



Student Name: Anshul Thakur

Branch: MCA

Semester: 1st

Subject Name: Design and analysis of algorithms

UID: 24MCI10221

Section/Group: 24MAM-4A

Date of Performance:

Subject Code: 24CAP-614

Q. Sudoku Game.

Aim/Overview of the practical:

The aim of creating a Sudoku solver project typically includes Algorithm Design and Optimization, Data Structures and Efficiency, Recursive Thinking etc.

1. Task to be done:

- Define the Data Structure.
- Input Parsing and Validation.
- Implement the Solving Algorithm.
- Constraint Checker.
- Testing and Debugging.

2. Code for experiment/practical:

```
public class sudoku {  
    public static boolean isSafe(int sudoku[][],int row, int col, int digit ) {  
        //column  
        for(int i = 0; i<= 8; i++) {  
            if(sudoku[i][col] == digit) {  
                return false;  
            }  
        }  
        //row  
        for(int j= 0; j<= 8; j++) {  
            if(sudoku[row][j] == digit) {  
                return false;  
            }  
        }  
    }  
}
```

```
//grid
int sr = (row/3)*3;
int sc = (col/3)*3;
for(int i = sr ; i<sr+3; i++) {
    for(int j = sc; j<sc+3; j++) {
        if(sudoku[i][j] == digit) {
            return false;
        }
    }
}
return true;
}

public static boolean sudokusolver(int sudoku[][], int row, int col){
    //base case
    if(row == 9 && col == 0){
        return true;
    }

    //recursion
    int nextRow = row, nextCol = col+1;
    if(col+1 == 9){
        nextRow = row+1;
        nextCol = 0;
    }
    if(sudoku[row][col] != 0) {
        return sudokusolver(sudoku, nextRow, nextCol);
    }
    for(int digit =1; digit <= 9; digit++) {
        if(isSafe(sudoku, row, col, digit)){
            sudoku[row][col] = digit;
            if(sudokusolver(sudoku, nextRow, nextCol)){
                return true;
            }
            sudoku[row][col] = 0;
        }
    }
}
```

```
}  
return false;  
  
}  
public static void printsudoku(int sudoku[][]) {  
for(int i = 0; i<9; i++) {  
    for(int j = 0; j<9;j++) {  
        System.out.print(sudoku[i][j]+" ");  
  
    }  
    System.out.println();  
}  
  
}  
public static void main (String args[]){  
int sudoku[][] = { {0, 0, 8, 0, 0, 0, 0, 0, 0},  
{4, 9, 0, 1, 5, 7, 0, 0, 2},  
{0, 0, 3, 0, 0, 4, 1, 9, 0},  
{1, 8, 5, 0, 6, 0, 0, 2, 0},  
{0, 0, 0, 0, 2, 0, 0, 6, 0},  
{9, 6, 0, 4, 0, 5, 3, 0, 0},  
{0, 3, 0, 0, 7, 2, 0, 0, 4},  
{0, 4, 9, 0, 3, 0, 0, 5, 7},  
{8, 2, 7, 0, 0, 9, 0, 1, 3} };  
if(sudokusolver(sudoku, 0, 0)){  
    System.out.println("Sol exists...");  
    printsudoku(sudoku);  
  
}else{  
    System.out.println("solution does not exist:");  
}  
}  
}
```

3. Result/Output/Writing Summary:

```
doku'  
Sol exists...  
2 1 8 3 9 6 7 4 5  
4 9 6 1 5 7 8 3 2  
7 5 3 2 8 4 1 9 6  
1 8 5 7 6 3 4 2 9  
3 7 4 9 2 8 5 6 1  
9 6 2 4 1 5 3 7 8  
5 3 1 6 7 2 9 8 4  
6 4 9 8 3 1 2 5 7  
8 2 7 5 4 9 6 1 3  
PS C:\Users\anshu\OneDrive\Desktop\practice>
```

Learning outcomes (What I have learnt):

- Understanding Algorithms and Data Structures.
- Recursive Thinking and Problem Solving.
- Practice in modular design, testing, and debugging.