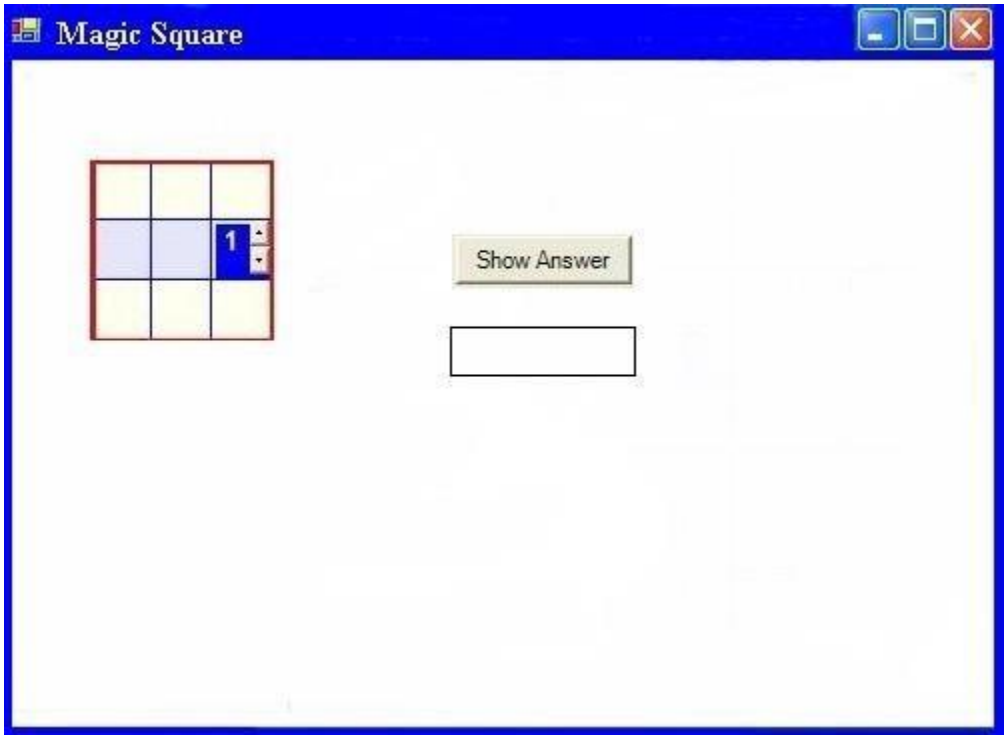
U18CO018
Shubham Shekhaliya
Assignment-3
Subject – ST

Write a java program to enter an integer number ‘n’. Create a grid of size n*n. Enter the value of each cell using a dropdown list. Check whether the created grid is a magic square or not. Display the result into the given text box by clicking on "Show Answer" button.

Magic Square:

A square matrix is said to be a Magic Square, if the sum of each row, each column and each diagonal is same.



Code:-

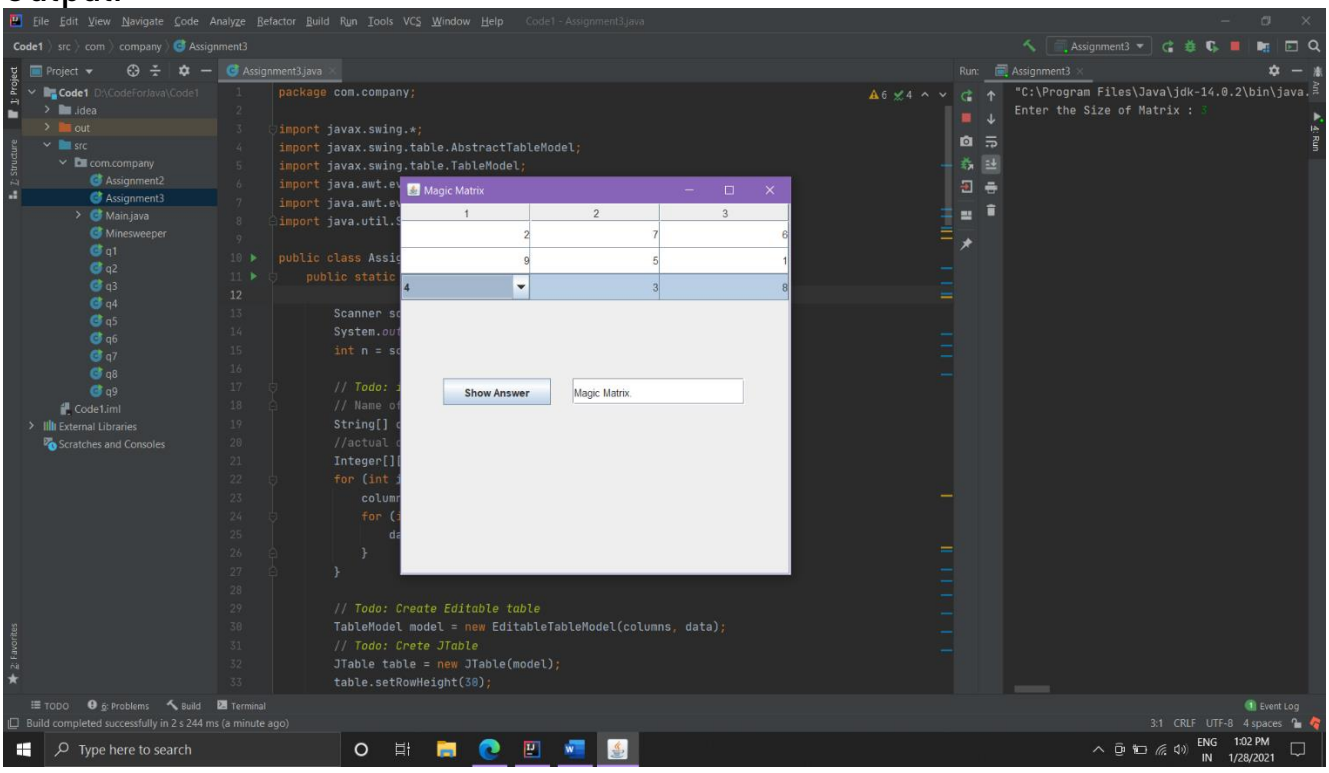
```
package com.company;

import javax.swing.*;
import javax.swing.table.AbstractTableModel;
import javax.swing.table.TableModel;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Scanner;

public class Assignment3 {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Size of Matrix : ");
        int n = sc.nextInt();

        // Todo: initialize matrix first time
        // Name of each column
        String[] columns = new String[n];
        //actual data for the table in a 2d array
        Integer[][] data = new Integer[n][n];
        for (int i = 0; i < n; i++) {
            columns[i] = String.valueOf(i + 1);
            for (int j = 0; j < n; j++) {
                data[i][j] = 0;
            }
        }

        // Todo: Create Editable table
        TableModel model = new EditableTableModel(columns, data);
        // Todo: Crete JTable
        JTable table = new JTable(model);
        table.setRowHeight(30);
        // todo: add DropDown List
        JComboBox comboBox = new JComboBox();
        for (int i = 0; i <= n * n; i++) {
            comboBox.addItem(i);
        }
        DefaultCellEditor editor = new DefaultCellEditor(comboBox);
        for (int i = 0; i < n; i++) {
            // added dropdown for each column
            table.getColumnModel().getColumn(i).setCellEditor(editor);
        }
```

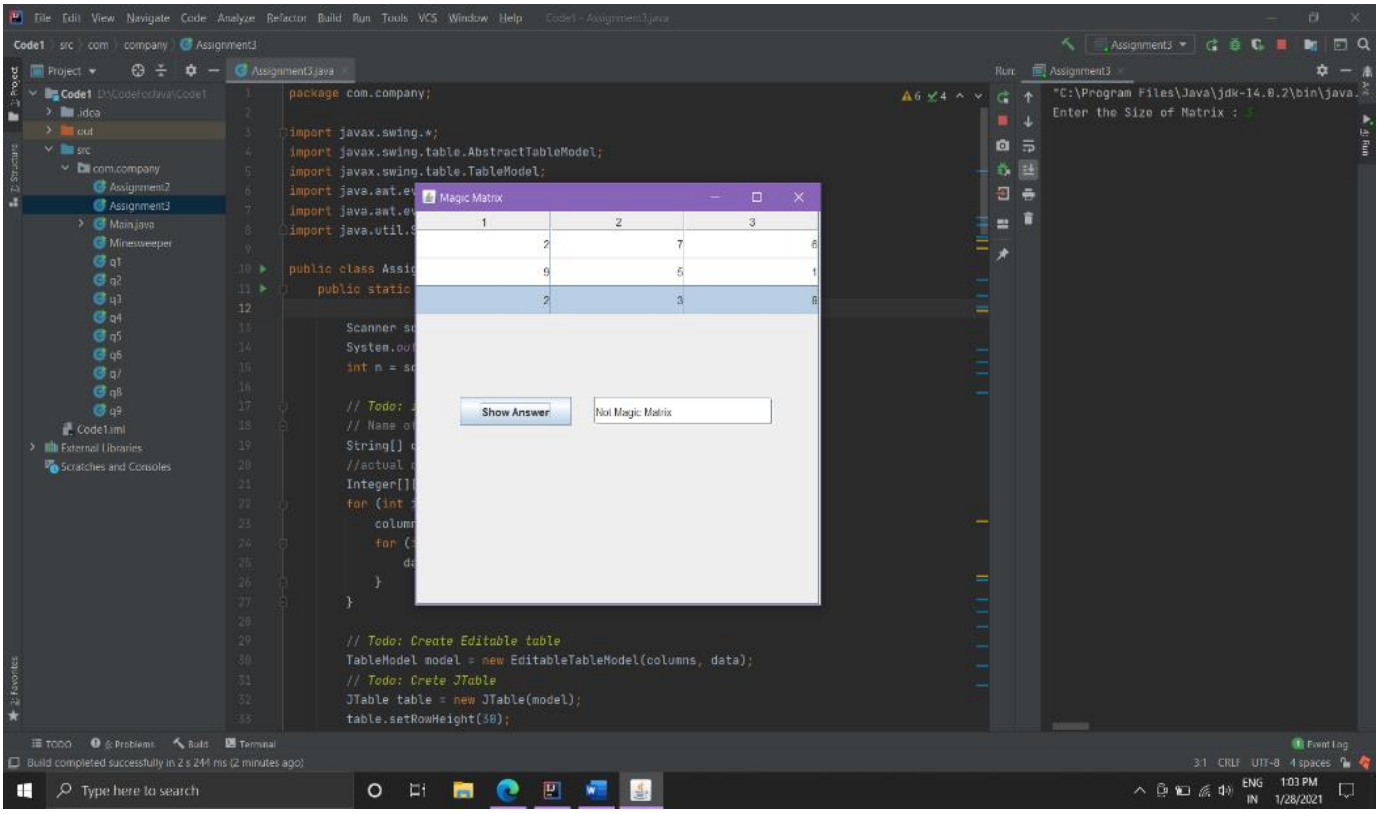
```
        //Todo: TextField to show answer
        JTextField tf = new JTextField();
        tf.setBounds(200, 200, 200, 30);

        //Todo: button to check matrix
        JButton button = new JButton("Show Answer");
        button.setBounds(50, 200, 125, 30);
        //Todo: Action Listenet to check matrix
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                if (CheckMatrix(data, n)) {
                    tf.setText("Magic Matrix.");
                } else {
                    tf.setText("Not Magic Matrix");
                }
            }
        });

        // Todo: Frame Creation
        JFrame frame = new JFrame();
        frame.setSize(600, 400);
        // Todo: crete scrollpane to add table in frame
        JScrollPane scrollpane = new JScrollPane(table);

        // Todo: add button, textField and table in fram
        frame.add(button);
        frame.add(tf);
        frame.add(scrollpane);

        // Todo: set constraint of frame
        frame.setTitle("Magic Matrix");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }

    // extend abstractTableModel to user for our purpose
    static class EditableTableModel extends AbstractTableModel {
        String[] columnTitles;
        Integer[][] dataEntries;

        public EditableTableModel(String[] columnTitles, Integer[][] dataEntries)
{
            this.columnTitles = columnTitles;
```

```
            this.dataEntries = dataEntries;
        }

        public int getRowCount() {
            return dataEntries.length;
        }

        public int getColumnCount() {
            return columnTitles.length;
        }

        public Integer getValueAt(int row, int column) {
            return dataEntries[row][column];
        }

        public String getColumnName(int column) {
            return columnTitles[column];
        }

        public Class getColumnClass(int column) {
            return getValueAt(0, column).getClass();
        }

        public boolean isCellEditable(int row, int column) {
            return true;
        }

        @Override
        public void setValueAt(Object aValue, int rowIndex, int columnIndex) {
            dataEntries[rowIndex][columnIndex] = (Integer) aValue;
        }
    }

    static boolean CheckMatrix(Integer mat[][], int N) {
        // todo: calculate the sum of the prime diagonal
        int sum = 0, sum2 = 0;
        for (int i = 0; i < N; i++)
            sum = sum + mat[i][i];

        // todo: sum of secondary diagonal
        for (int i = 0; i < N; i++)
            sum2 = sum2 + mat[i][N - 1 - i];
        // todo: both sum are same or not
        if (sum != sum2)
            return false;
```

```
        //todo: sums of each Rows
        for (int i = 0; i < N; i++) {
            int rowSum = 0;
            for (int j = 0; j < N; j++)
                rowSum += mat[i][j];
            // todo: if every row's sum is equal to prime diagonal sum
            if (rowSum != sum)
                return false;
        }

        //todo: sums of each Columns
        for (int i = 0; i < N; i++) {
            int colSum = 0;
            for (int j = 0; j < N; j++)
                colSum += mat[j][i];
            // todo: if every column's sum is equal to prime diagonal sum
            if (sum != colSum)
                return false;
        }
        return true;
    }
}
```

Output:-




Magic Matrix
1
2
3
2
7
6
9
5
1
4
3
8
Show Answer
Not Magic Matrix
"C:\Program Files\Java\jdk-14.0.2\bin\java.
Enter the Size of Matrix : 3
package com.company;
import javax.swing.*;
import javax.swing.table.AbstractTableModel;
import javax.swing.table.TableModel;
public class Assi
public static
// Todo:
// Name o
String[]
//actual
Integer[]
for (int
colum
for (
d
}
}
// Todo: Create Editable table
TableModel model = new EditableTableModel(columns, data);
// Todo: Crete JTable
JTable table = new JTable(model);
table.setRowHeight(30);
Build completed successfully in 2 s 244 ms (2 minutes ago)
Type here to search
ENG
IN
1:03 PM
1/28/2021