

Name:Snehal S Patil

TrackCode:SD

Task3:Implement a Sudoku Solver

```

1 def print_board(board):
2     for i in range(len(board)):
3         if i % 3 == 0 and i != 0:
4             print("- - - - -")
5
6         for j in range(len(board[0])):
7             if j % 3 == 0 and j != 0:
8                 print(" | ", end="")
9
10            if j == 8:
11                print(board[i][j])
12            else:
13                print(str(board[i][j]) + " ", end="")
14
15 def find_empty_location(board):
16     for row in range(len(board)):
17         for col in range(len(board[0])):
18             if board[row][col] == 0:
19                 return (row, col)
20     return None
21
22 def is_valid(board, num, pos):
23     # Check row
24     for i in range(len(board[0])):
25         if board[pos[0]][i] == num and pos[1] != i:
26             return False
27
28     # Check column
29     for i in range(len(board)):
30         if board[i][pos[1]] == num and pos[0] != i:
31             return False
32
33     # Check 3x3 box
34     box_x = pos[1] // 3
35     box_y = pos[0] // 3
36
37     for i in range(box_y * 3, box_y * 3 + 3):
38         for j in range(box_x * 3, box_x * 3 + 3):
39             if board[i][j] == num and (i, j) != pos:
40                 return False
41
42     return True
43
44 def solve_sudoku(board):
45     find = find_empty_location(board)
46     if not find:
47         return True
48     else:
49         row, col = find
50
51     for i in range(1, 10):
52         if is_valid(board, i, (row, col)):
53             board[row][col] = i
54
55             if solve_sudoku(board):
56                 return True
57
58             board[row][col] = 0
59
60     return False
61
62 # Example usage:
63 if __name__ == "__main__":
64     # Example board to solve
65     board = [
66         [5, 3, 0, 0, 7, 0, 0, 0, 0],
67         [6, 0, 0, 1, 9, 5, 0, 0, 0],
68         [0, 9, 8, 0, 0, 0, 0, 6, 0],
69         [8, 0, 0, 0, 6, 0, 0, 0, 3],
70         [4, 0, 0, 0, 0, 0, 0, 0, 1]

```

```

70         [4, 0, 0, 0, 0, 0, 0, 0, 1],
71         [7, 0, 0, 0, 2, 0, 0, 0, 6],
72         [0, 6, 0, 0, 0, 0, 2, 8, 0],
73         [0, 0, 0, 4, 1, 9, 0, 0, 5],
74         [0, 0, 0, 0, 8, 0, 0, 7, 9]
75     ]
76
77     print("Sudoku board to solve:")
78     print_board(board)
79
80     if solve_sudoku(board):
81         print("\nSolution:")
82         print_board(board)
83     else:
84         print("\nNo solution exists.")
85

```

→ Sudoku board to solve:

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

Solution:

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