# sudoku\_solver.py

def print\_board(board):

for r in range(len(board)):

if r % 3 == 0 and r != 0:

print("- - - - - - - - - - - -")

for c in range(len(board[0])):

if c % 3 == 0 and c != 0:

print("| ", end="")

print(board[r][c] if board[r][c] != 0 else ".", end=" ")

print()

def find\_empty(board):

for r in range(len(board)):

for c in range(len(board[0])):

if board[r][c] == 0:

return (r, c)

return None

def valid(board, num, pos):

row, col = pos

if num in board[row]: # Row check

return False

if num in [board[i][col] for i in range(9)]: # Col check

return False

box\_x = col // 3

box\_y = row // 3

for i in range(box\_y\*3, box\_y\*3 + 3):

for j in range(box\_x\*3, box\_x\*3 + 3):

if board[i][j] == num:

return False

return True

def solve(board):

empty = find\_empty(board)

if not empty:

return True

row, col = empty

for num in range(1, 10):

if valid(board, num, (row, col)):

board[row][col] = num

if solve(board):

return True

board[row][col] = 0

return False

# Example board

sudoku\_board = [

[7, 8, 0, 4, 0, 0, 1, 2, 0],

[6, 0, 0, 0, 7, 5, 0, 0, 9],

[0, 0, 0, 6, 0, 1, 0, 7, 8],

[0, 0, 7, 0, 4, 0, 2, 6, 0],

[0, 0, 1, 0, 5, 0, 9, 3, 0],

[9, 0, 4, 0, 6, 0, 0, 0, 5],

[0, 7, 0, 3, 0, 0, 0, 1, 2],

[1, 2, 0, 0, 0, 7, 4, 0, 0],

[0, 4, 9, 2, 0, 6, 0, 0, 7]

]

print\_board(sudoku\_board)

solve(sudoku\_board)

print("\nSolved Sudoku:\n")

print\_board(sudoku\_board)