file for details.

## Contributing

Contributions are welcome! Please fork the repository and submit a pull request for any enhancements or bug fixes.