# Lab Assignment 2

**Subject:** Artificial Intelligence **Guided by:** Dr. Anuradha Yenkikar

**Student name:** Neha Sawant

**Experiment Name:** Implement Constraint Satisfaction Problem for Sudoku

## Objective:

- To solve the Sudoku puzzle using Constraint Satisfaction Problem (CSP) with backtracking.

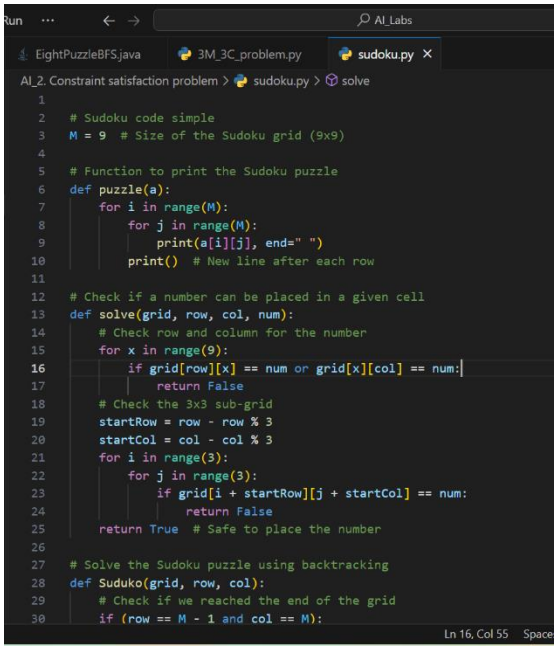
## Problem Statement:

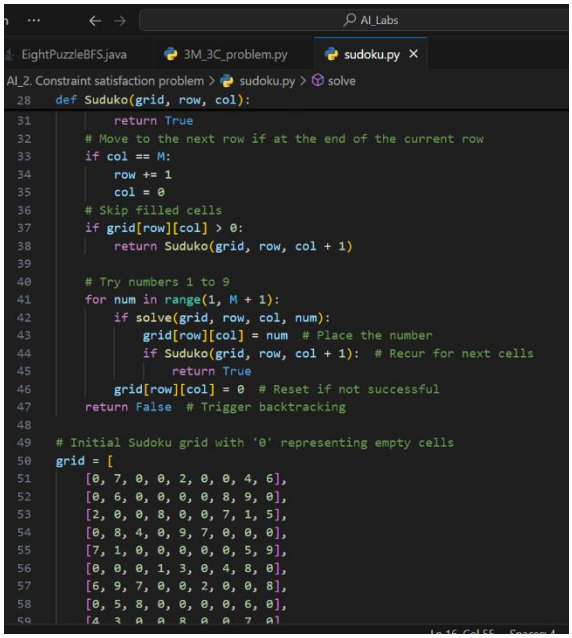
Solve a 9x9 Sudoku puzzle where some cells are filled with numbers and others are empty (denoted by 0). The objective is to fill the grid such that each row, column, and 3x3 sub-grid contains all digits from 1 to 9 without repetition.

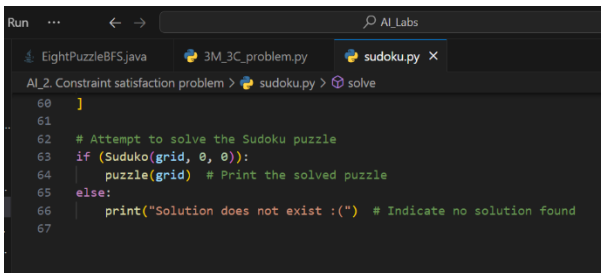
## Algorithm:

1. Backtracking:
   - Start by placing digits in empty cells.
   - Check if placing the digit satisfies Sudoku constraints (row, column, and 3x3 sub-grid).
   - If valid, place the number and continue solving the remaining puzzle recursively.
   - If no solution is found, backtrack and try different digits.

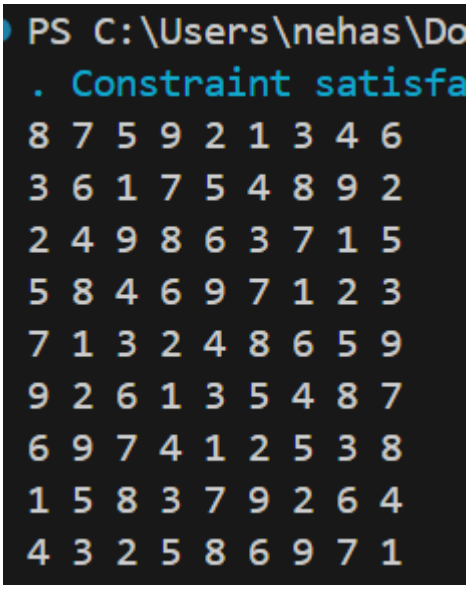
## Code Implementation (Sudoku Solver using CSP):




## Output Example:




## Conclusion:

From this assignment, I learned how to implement a Constraint Satisfaction Problem (CSP) to solve real-world puzzles like Sudoku. I gained a deeper understanding of how backtracking is used to explore possible solutions, ensuring that constraints (like unique digits in rows, columns, and sub-grids) are satisfied at each step. Additionally, I learned the importance of pruning in the solution space, which helps in optimizing the algorithm by reducing unnecessary computations.