Korn Rojrattanapanya (64010009)

```java
package labs.lab3.pro3;

import java.util.Scanner;

public class Lab3_Pro3_64010009 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int matrix_size;

        while (true) {

            System.out.print("Enter the size for the matrix: ");
            matrix_size = scanner.nextInt();

            if (matrix_size ≥ 2) break;
            System.out.println("ERROR: matrix size cannot be less than 2");
        }
        scanner.close();

        int[][] matrix = new int[matrix_size][matrix_size];

        for (int y = 0; y < matrix_size; y++) {
            for (int x = 0; x < matrix_size; x++) {

                matrix[y][x] = (int) (Math.random() * 2);
                System.out.print(matrix[y][x]);
            }
            System.out.print("\n");
        }

        findDuplicateOnRow(matrix);
        findDuplicateOnColumn(matrix);
        findDuplicateOnSuperDiagonal(matrix);
        findDuplicateOnDiagonal(matrix);
        findDuplicateOnSubDiagonal(matrix);
    }

    public static void findDuplicateOnRow(int[][] matrix) {

        boolean is_found = false;
        for (int y = 0; y < matrix.length; y++) {

            int sum = 0;
            for (int x = 0; x < matrix.length; x++) sum += matrix[y][x];

            if (sum == 0) System.out.println("All 0s on row " + y);
            if (sum == matrix.length) System.out.println("All 1s on row " + y);
            if (sum == 0 || sum == matrix.length) is_found = true;
        }

        if (!is_found) System.out.println("No same numbers on a row");
    }

    public static void findDuplicateOnColumn(int[][] matrix) {

        boolean is_found = false;
        for (int x = 0; x < matrix.length; x++) {

            int sum = 0;
            for (int[] rows : matrix) sum += rows[x];

            if (sum == 0) System.out.println("All 0s on column " + x);
            if (sum == matrix.length) System.out.println("All 1s on column " + x);
            if (sum == 0 || sum == matrix.length) is_found = true;
        }
```

```
        if (!is_found) System.out.println("No same numbers on a column");
    }

    public static void findDuplicateOnSuperDiagonal(int[][] matrix) {

        int sum = 0;
        for (int i = 0; i < matrix.length - 1; i++) sum += matrix[i][i + 1];

        if (sum == 0) System.out.println("All 0s on the super diagonal");
        else if (sum == matrix.length - 1) System.out.println("All 1s on the super diagonal");
        else System.out.println("No same numbers on the super diagonal");
    }

    public static void findDuplicateOnDiagonal(int[][] matrix) {

        int sum = 0;
        for (int i = 0; i < matrix.length; i++) sum += matrix[i][i];

        if (sum == 0) System.out.println("All 0s on the diagonal");
        else if (sum == matrix.length) System.out.println("All 1s on the diagonal");
        else System.out.println("No same numbers on the diagonal");
    }

    public static void findDuplicateOnSubDiagonal(int[][] matrix) {

        int sum = 0;
        for (int i = 0; i < matrix.length - 1; i++) sum += matrix[i + 1][i];

        if (sum == 0) System.out.println("All 0s on the sub diagonal");
        else if (sum == matrix.length - 1) System.out.println("All 1s on the sub diagonal");
        else System.out.println("No same numbers on the sub diagonal");
    }
}
```