Sudoku Game

- **Team Members**: - Pratham Patel, Nilay Patel

➢ Description of Sudoku Game In Java:-

Sudoku is very famous game which usually printed in newspapers and many people play thisgame while travelling and in their free time to increase the qi level of their brain.

➢ Information about Sudoku Game:-

sudoku, also known as **Su Doku**, popular form of number game. In its simplest and most common configuration, sudoku consists of a 9 × 9 grid withnumbers appearing in some of the squares. The object of the puzzle is to fill the remaining squares, using all the numbers 1–9 exactly once in each row, column, and the nine 3 × 3 subgrids.

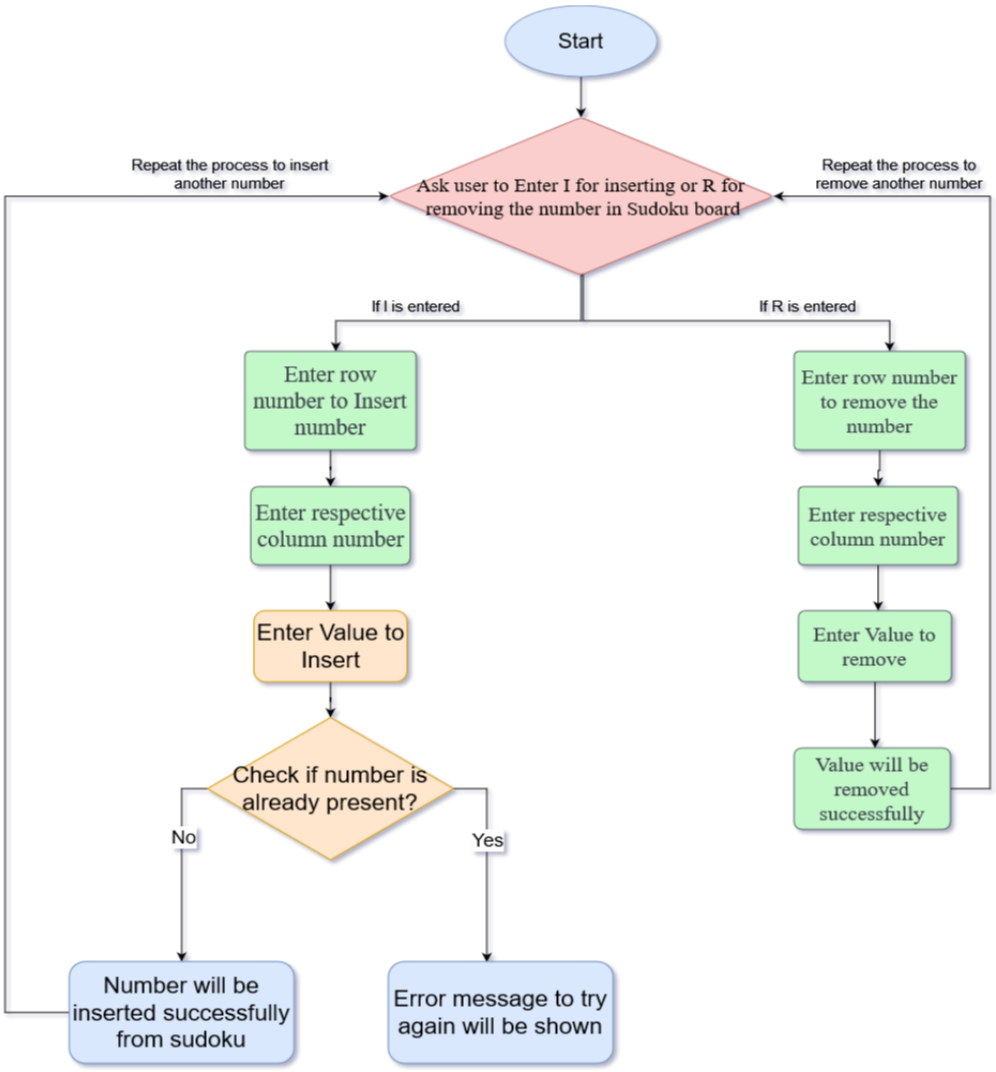
Sudoku is based entirely on logic, without any arithmetic involved, and the level of difficulty is determined by the quantity and positions of the original numbers. The puzzle, however, raised interesting combinatorial problems formathematicians, two of whom proved in 2005 that there are 6,670,903,752,021,072,936,960 possible sudoku grids.

➢ Functions Which Are Used In Project:-

- First Function is To create the Sudoku board and thewith the help of arrays we created 9*9 sudoku board.
- Second Function Is CheckRow Which Determine thatvalue which is inserted by player is already exist or not in the row.If value is already existed it shows thatvalue is already exist otherwise it will allow to add thevalue given by Player.

- Third Function Is check Col which determines that value which is inserted by player is already exist or not in the column. If value is already existed it showsthat value is already exist otherwise it will allow to add the value given by Player.
- Forth Function Is checkbox which determines that value which is inserted bye player is already exist or not in the Grid. If value is already existed it shows thatvalue is already exist otherwise it will allow to add the value given by Player.
- Fifth Function Is Insert value. This function is used to check that value which in inserted by player are in the valid row, column and grid.
- Sixth Function Is Remove value. This Function Is used replace value of the board with inserted value

➢ <u>Flow chart on "How the game works"</u>:-



➢ Code of the SudokuGame.java:-

```
public class SudokuGame
{
   public int[][] grid;

   SudokuGame()
   {
      grid = new int[9][9]; //empty
   }

   SudokuGame(String start){
      grid = new int[9][9];
      for(int x = 0; x <9; x++)
      {
         for(int y =0; y<9; y++)
         {
            grid[x][y] = 0;
         }
      }

      grid[0][0] = 5; //top left most
      grid[0][1] = 3;
      grid[0][4] = 7;
      grid[1][0] = 6;
      grid[1][3] = 1;
      grid[1][4] = 9;
      grid[1][5] = 5;
      grid[2][1] = 9;
      grid[2][2] = 8;
      grid[2][7] = 6;
      grid[3][0] = 8;
      grid[3][4] = 6;
      grid[3][8] = 3;
      grid[4][0] = 4;
      grid[4][3] = 8;
      grid[4][5] = 3;
      grid[4][8] = 1;
      grid[5][0] = 7;
      grid[5][4] = 2;
      grid[5][8] = 6;
```

```
        grid[6][1] = 6;
        grid[6][6] = 2;
        grid[6][7] = 8;
        grid[7][3] = 4;
        grid[7][4] = 1;
        grid[7][5] = 9;
        grid[7][8] = 5;
        grid[8][4] = 8;
        grid[8][7] = 7;
        grid[8][8] = 9; //bottom right most

    }

    public void printMySudokuGame()
    {
        for (int i = 0; i < 9; ++i)
        {
            if (i % 3 == 0)
                System.out.println(" ----------------------");
            for (int j = 0; j < 9; ++j)
            {
                if (j % 3 == 0)
                System.out.print("| ");
                System.out.print(grid[i][j] == 0 ? " ": Integer.toString(grid[i][j]));
                System.out.print(' ');
            }
            System.out.println("|");
        }
        System.out.println(" ----------------------");
    }

    public boolean insertVal(int row, int col, int myVal)
    {
        System.out.println("Entered insertVal " + "row " + row + "column
" + col + " myVal " + myVal);
        if(checkRow(row, col, myVal) == false)
            return false;

        if(checkCol(row, col, myVal) == false)
            return false;

        if(checkBox(row, col, myVal) == false)
```

```
        return false;

    grid[row][col] = myVal;


    return true;

}

public boolean removeVal(int row, int col)
{
    grid[row][col] = 0;

    return true;
}

private boolean checkRow(int row, int col, int myVal)
{
    for (int a = 0; a < 9; ++a)  // row
        if (myVal == grid[row][a])
        {
            System.out.println(myVal + " Already in Row: " + row);
            return false;
        }
    return true;

}
private boolean checkCol(int row, int col, int myVal)
{
    for (int b = 0; b < 9; ++b)  // column
        if (myVal == grid[b][col])
        {
            System.out.println(myVal + " Already in Column: " + col);
            return false;
        }
    return true;

}

private boolean checkBox(int row, int col, int myVal)
{
```

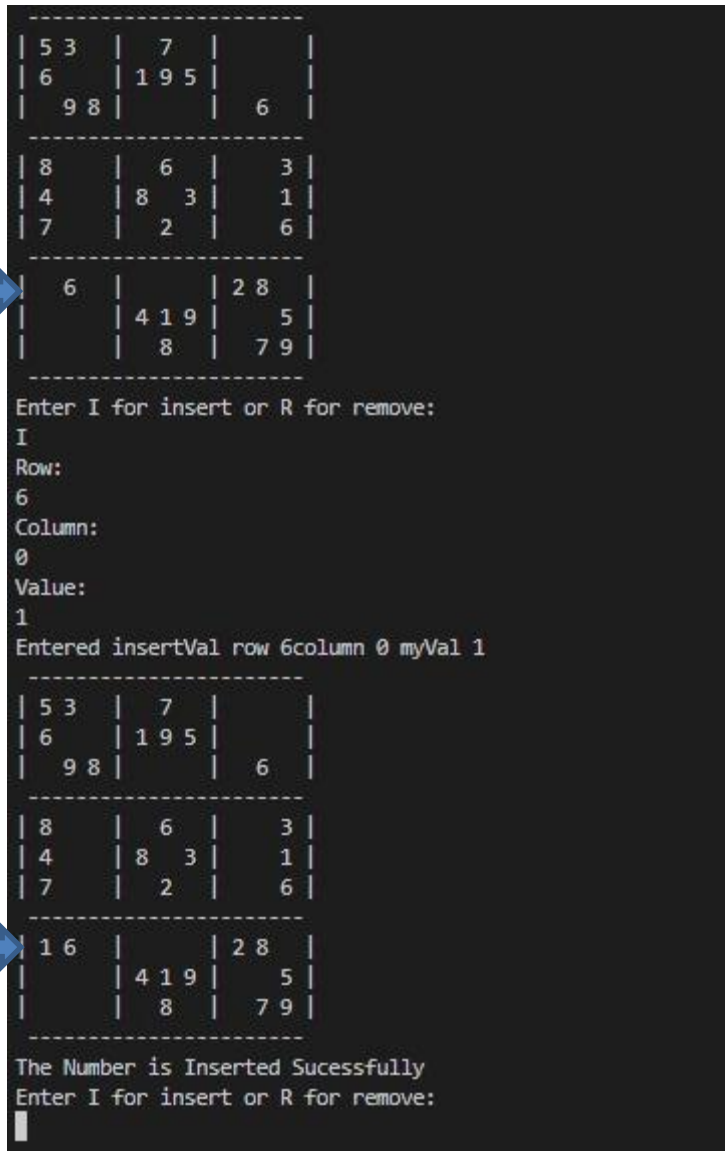
```
        int boxRowOffset = (row / 3)*3;
        int boxColOffset = (col / 3)*3;
        for (int c = 0; c < 3; ++c) // box
           for (int d = 0; d < 3; ++d)
              if (myVal == grid[boxRowOffset+c][boxColOffset+d])
              {
                 System.out.println(myVal + " Already in Box " );
                  return false;
              }
        return true;

    }
}
```

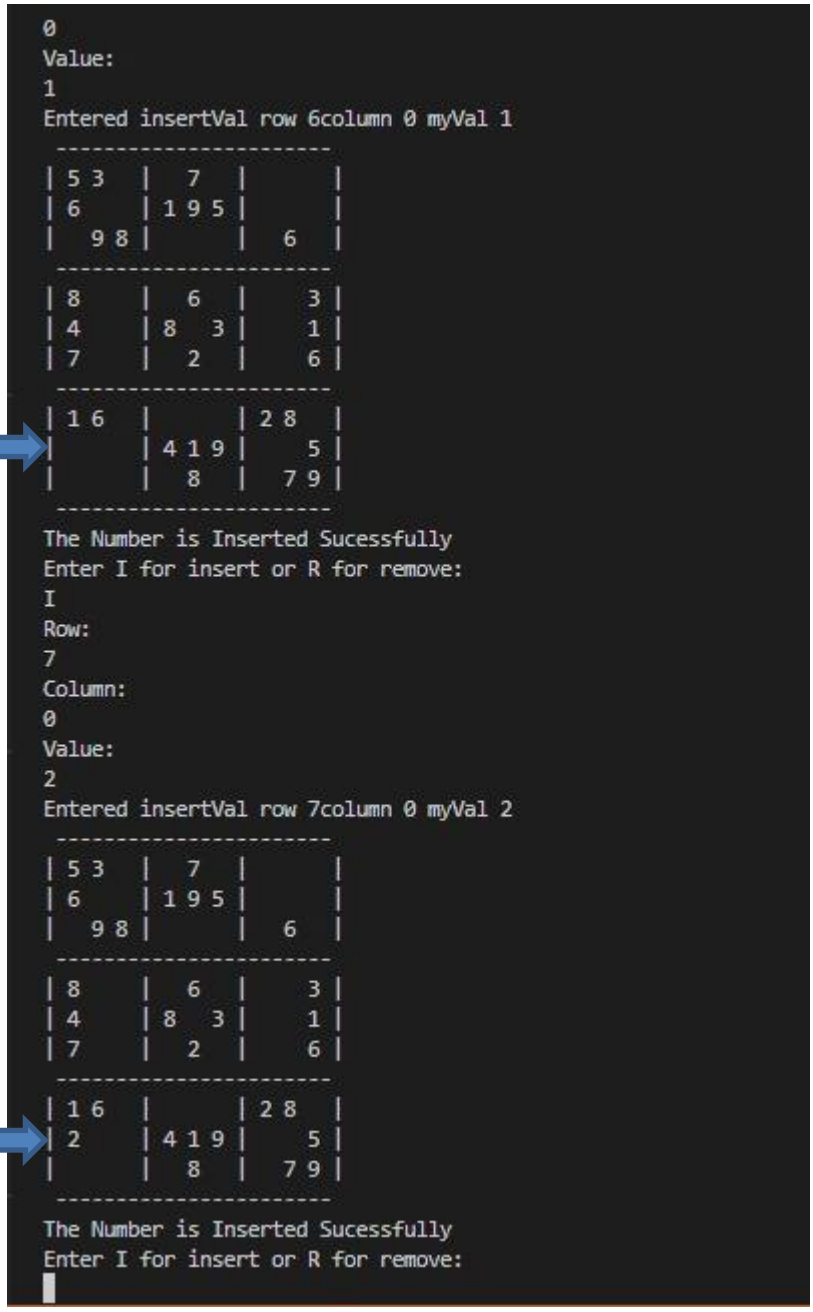
➢ Code of the GameTester.java:-

```
import java.util.Scanner;
  public class GameTester
  {
  public static void main(String args[])
  {
     SudokuGame i = new SudokuGame("start");
     i.printMySudokuGame();
     Scanner guess = new Scanner(System.in);
     {
      int row, col, val;
      String action;
        while(true)
        {
           System.out.println("Enter I for insert or R for remove: ");
           action = guess.next();

           System.out.println("Row: ");
           row = guess.nextInt();
           System.out.println("Column: ");
           col = guess.nextInt();

           if(action.equals("I"))
           {
              System.out.println("Value: ");
```

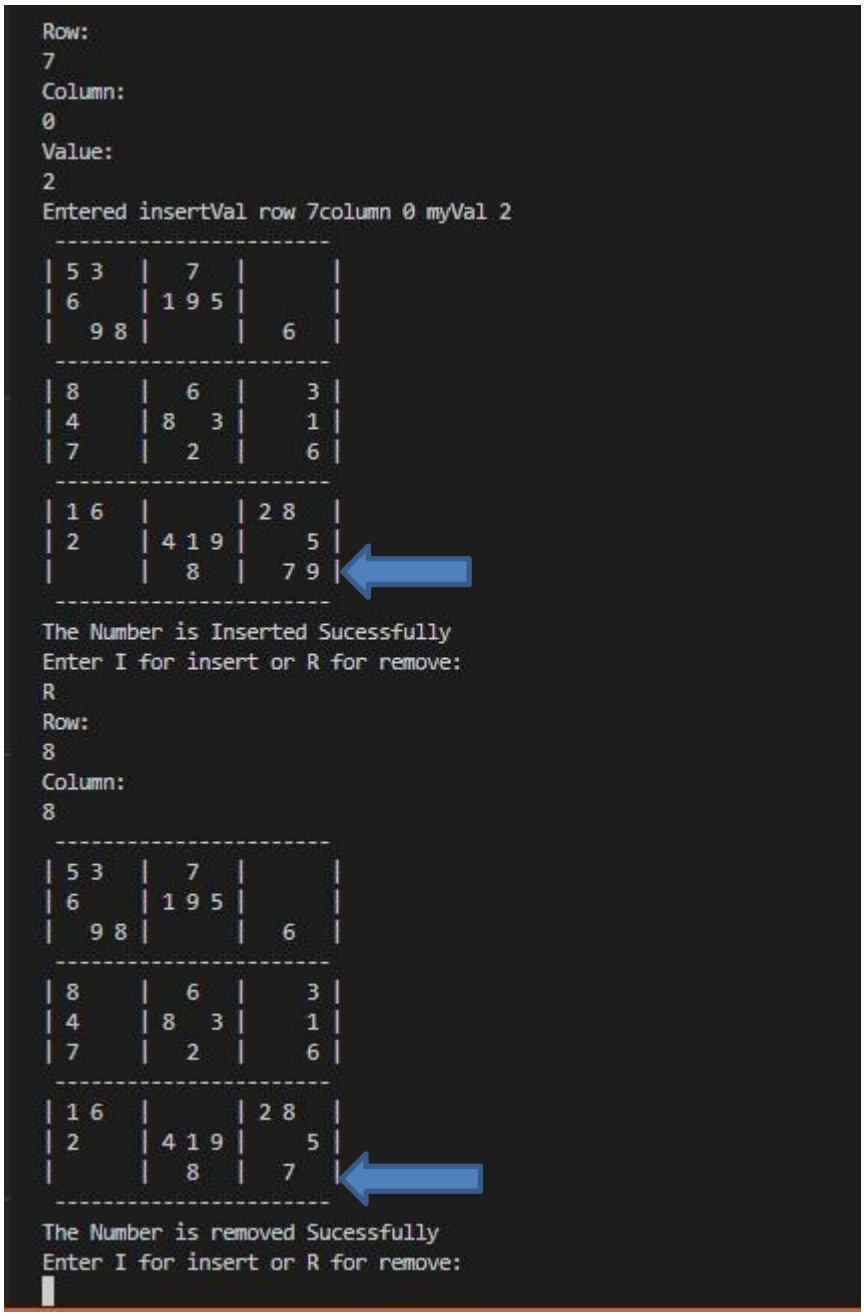
```
                val = guess.nextInt();
                if(i.insertVal(row, col, val))
                {
                    i.printMySudokuGame();
                    System.out.println("The Number is Inserted Sucessfully");
                }
                else
                {
                    i.printMySudokuGame();
                    System.out.println("Try Again, Number you have entered
is already present");
                }
            }
            else
            {
             i.removeVal(row, col);
             i.printMySudokuGame();
             System.out.println("The Number is removed Sucessfully");
            }
                //guess.close();
        }
    }
  }
```

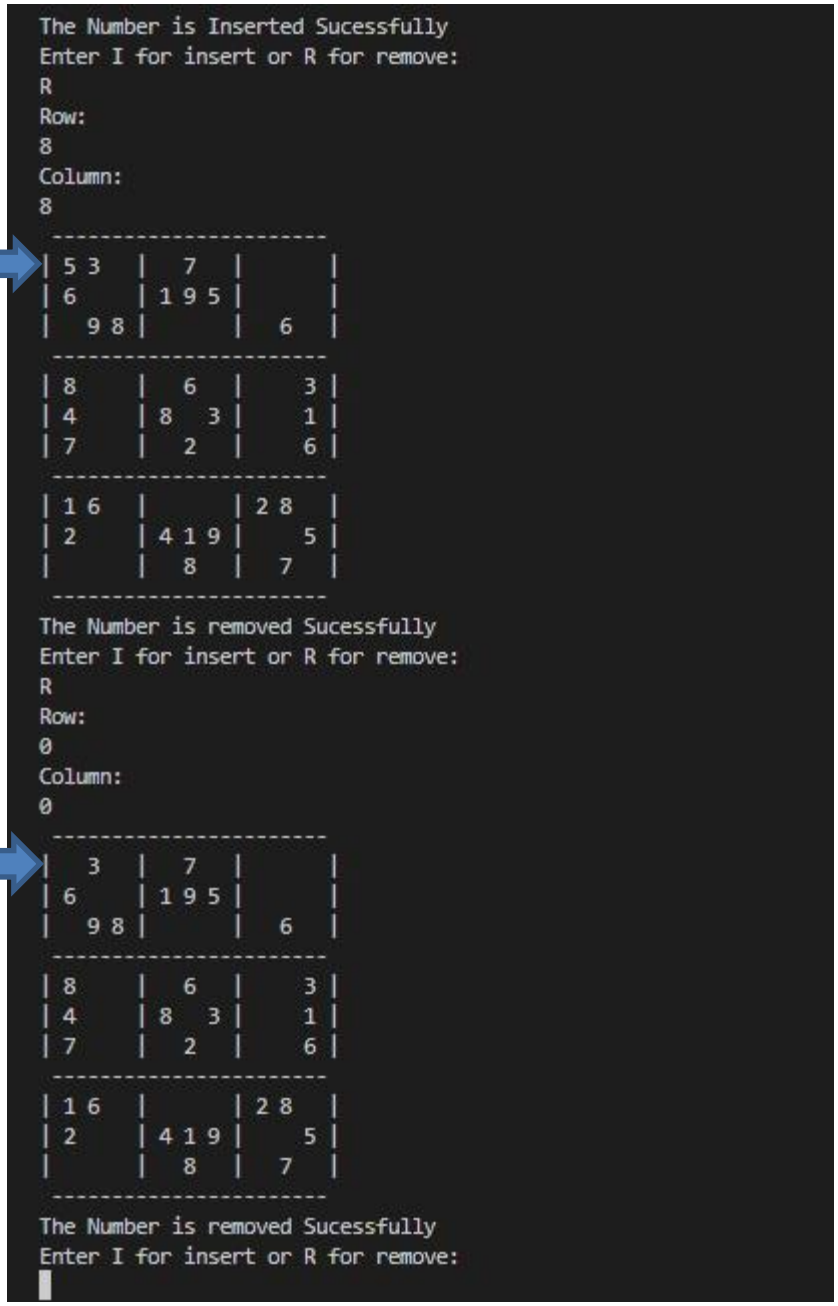
- **Output for inserting the number in the Sudoku board:-**




0
Value:
1
Entered insertVal row 6column 0 myVal 1
5 3	7	
6	1 9 5	
9 8		6
8	6	3
4	8 3	1
7	2	6
1 6		2 8
	4 1 9	5
	8	7 9
The Number is Inserted Sucessfully
Enter I for insert or R for remove:
I
Row:
7
Column:
0
Value:
2
Entered insertVal row 7column 0 myVal 2
5 3	7	
6	1 9 5	
9 8		6
8	6	3
4	8 3	1
7	2	6
1 6		2 8
2	4 1 9	5
	8	7 9
The Number is Inserted Sucessfully
Enter I for insert or R for remove:

➢ Output for removing the number in the Sudoku board:-





➢ Work Division in completing the Project:-

Both the team members have equally did the work in completing this game which includes logic building in code, flow chart and making the report for this project. We will equally present this project in the class.