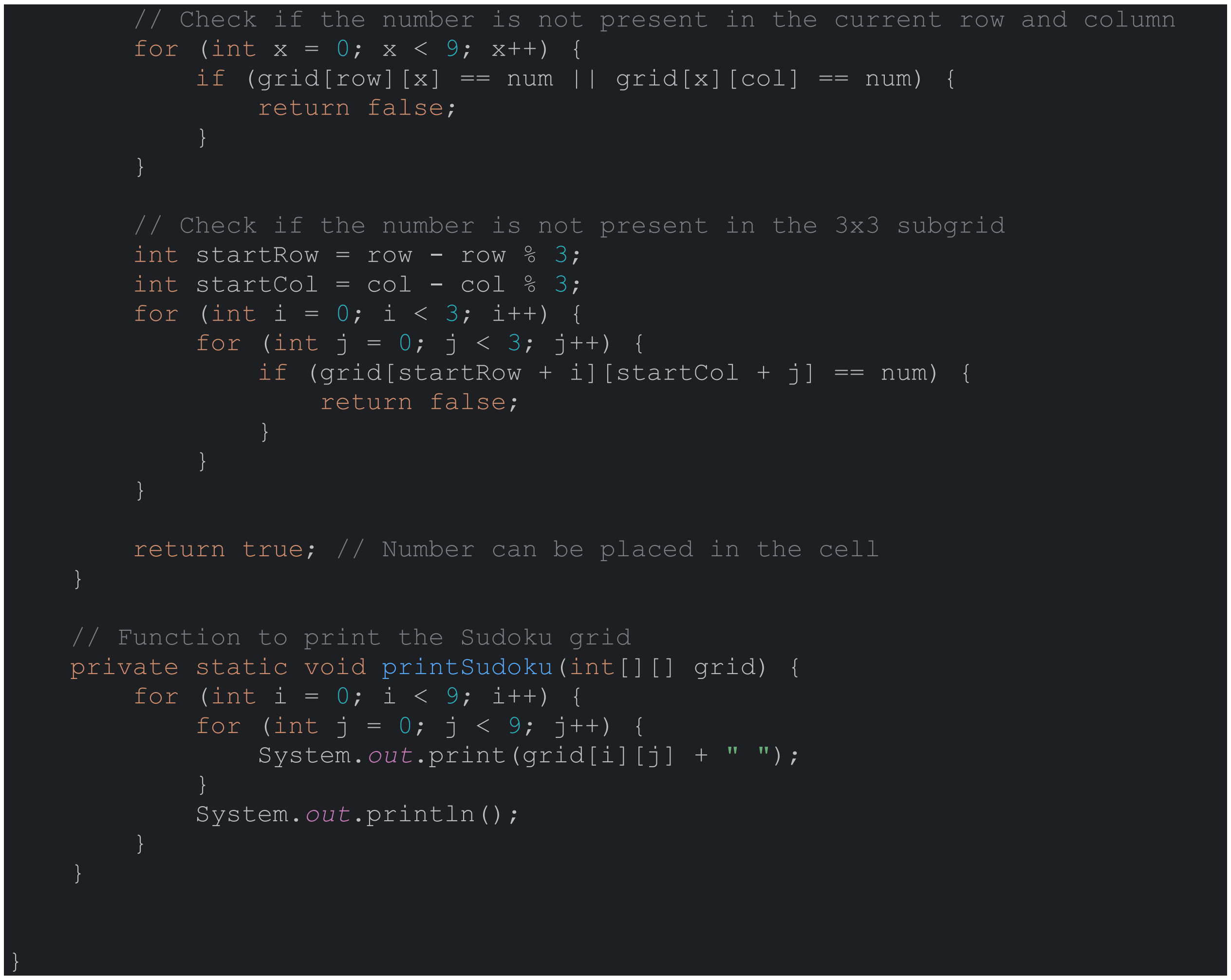
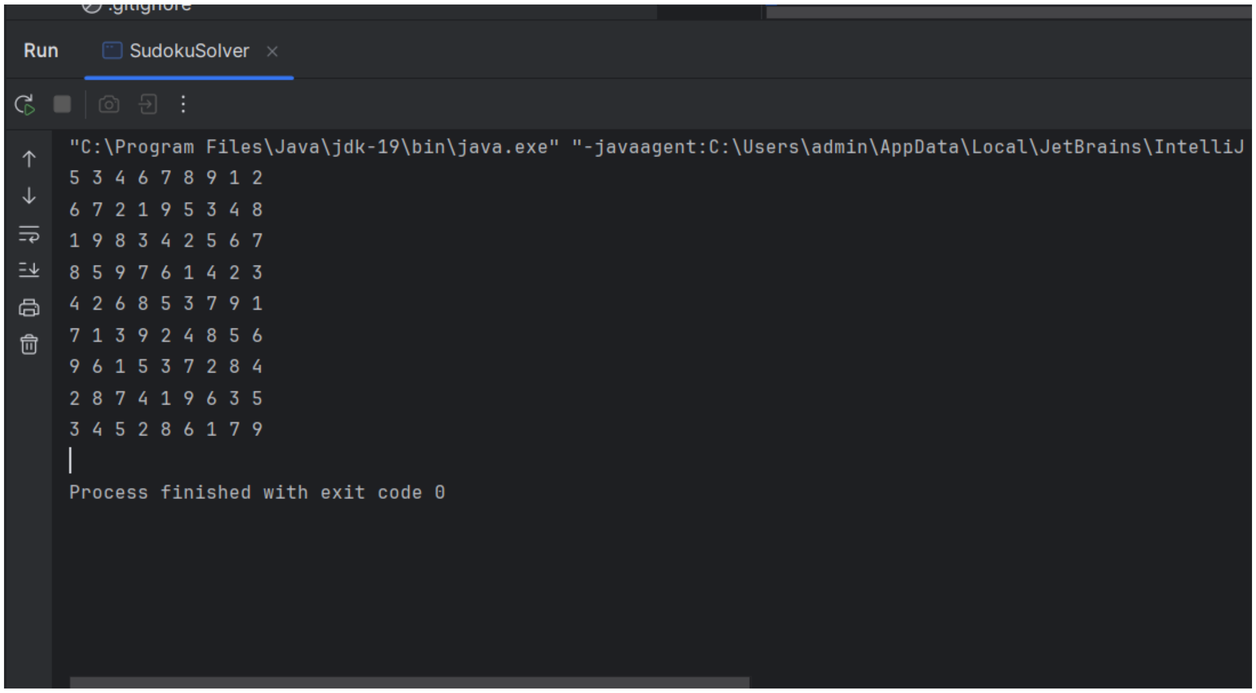
Code:

```java
package Sudoku;
import java.util.*;
public class SudokuSolver {

    public static void main(String[] args) {
        int[][] sudokuGrid = {
                {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 (solveSudoku(sudokuGrid)) {
            printSudoku(sudokuGrid);
        } else {
            System.out.println("No solution exists.");
        }
    }

    // Function to solve Sudoku using recursion
    private static boolean solveSudoku(int[][] grid) {
        for (int row = 0; row < 9; row++) {
            for (int col = 0; col < 9; col++) {
                if (grid[row][col] == 0) {
                    for (int num = 1; num <= 9; num++) {
                        if (isSafe(grid, row, col, num)) {
                            // Mark the cell with the current number
                            grid[row][col] = num;

                            // Recursively try to solve the rest of the
puzzle
                            if (solveSudoku(grid)) {
                                return true; // Success
                            }

                            // If the current assignment does not lead to a
solution, backtrack
                            grid[row][col] = 0;
                        }
                    }
                    return false; // No number is valid, backtrack
                }
            }
        }
        return true; // Puzzle solved
    }

    // Function to check if it's safe to place a number in a given cell
    private static boolean isSafe(int[][] grid, int row, int col, int num) {
```



```java
        // Check if the number is not present in the current row and column
        for (int x = 0; x < 9; x++) {
            if (grid[row][x] == num || grid[x][col] == num) {
                return false;
            }
        }

        // Check if the number is not present in the 3x3 subgrid
        int startRow = row - row % 3;
        int startCol = col - col % 3;
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                if (grid[startRow + i][startCol + j] == num) {
                    return false;
                }
            }
        }

        return true; // Number can be placed in the cell
    }

    // Function to print the Sudoku grid
    private static void printSudoku(int[][] grid) {
        for (int i = 0; i < 9; i++) {
            for (int j = 0; j < 9; j++) {
                System.out.print(grid[i][j] + " ");
            }
            System.out.println();
        }
    }


}
```

Output:



```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Users\admin\AppData\Local\JetBrains\IntelliJ
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

Process finished with exit code 0
```