# SODUKO SOLVER AGENT

## AI PROJECT 4

***Introduction:***

***The goal of the project is to develop a powerful AI Soduko Solving agent, and a Soduko generator which generates puzzles of various difficulties. Which showcases the strength of AI in solving logical puzzles.***

***Modes:***

### Mode 1: AI Generated Puzzle

- **Purpose:** This mode showcases the AI's problem-solving capabilities. The user generates a random Sudoku puzzle, and the AI solves it step by step.
- **Algorithms:** Emphasize that the AI utilizes a combination of backtracking and arc consistency for efficient solving.

### Mode 2: User Input, AI Solves

- **Purpose:** Allows the user to create a custom Sudoku puzzle and then challenges the AI to solve it.
- **GUI Features:**
  - Input mechanism for the user to fill cells on the Sudoku board with numbers.
  - "Solve" button to trigger the AI's solving process.
  - Visual representation of the AI's solving steps (similar to Mode 1).
- **Algorithms:** The AI employs the same backtracking and arc consistency algorithms as in Mode 1.

### Mode 3: Interactive Solving with AI Assistance

- **Purpose:** A collaborative mode where the user attempts to solve the puzzle with real-time guidance from the AI.
- **GUI Features:**
  - Input mechanism for the user to enter numbers into cells.
  - Immediate visual feedback (e.g., highlighting cells in red) if the user enters an invalid number that violates Sudoku rules.
- **Algorithms:** The AI continuously runs in the background, using the `is_valid_move()` function to check the validity of each user input.

***Implementation:***

Our Sudoku solver utilizes Constraint Satisfaction Problems (CSP) to crack the puzzles. It translates the Sudoku grid into variables (individual cells) and assigns each a domain of possible values (1-9). Leveraging arc consistency, the solver enforces the core Sudoku rules: no number can appear more than once in its row, column, or sub-grid.

**Data Structures:**

- **Grid Representation:** The Sudoku puzzle itself is stored as a two-dimensional array, mirroring the familiar 9x9 grid.
- **Variable Domains:** Each cell in the grid is considered a variable. Possible values for a variable (the numbers 1 through 9) are kept track of using lists.
- Arc consistency is implemented using a queue-based algorithm to revise inconsistent values in domains.

**Algorithms used:**

- **Backtracking Detective:** This algorithm serves a dual purpose. It checks if a given Sudoku puzzle has a valid solution (think of it as an inspector). Additionally, it can be used to generate new, solvable puzzles from scratch (like a puzzle architect!).
- **Arc Consistency Enforcer:** This technique ensures that the domains of connected cells (those in the same row, column, or sub-grid) are consistent with the Sudoku rules. It's applied repeatedly until no further adjustments are needed, guaranteeing a more efficient solving process.
- **MRV is utilized to prioritize the selection of the next node, by selecting the one with the least count of legal moves**

## Assumptions:

The solver assumes that the user's input in Mode 2 and Mode 3 follows Sudoku rules. Input is kept valid using pygame events

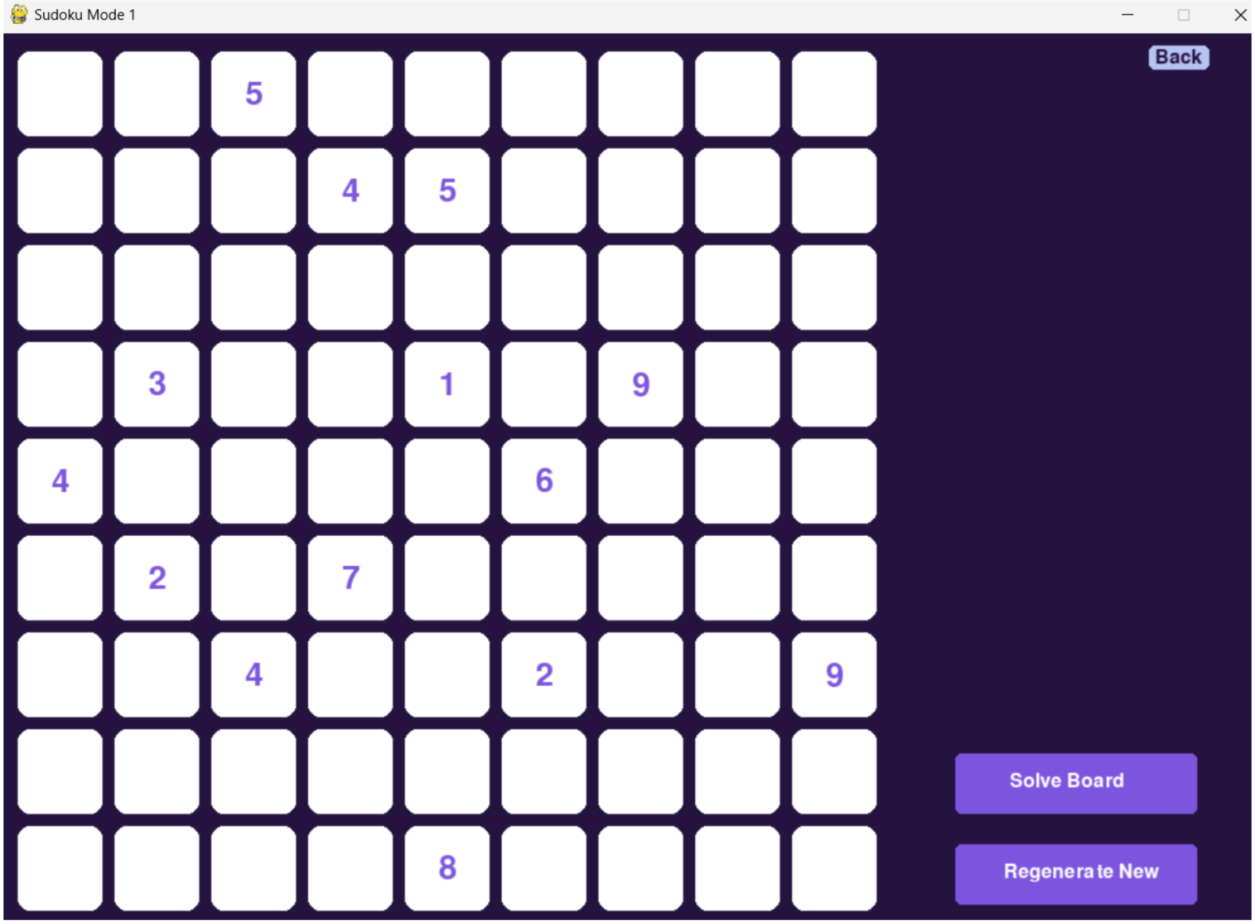
The solver uses a depth-first search with backtracking to explore possible solutions.

The solver updates the Sudoku grid based on the reduced domains until the entire board is filled.

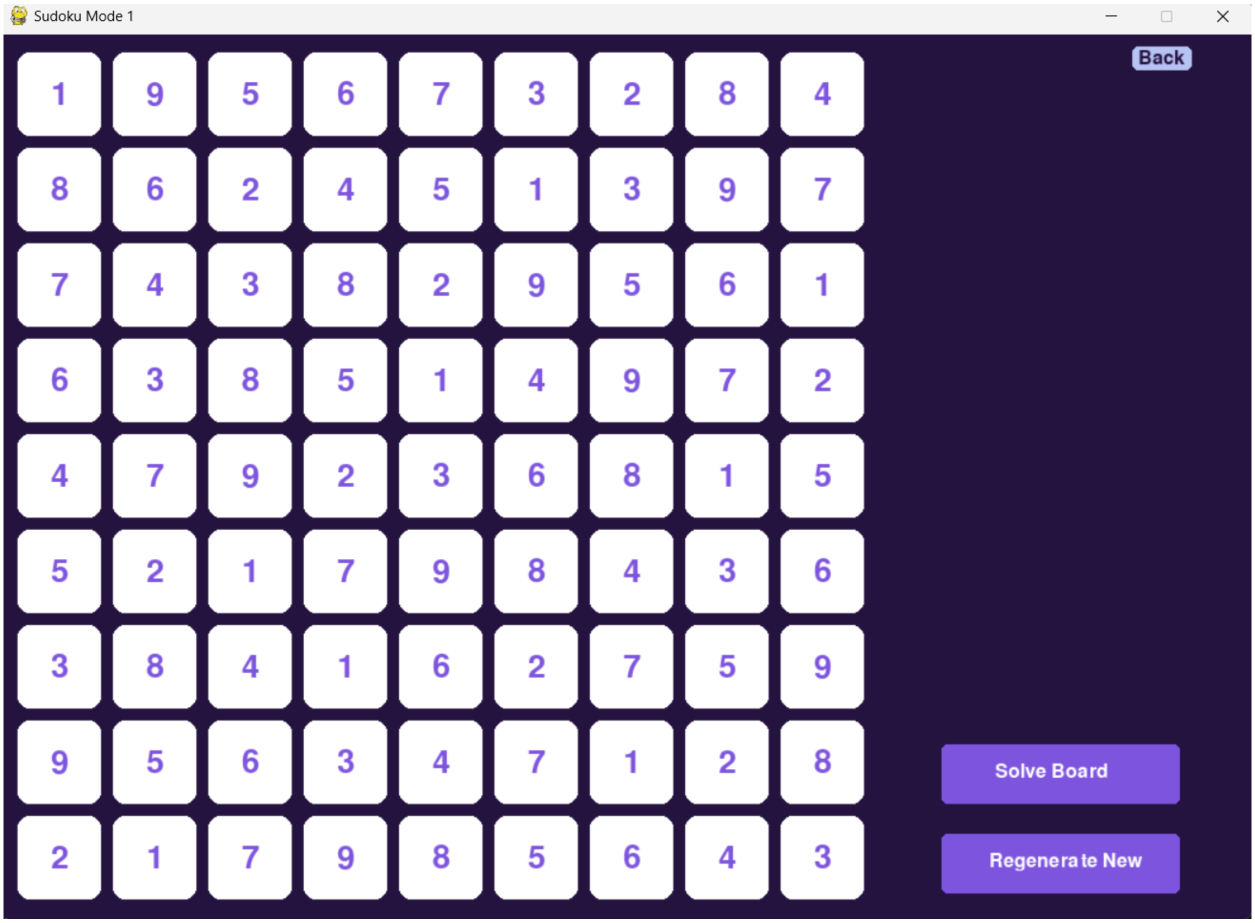
***SUPPORT RUNS:***

***Mode 1:***

***Easy:***

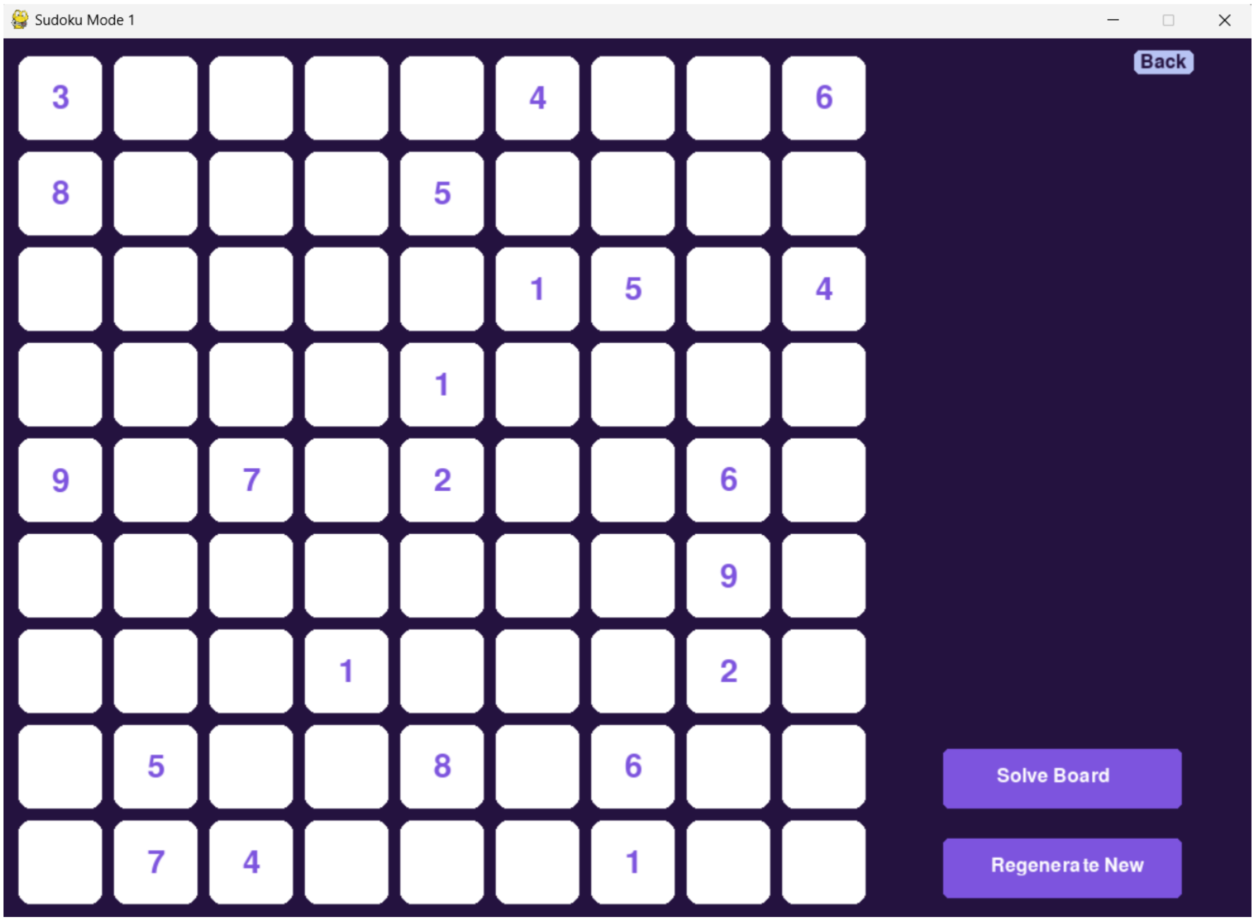


| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 5 | | | | | | |
| | | | 4 | 5 | | | | |
| | | | | | | | | |
| | 3 | | | 1 | | 9 | | |
| 4 | | | | | 6 | | | |
| | 2 | | 7 | | | | | |
| | | 4 | | | 2 | | | 9 |
| | | | | | | | | |
| | | | | 8 | | | | |

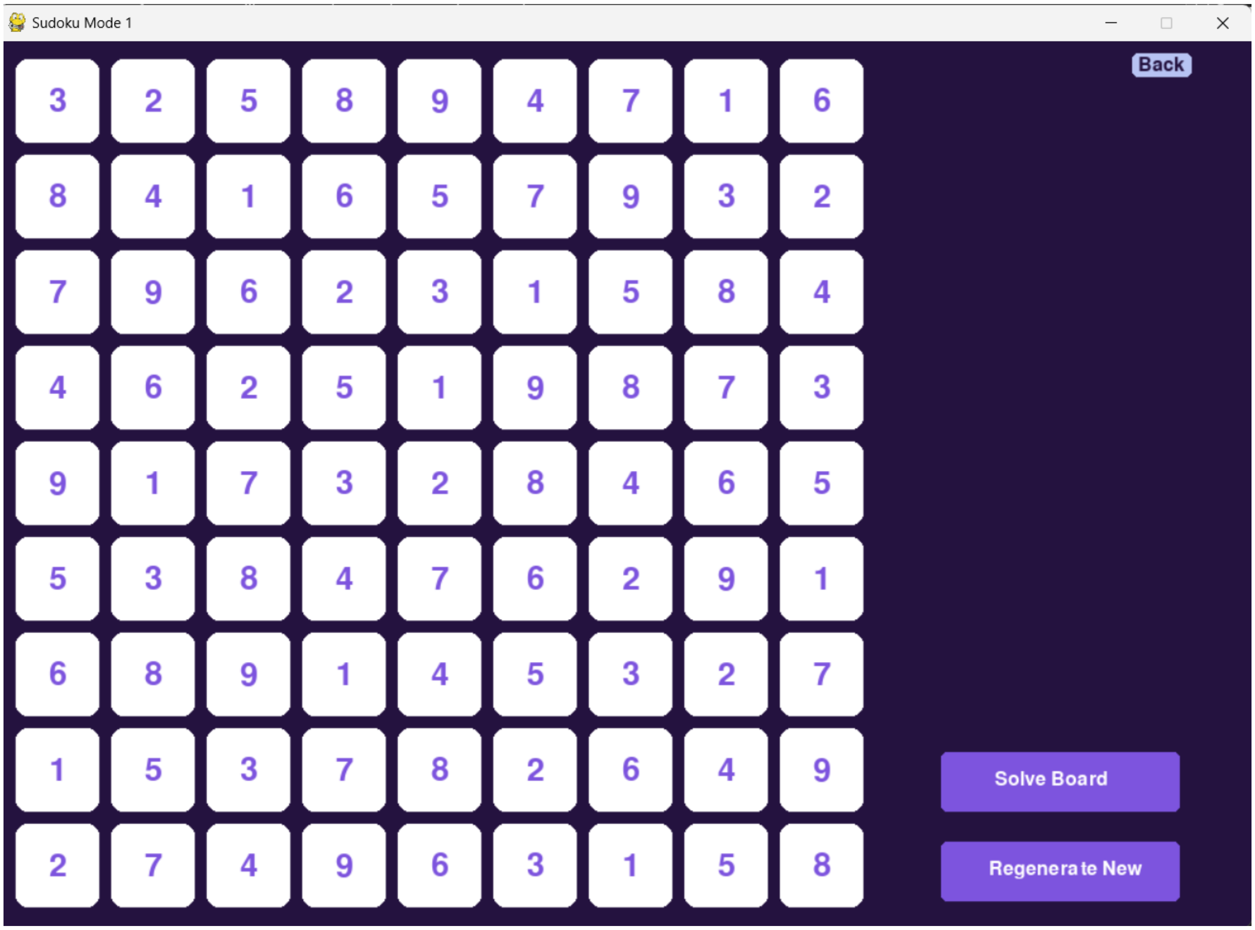


| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 9 | 5 | 6 | 7 | 3 | 2 | 8 | 4 |
| 8 | 6 | 2 | 4 | 5 | 1 | 3 | 9 | 7 |
| 7 | 4 | 3 | 8 | 2 | 9 | 5 | 6 | 1 |
| 6 | 3 | 8 | 5 | 1 | 4 | 9 | 7 | 2 |
| 4 | 7 | 9 | 2 | 3 | 6 | 8 | 1 | 5 |
| 5 | 2 | 1 | 7 | 9 | 8 | 4 | 3 | 6 |
| 3 | 8 | 4 | 1 | 6 | 2 | 7 | 5 | 9 |
| 9 | 5 | 6 | 3 | 4 | 7 | 1 | 2 | 8 |
| 2 | 1 | 7 | 9 | 8 | 5 | 6 | 4 | 3 |

***Medium:***

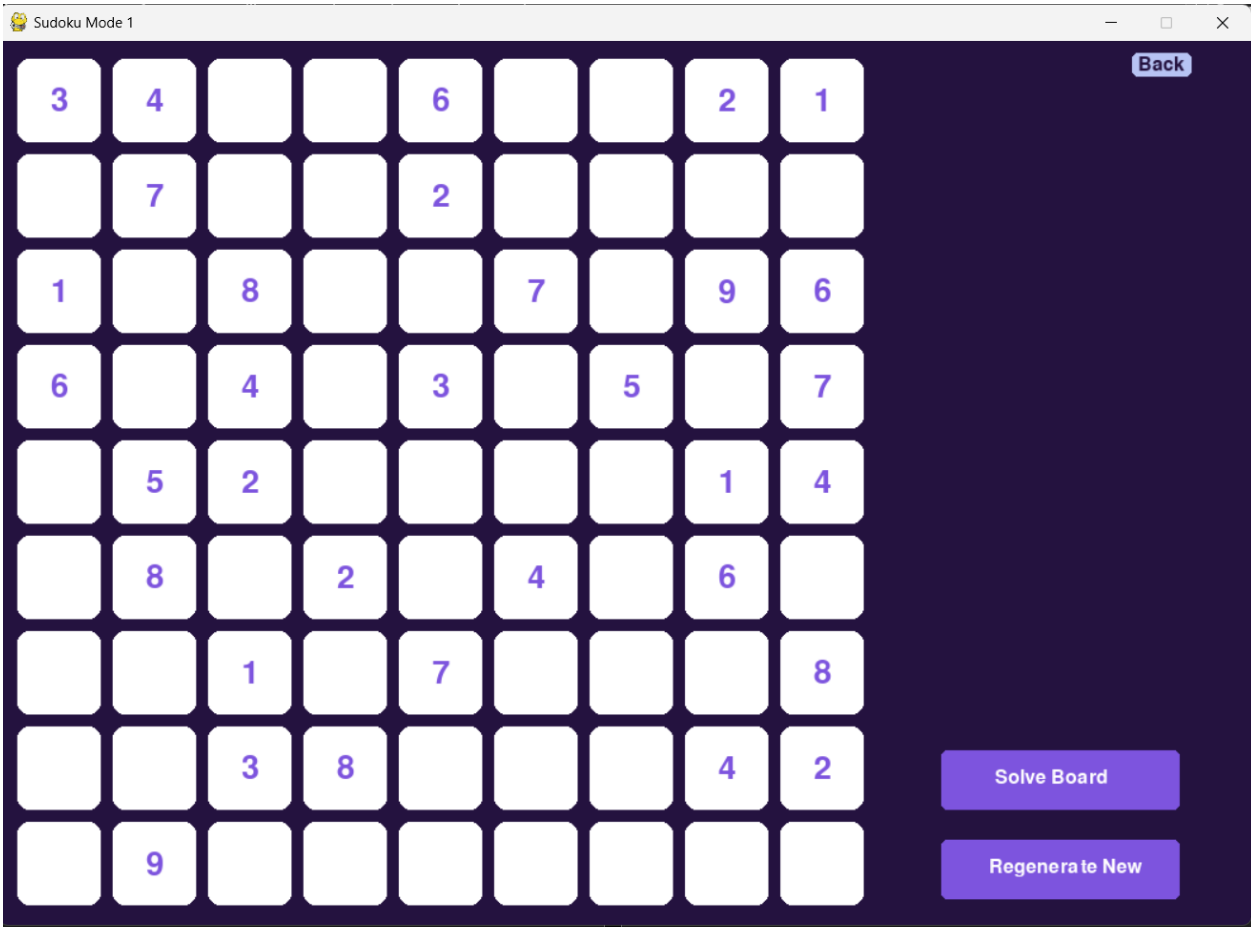


| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | | | | | 4 | | | 6 |
| 8 | | | | 5 | | | | |
| | | | | | 1 | 5 | | 4 |
| | | | | 1 | | | | |
| 9 | | 7 | | 2 | | | 6 | |
| | | | | | | | 9 | |
| | | | 1 | | | | 2 | |
| | 5 | | | 8 | | 6 | | |
| | 7 | 4 | | | | 1 | | |

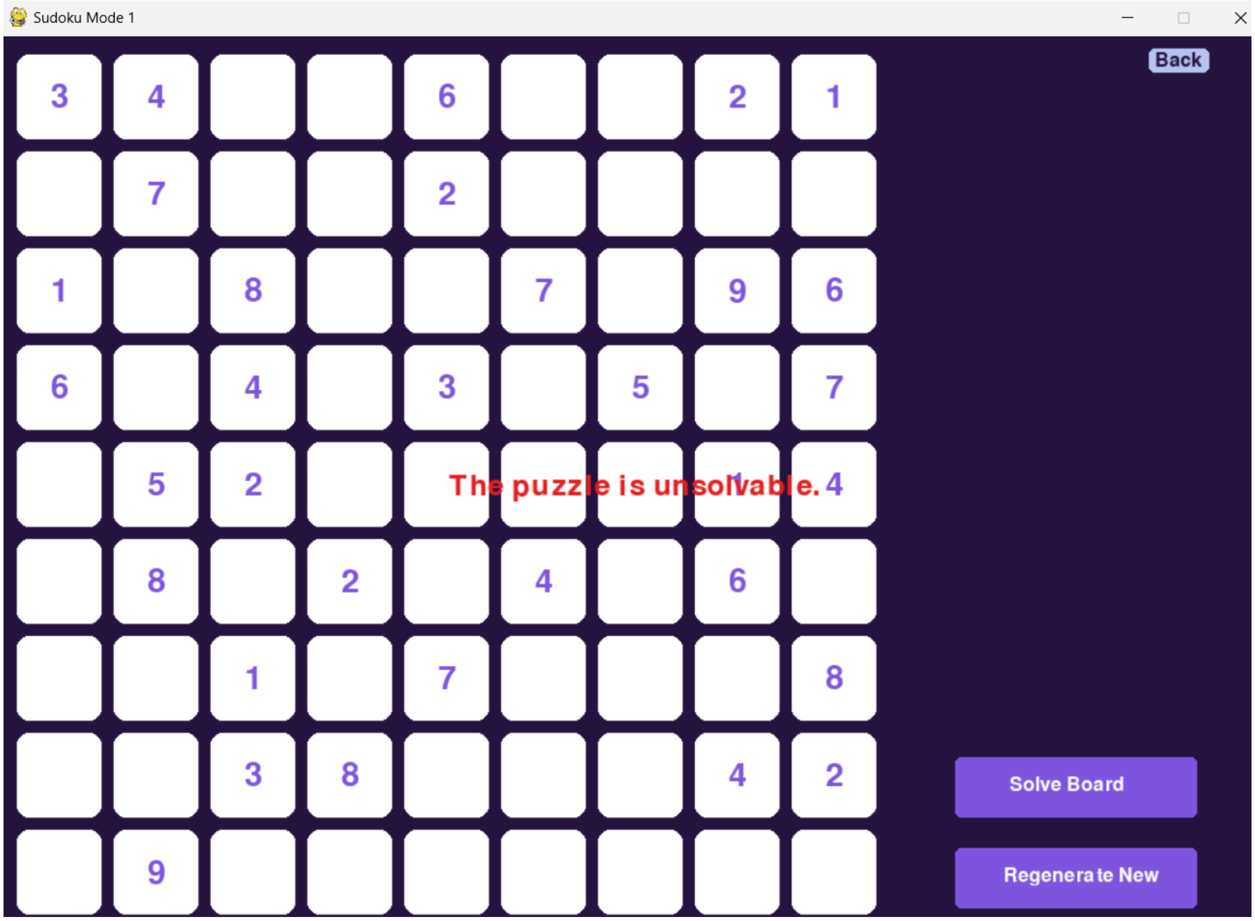


| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | 2 | 5 | 8 | 9 | 4 | 7 | 1 | 6 |
| 8 | 4 | 1 | 6 | 5 | 7 | 9 | 3 | 2 |
| 7 | 9 | 6 | 2 | 3 | 1 | 5 | 8 | 4 |
| 4 | 6 | 2 | 5 | 1 | 9 | 8 | 7 | 3 |
| 9 | 1 | 7 | 3 | 2 | 8 | 4 | 6 | 5 |
| 5 | 3 | 8 | 4 | 7 | 6 | 2 | 9 | 1 |
| 6 | 8 | 9 | 1 | 4 | 5 | 3 | 2 | 7 |
| 1 | 5 | 3 | 7 | 8 | 2 | 6 | 4 | 9 |
| 2 | 7 | 4 | 9 | 6 | 3 | 1 | 5 | 8 |

***Hard:***



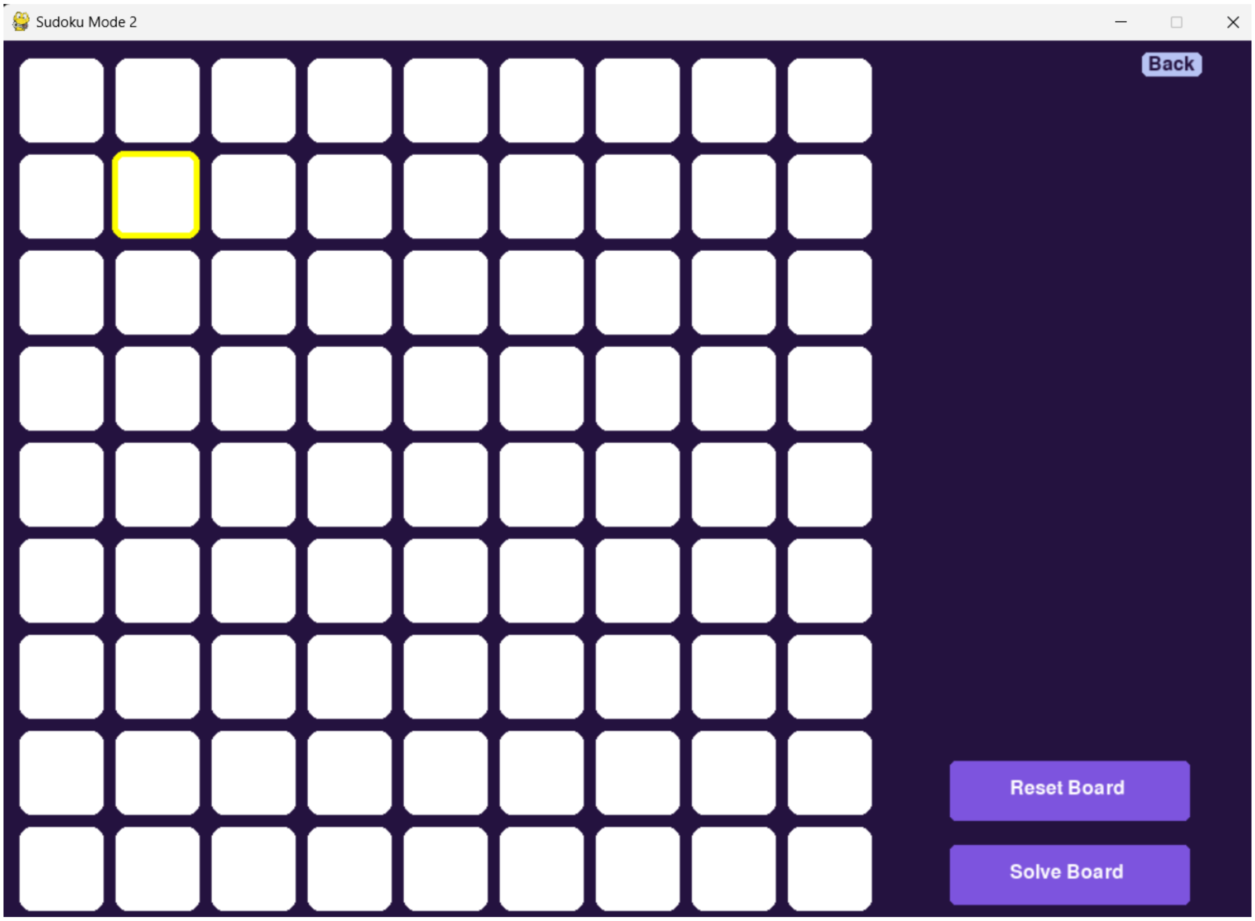
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | 4 | | | 6 | | | 2 | 1 |
| | 7 | | | 2 | | | | |
| 1 | | 8 | | | 7 | | 9 | 6 |
| 6 | | 4 | | 3 | | 5 | | 7 |
| | 5 | 2 | | | | | 1 | 4 |
| | 8 | | 2 | | 4 | | 6 | |
| | | 1 | | 7 | | | | 8 |
| | | 3 | 8 | | | | 4 | 2 |
| | 9 | | | | | | | |

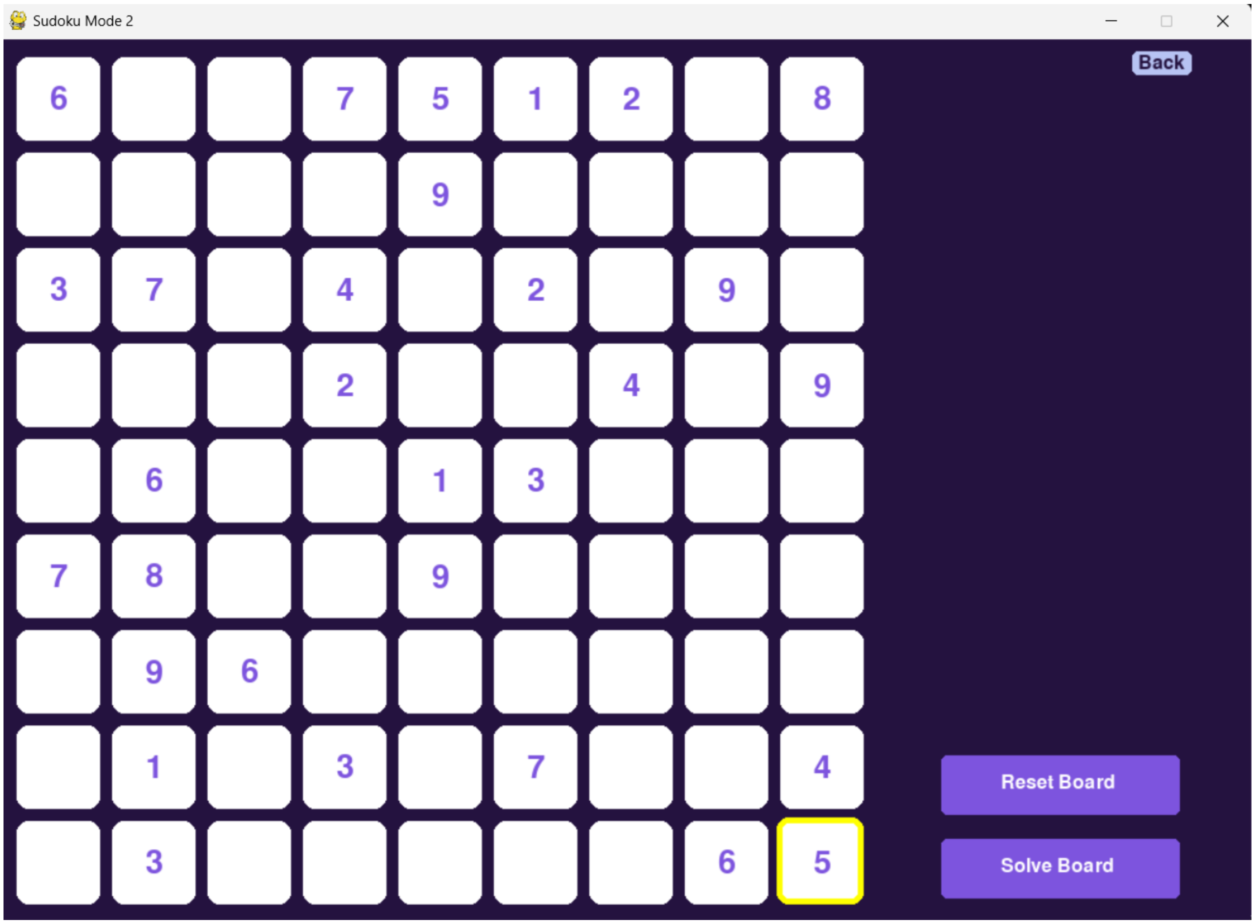


|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
| 3 | 4 |  |  | 6 |  |  | 2 | 1 |
|  | 7 |  |  | 2 |  |  |  |  |
| 1 |  | 8 |  |  | 7 |  | 9 | 6 |
| 6 |  | 4 |  | 3 |  | 5 |  | 7 |
|  | 5 | 2 |  |  |  |  | 1 | 4 |
|  | 8 |  | 2 |  | 4 |  | 6 |  |
|  |  | 1 |  | 7 |  |  |  | 8 |
|  |  | 3 | 8 |  |  |  | 4 | 2 |
|  | 9 |  |  |  |  |  |  |  |

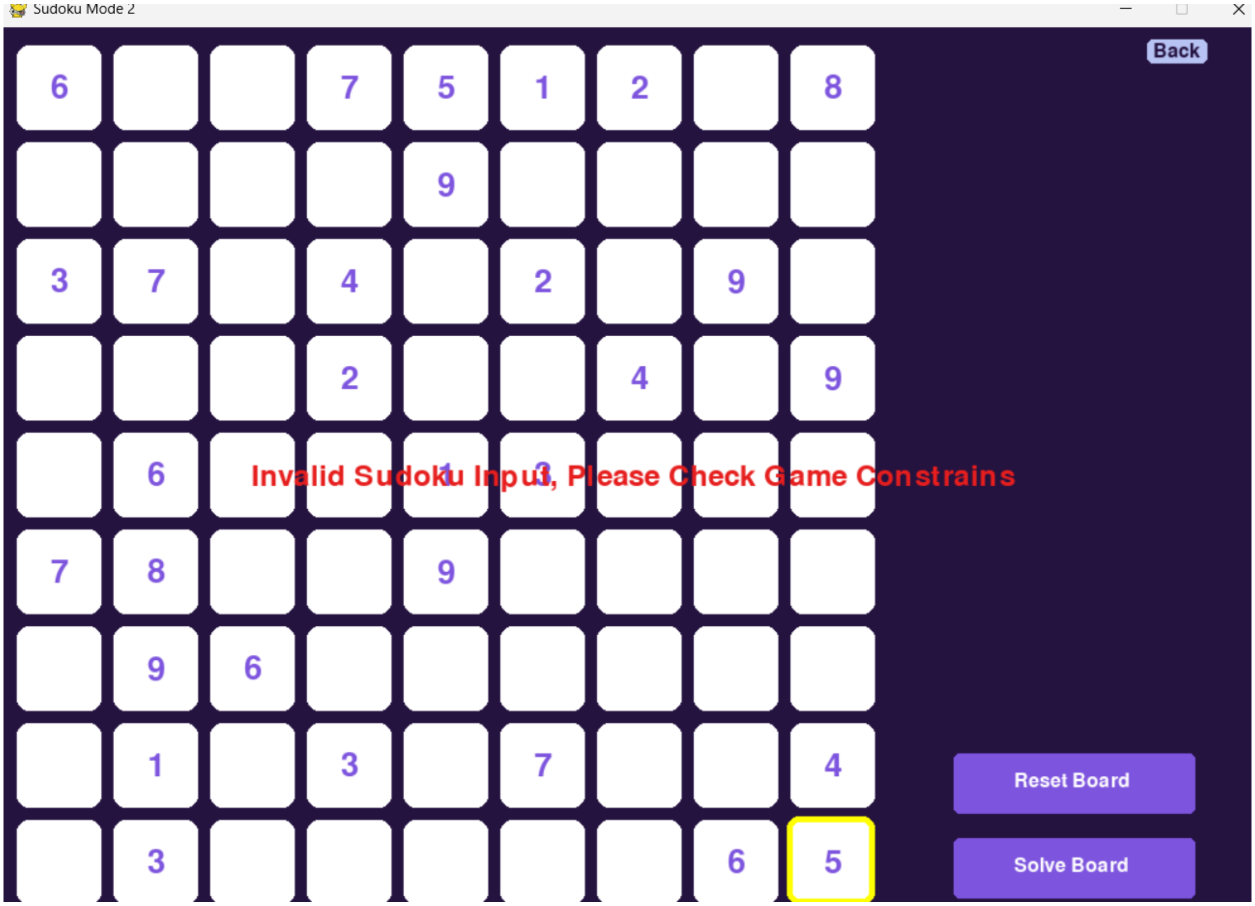
The puzzle is unsolvable.

***Mode 2:***


Sudoku Mode 2
Back
Reset Board
Solve Board



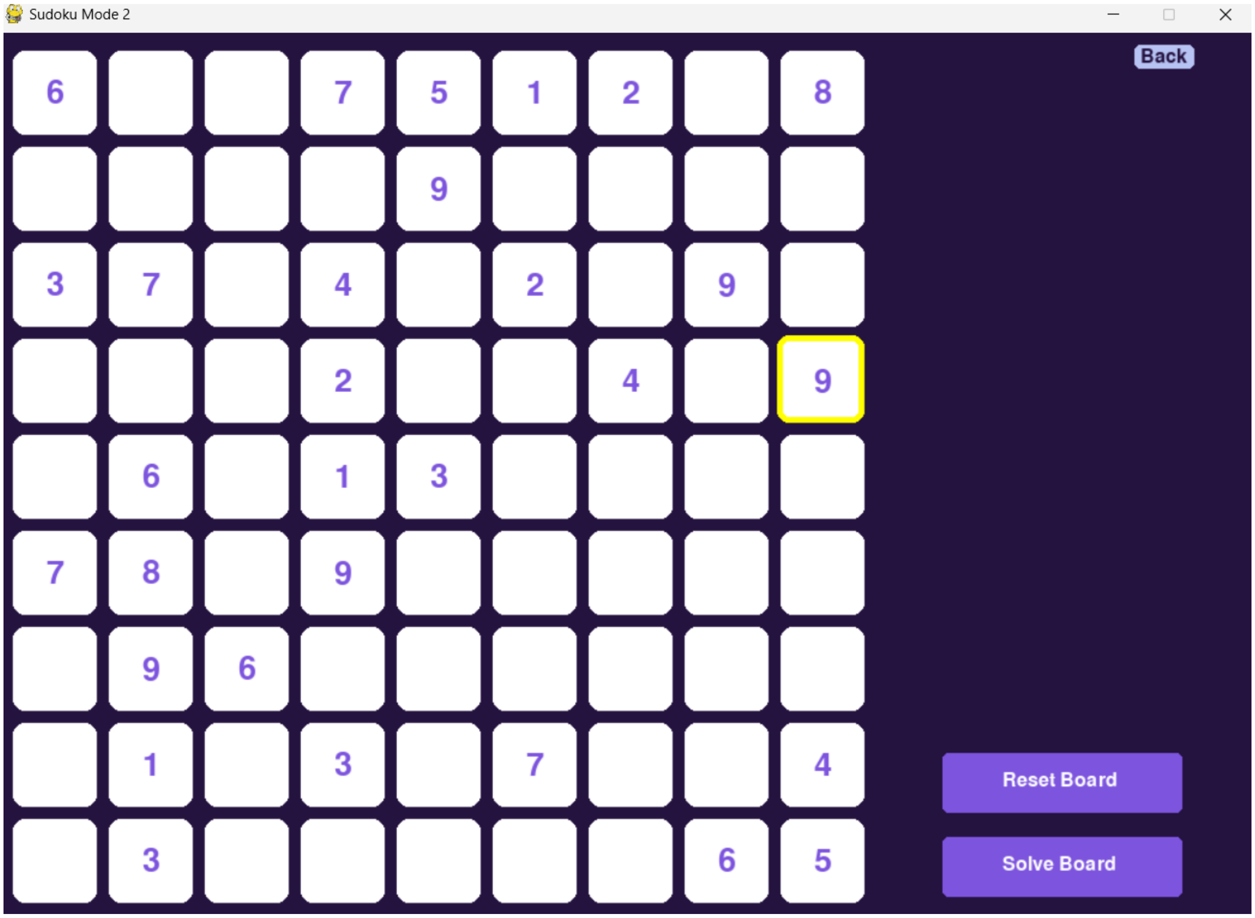
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 6 | | | 7 | 5 | 1 | 2 | | 8 |
| | | | | 9 | | | | |
| 3 | 7 | | 4 | | 2 | | 9 | |
| | | | 2 | | | 4 | | 9 |
| | 6 | | | 1 | 3 | | | |
| 7 | 8 | | | 9 | | | | |
| | 9 | 6 | | | | | | |
| | 1 | | 3 | | 7 | | | 4 |
| | 3 | | | | | | 6 | 5 |



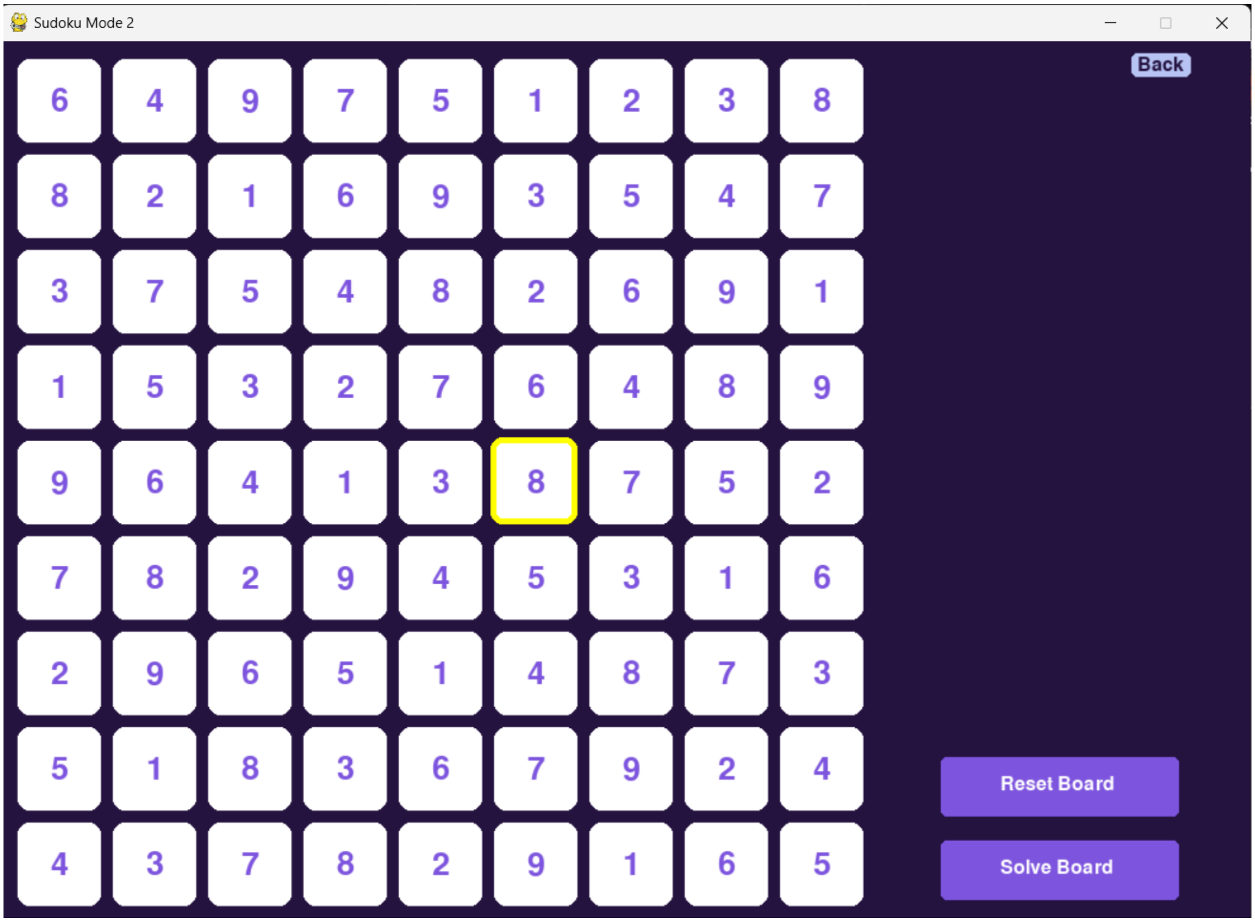
|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 6 |   |   | 7 | 5 | 1 | 2 |   | 8 |
|   |   |   |   | 9 |   |   |   |   |
| 3 | 7 |   | 4 |   | 2 |   | 9 |   |
|   |   |   | 2 |   |   | 4 |   | 9 |
|   | 6 |   |   | 1 | 3 |   |   |   |
| 7 | 8 |   |   | 9 |   |   |   |   |
|   | 9 | 6 |   |   |   |   |   |   |
|   | 1 |   | 3 |   | 7 |   |   | 4 |
|   | 3 |   |   |   |   |   | 6 | 5 |

Invalid Sudoku Input, Please Check Game Constrains

***Solvable Puzzle:***



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 6 | | | 7 | 5 | 1 | 2 | | 8 |
| | | | | 9 | | | | |
| 3 | 7 | | 4 | | 2 | | 9 | |
| | | | 2 | | | 4 | | 9 |
| | 6 | | 1 | 3 | | | | |
| 7 | 8 | | 9 | | | | | |
| | 9 | 6 | | | | | | |
| | 1 | | 3 | | 7 | | | 4 |
| | 3 | | | | | | 6 | 5 |



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 6 | 4 | 9 | 7 | 5 | 1 | 2 | 3 | 8 |
| 8 | 2 | 1 | 6 | 9 | 3 | 5 | 4 | 7 |
| 3 | 7 | 5 | 4 | 8 | 2 | 6 | 9 | 1 |
| 1 | 5 | 3 | 2 | 7 | 6 | 4 | 8 | 9 |
| 9 | 6 | 4 | 1 | 3 | 8 | 7 | 5 | 2 |
| 7 | 8 | 2 | 9 | 4 | 5 | 3 | 1 | 6 |
| 2 | 9 | 6 | 5 | 1 | 4 | 8 | 7 | 3 |
| 5 | 1 | 8 | 3 | 6 | 7 | 9 | 2 | 4 |
| 4 | 3 | 7 | 8 | 2 | 9 | 1 | 6 | 5 |