import java.util.Scanner;

public class EE305\_Project {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Enter 1 to test a relation or 0 to exit: ");

        int enter = input.nextInt();

        System.out.println();

        while (enter == 1) {

            System.out.print("Enter the number of rows:");

            int row = input.nextInt();

            System.out.print("Enter the number of columns:");

            int col = input.nextInt();

            System.out.println();

            int matrix[][] = new int[row][col];

            // Read the matrix values-----------------------

            System.out.println("(((((Enter the elements of the matrix)))))");

            for (int r = 0; r < col; r++) {

                System.out.println();

                for (int c = 0; c < row; c++) {

                    System.out.print("Enter the index (" + r + "," + c + "): ");

                    matrix[r][c] = input.nextInt();

                }

            }

            System.out.println();

            System.out.println("Your matrix is: ");

            // Display the elements of the matrix----------------

            // loop through array's rows

            for (int r = 0; r < matrix.length; r++) {

                // loop through columns of current row

                for (int c = 0; c < matrix[r].length; c++)

                    System.out.printf("%d ", matrix[r][c]);

                System.out.println();

            }

            // find Matrix^2 to find if transitive

            int[][] matrixSquared = multiplyMatrices(matrix, matrix, row, col);

            System.out.println("The matrix squared is: ");

            displayProduct(matrixSquared);

            System.out.println();

            System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

            System.out.println("Choose from the following tests: ");

            System.out.println("AllTest           = 1");

            System.out.println("SymmetricTest     = 2");

            System.out.println("AsymmetricTest    = 3");

            System.out.println("AntisymmetricTest = 4");

            System.out.println("ReflexiveTest     = 5");

            System.out.println("IrreflexiveTest   = 6");

            System.out.println("TransitiveTest    = 7");

            System.out.println("EquivalenceTest   = 8");

            int Test = input.nextInt();

            System.out.println();

            System.out.println("--------------------------------");

            switch (Test) {

            case 1:

                // All tests

                Symmetric(matrix);

                Asymmetric(matrix);

                Antisymmetric(matrix);

                Reflexive(matrix);

                Irreflexive(matrix);

                Transitive(matrix, matrixSquared);

                equivalence();

                break;

            case 2:

                // SymmetricTest

                Symmetric(matrix);

                break;

            case 3:

                // AsymmetricTest

                Asymmetric(matrix);

                break;

            case 4:

                // AntisymmetricTest

                Antisymmetric(matrix);

                break;

            case 5:

                // ReflexiveTest

                Reflexive(matrix);

                break;

            case 6:

                // IrreflexiveTest

                Irreflexive(matrix);

                break;

            case 7:

                // TransitiveTest

                Transitive(matrix, matrixSquared);

                break;

            case 8:

                // EquivalenceTest

                equivalence();

                break;

            }

            System.out.println("--------------------------------");

            System.out.print("Enter 1 to test another a relation or 0 to exit: ");

            enter = input.nextInt();

        }

    }

    // -----------------------functions--------------------

    // Display matrix

    public static void displayProduct(int[][] product) {

        for (int[] row : product) {

            for (int column : row) {

                System.out.print(column + "    ");

            }

            System.out.println();

        }

    }

    // Find M^2

    public static int[][] multiplyMatrices(int[][] matrix, int[][] matrix1, int row, int col) {

        int[][] product = new int[row][col];

        for (int i = 0; i < row; i++) {

            for (int j = 0; j < col; j++) {

                for (int k = 0; k < col; k++) {

                    product[i][j] += matrix[i][k] \* matrix[k][j];

                }

            }

        }

        for (int v = 0; v < row; v++) {

            for (int u = 0; u < col; u++) {

                if (product[u][v] >= 1)

                    product[u][v] = 1;

            }

        }

        return product;

    }

    // SymmetricTest

    public static boolean q\_for\_symmetric = true;

    public static void Symmetric(int array[][]) {

        for (int r = 0; r < array.length; r++) {

            for (int c = 0; c < array.length; c++) {

                if (r == c)

                    q\_for\_symmetric = true;

                else {

                    if (array[r][c] != array[c][r]) {

                        q\_for\_symmetric = false;

                        break;

                    }

                }

            }

        }

        if (q\_for\_symmetric == true)

            System.out.println("Symmetric");

        else

            System.out.println("Not Symmetric");

    }

    // AsymmetricTest

    public static void Asymmetric(int array[][]) {

        boolean q = true;

        for (int r = 0; r < array.length; r++) {

            for (int c = 0; c < array.length; c++) {

                if (r == c) {

                    if (array[r][c] != 0) {

                        q = false;

                        break;

                    } else {

                        if ((array[r][c] == 1) && (array[c][r] == 1)) {

                            q = false;

                            break;

                        }

                    }

                }

            }

        }

        if (q == true)

            System.out.println("Asymmetric");

        else

            System.out.println("Not Asymmetric");

    }

    // AntisymmetricTest

    public static void Antisymmetric(int array[][]) {

        boolean q = true;

        for (int r = 0; r < array.length; r++) {

            for (int c = 0; c < array.length; c++) {

                if (r == c)

                    q = true;

                else {

                    if ((array[r][c] == 1) && (array[c][r] == 1)) {

                        q = false;

                        break;

                    }

                }

            }

        }

        if (q == true)

            System.out.println("Antisymmetric");

        else

            System.out.println("Not Antisymmetric");

    }

    public static boolean q\_for\_reflexive = true;

    // ReflexiveTest

    public static void Reflexive(int array[][]) {

        for (int i = 0; i < array.length; i++) {

            if (array[i][i] != 1) {

                q\_for\_reflexive = false;

                break;

            }

        }

        if (q\_for\_reflexive == true)

            System.out.println("Reflexsive");

        else

            System.out.println("Not Reflexsive");

    }

    // IrreflexiveTest

    public static void Irreflexive(int array[][]) {

        boolean q = true;

        for (int i = 0; i < array.length; i++) {

            if (array[i][i] != 0) {

                q = false;

                break;

            }

        }

        if (q == true)

            System.out.println("Irreflexive");

        else

            System.out.println("Not Irreflexive");

    }

    // TransitiveTest

    public static boolean q\_for\_transitive = true;

    public static void Transitive(int array[][], int array2[][]) {

        for (int r = 0; r < array.length; r++) {

            for (int c = 0; c < array.length; c++) {

                if (array[r][c] != array2[r][c]) {

                    q\_for\_transitive = false;

                    break;

                }

            }

        }

        if (q\_for\_transitive == true)

            System.out.println("Transitive");

        else

            System.out.println("Not Transitive");

    }

    // EquivalenceTest

    public static void equivalence() {

        if (q\_for\_symmetric == true && q\_for\_reflexive == true && q\_for\_transitive == true)

            System.out.println("Equivalence");

        else

            System.out.println("Not Equivalence");

    }

}