Sudoku Solver Game

Title Page

Sudoku Solver Game

Introduction

Sudoku is a popular number puzzle game that requires logic and problem-solving skills. The Sudoku Solver is a Python-based program designed to solve Sudoku puzzles efficiently. It allows users to either enter their own Sudoku puzzle or use a predefined puzzle for practice. The program then uses an intelligent algorithm to solve the puzzle step by step, providing a quick and accurate solution.

Methodology

1. The program starts with a welcome message and prompts the user to choose between entering their own Sudoku puzzle or using a default practice puzzle.
2. If the user opts to enter their own puzzle, they input a 9x9 Sudoku grid where empty cells are represented by 0.
3. If the user chooses not to enter a puzzle, the program provides a predefined Sudoku puzzle for practice.
4. The program employs the **Backtracking Algorithm** to systematically fill in the missing numbers while ensuring the Sudoku rules are met.
5. The solved Sudoku grid is displayed to the user.
6. The program concludes with a thank-you message, ensuring a smooth and interactive user experience.

Code Typed:-

```
def is_valid(board, row, col, num):
    """Check if 'num' can be placed at board[row][col]."""

    # Check row and column
    for i in range(9):
        if board[row][i] == num or board[i][col] == num:
            return False

    # Check 3x3 grid
    box_x, box_y = (row // 3) * 3, (col // 3) * 3
    for i in range(3):
        for j in range(3):
            if board[box_x + i][box_y + j] == num:
                return False

    return True

def solve_sudoku(board):
    """Solve Sudoku using backtracking."""

    for row in range(9):
        for col in range(9):
            if board[row][col] == 0:  # Find empty cell
                for num in range(1, 10):  # Try numbers 1-9
                    if is_valid(board, row, col, num):
                        board[row][col] = num
                        if solve_sudoku(board):
                            return True
                        board[row][col] = 0  # Backtrack
                return False  # No valid number found

    return True

def print_board(board):
    """Print the Sudoku board."""
    for row in board:
        print(" ".join(map(str, row)))

# Welcome message
print("\nWelcome to Sudoku Solver!\n")
```

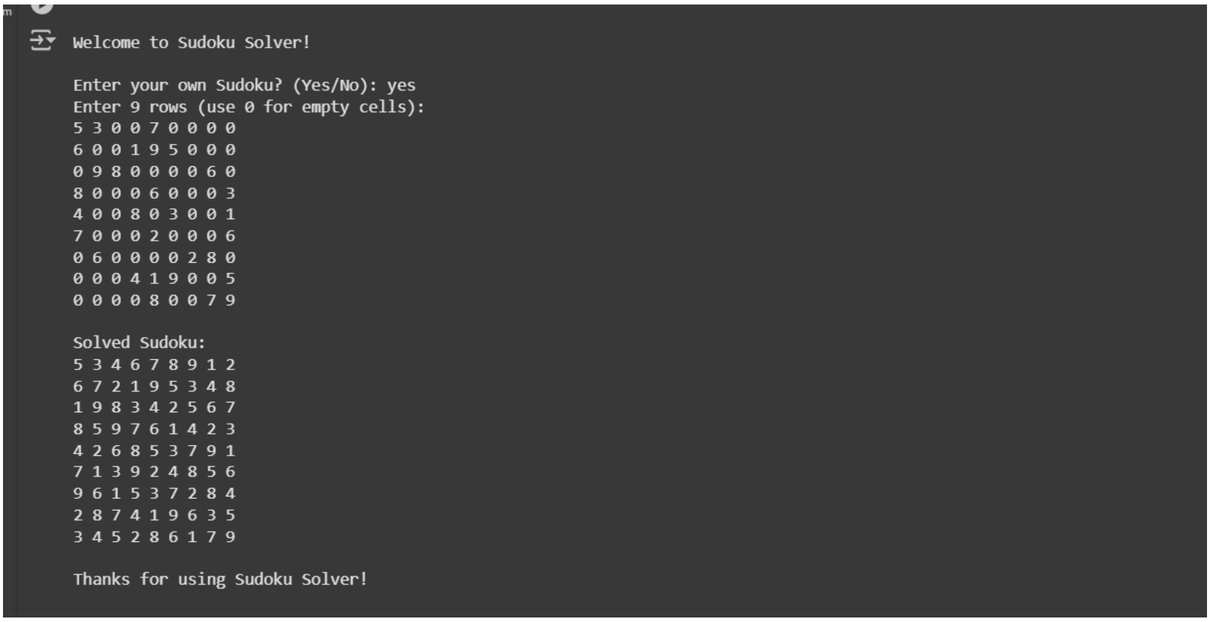
```
# User input or default Sudoku
choice = input("Enter your own Sudoku? (Yes/No): ").strip().lower()

default_sudoku = [
    [5, 3, 0, 0, 7, 0, 0, 0, 0],
    [6, 0, 0, 1, 9, 5, 0, 0, 0],
    [0, 9, 8, 0, 0, 0, 0, 6, 0],
    [8, 0, 0, 0, 6, 0, 0, 0, 3],
    [4, 0, 0, 8, 0, 3, 0, 0, 1],
    [7, 0, 0, 0, 2, 0, 0, 0, 6],
    [0, 6, 0, 0, 0, 0, 2, 8, 0],
    [0, 0, 0, 4, 1, 9, 0, 0, 5],
    [0, 0, 0, 0, 8, 0, 0, 7, 9]
]

if choice == 'yes':
    print("Enter 9 rows (use 0 for empty cells):")
    sudoku_board = [list(map(int, input().split())) for _ in range(9)]
else:
    print("\nSample Sudoku for practice:")
    print_board(default_sudoku)
    sudoku_board = [row[:] for row in default_sudoku]  # Copy for solving

# Solve and print result
if solve_sudoku(sudoku_board):
    print("\nSolved Sudoku:")
    print_board(sudoku_board)
    print("\nThanks for using Sudoku Solver!\n")
else:
    print("No solution exists.")
```

Screenshots Output Photo Pasted:



```
Welcome to Sudoku Solver!

Enter your own Sudoku? (Yes/No): yes
Enter 9 rows (use 0 for empty cells):
5 3 0 0 7 0 0 0 0
6 0 0 1 9 5 0 0 0
0 9 8 0 0 0 0 6 0
8 0 0 0 6 0 0 0 3
4 0 0 8 0 3 0 0 1
7 0 0 0 2 0 0 0 6
0 6 0 0 0 0 2 8 0
0 0 0 4 1 9 0 0 5
0 0 0 0 8 0 0 7 9

Solved Sudoku:
5 3 4 6 7 8 9 1 2
6 7 2 1 9 5 3 4 8
1 9 8 3 4 2 5 6 7
8 5 9 7 6 1 4 2 3
4 2 6 8 5 3 7 9 1
7 1 3 9 2 4 8 5 6
9 6 1 5 3 7 2 8 4
2 8 7 4 1 9 6 3 5
3 4 5 2 8 6 1 7 9

Thanks for using Sudoku Solver!
```