CSE-2008 Assignment

Academic year: 2020-2021 **Semester:** WIN

Faculty Name: Mr. Sanket Mishra sir **Date:** 15 /11//2021

Student name: M.Taran **Reg. no.:** 19BCE7346

Program :

Write a program that prompts the user to enter the length of a square matrix, randomly fills in 0s and 1s into the matrix, prints the matrix, and finds the rows, columns, and diagonals with all 0s or 1s. Here is a sample run of the program:

Enter the size for the matrix: 4

0111

0000

0100

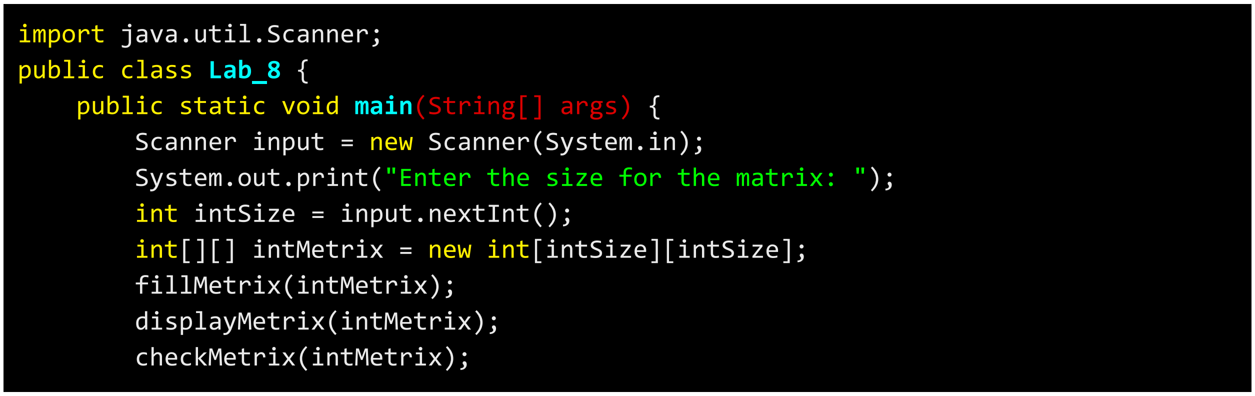
1111

T0: All 0s on row 1

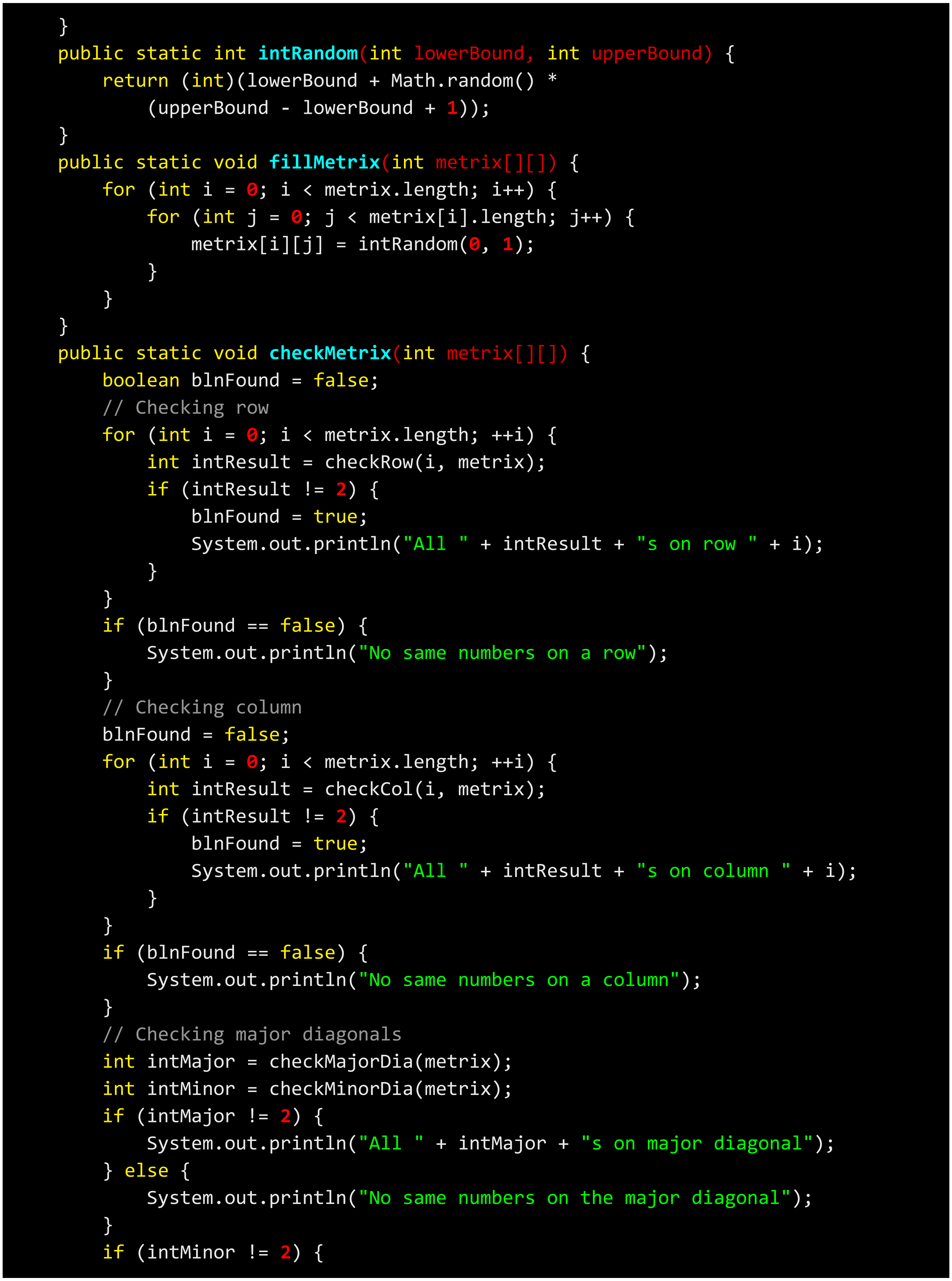
T1: All 1s on row 3

Annotate the child threads with names X and Y thus helping to know the order of execution

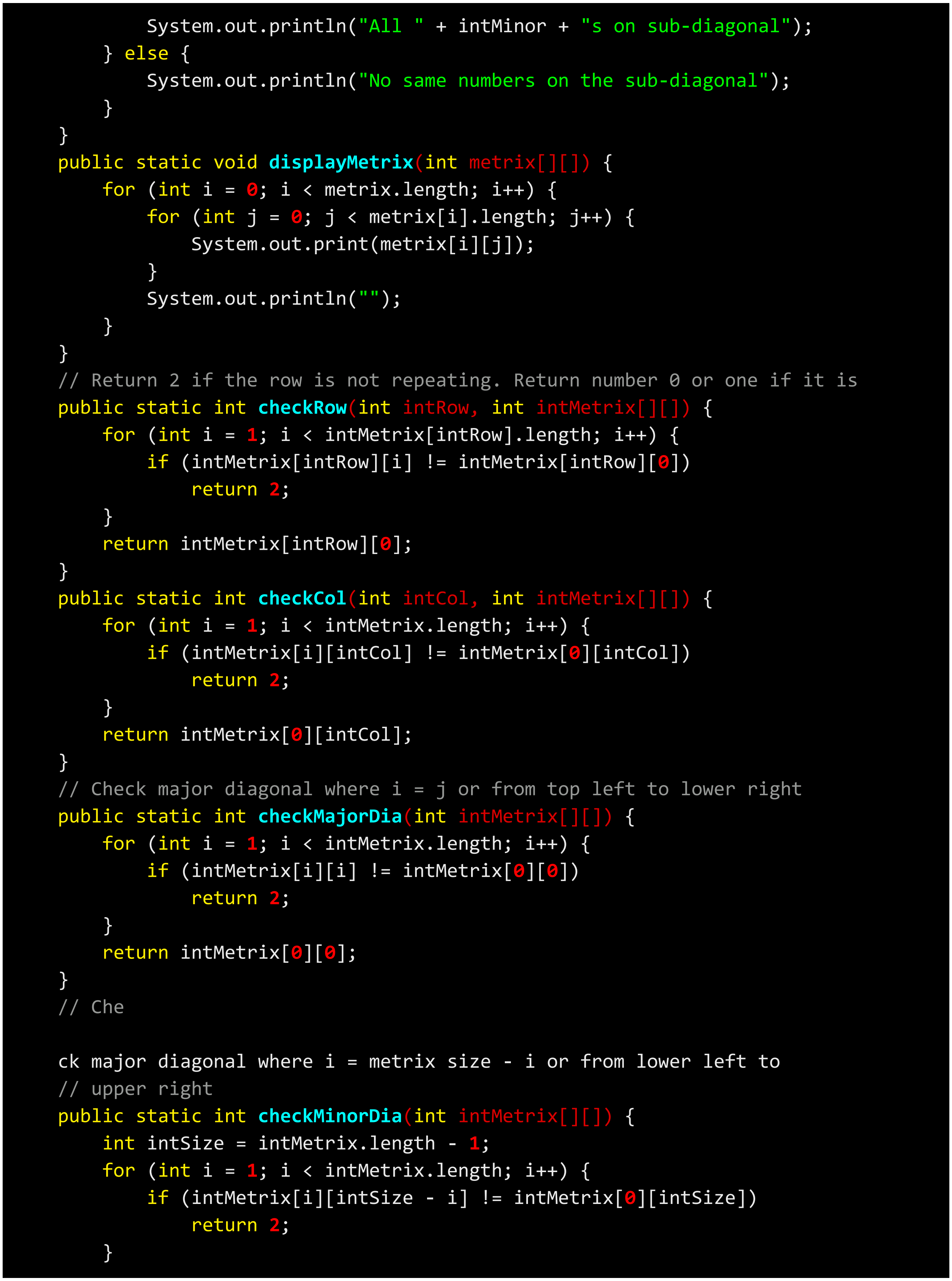
CODE:



```
import java.util.Scanner;
public class Lab_8 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the size for the matrix: ");
        int intSize = input.nextInt();
        int[][] intMetrix = new int[intSize][intSize];
        fillMetrix(intMetrix);
        displayMetrix(intMetrix);
        checkMetrix(intMetrix);
```



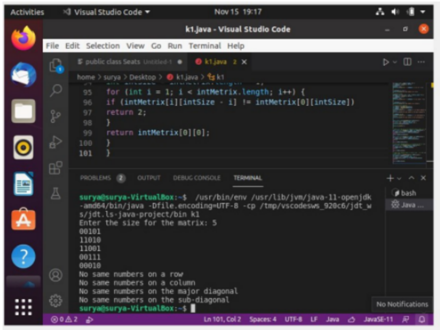
```
    }
    public static int intRandom(int lowerBound, int upperBound) {
        return (int)(lowerBound + Math.random() *
            (upperBound - lowerBound + 1));
    }
    public static void fillMetrix(int metrix[][]) {
        for (int i = 0; i < metrix.length; i++) {
            for (int j = 0; j < metrix[i].length; j++) {
                metrix[i][j] = intRandom(0, 1);
            }
        }
    }
    public static void checkMetrix(int metrix[][]) {
        boolean blnFound = false;
        // Checking row
        for (int i = 0; i < metrix.length; ++i) {
            int intResult = checkRow(i, metrix);
            if (intResult != 2) {
                blnFound = true;
                System.out.println("All " + intResult + "s on row " + i);
            }
        }
        if (blnFound == false) {
            System.out.println("No same numbers on a row");
        }
        // Checking column
        blnFound = false;
        for (int i = 0; i < metrix.length; ++i) {
            int intResult = checkCol(i, metrix);
            if (intResult != 2) {
                blnFound = true;
                System.out.println("All " + intResult + "s on column " + i);
            }
        }
        if (blnFound == false) {
            System.out.println("No same numbers on a column");
        }
        // Checking major diagonals
        int intMajor = checkMajorDia(metrix);
        int intMinor = checkMinorDia(metrix);
        if (intMajor != 2) {
            System.out.println("All " + intMajor + "s on major diagonal");
        } else {
            System.out.println("No same numbers on the major diagonal");
        }
        if (intMinor != 2) {
```



```
            System.out.println("All " + intMinor + "s on sub-diagonal");
        } else {
            System.out.println("No same numbers on the sub-diagonal");
        }
    }
    public static void displayMetrix(int metrix[][]) {
        for (int i = 0; i < metrix.length; i++) {
            for (int j = 0; j < metrix[i].length; j++) {
                System.out.print(metrix[i][j]);
            }
            System.out.println("");
        }
    }
    // Return 2 if the row is not repeating. Return number 0 or one if it is
    public static int checkRow(int intRow, int intMetrix[][]) {
        for (int i = 1; i < intMetrix[intRow].length; i++) {
            if (intMetrix[intRow][i] != intMetrix[intRow][0])
                return 2;
        }
        return intMetrix[intRow][0];
    }
    public static int checkCol(int intCol, int intMetrix[][]) {
        for (int i = 1; i < intMetrix.length; i++) {
            if (intMetrix[i][intCol] != intMetrix[0][intCol])
                return 2;
        }
        return intMetrix[0][intCol];
    }
    // Check major diagonal where i = j or from top left to lower right
    public static int checkMajorDia(int intMetrix[][]) {
        for (int i = 1; i < intMetrix.length; i++) {
            if (intMetrix[i][i] != intMetrix[0][0])
                return 2;
        }
        return intMetrix[0][0];
    }
    // Che

    ck major diagonal where i = metrix size - i or from lower left to
    // upper right
    public static int checkMinorDia(int intMetrix[][]) {
        int intSize = intMetrix.length - 1;
        for (int i = 1; i < intMetrix.length; i++) {
            if (intMetrix[i][intSize - i] != intMetrix[0][intSize])
                return 2;
        }
        return intMetrix[0][0];
    }
}
```

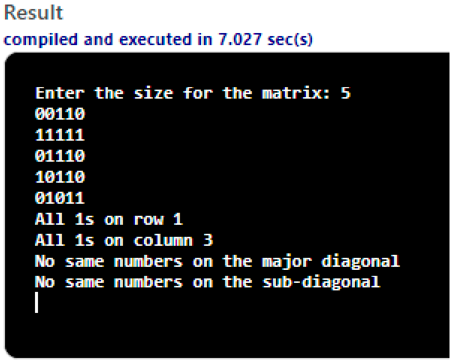
OUTPUT:



Activities
Visual Studio Code
Nov 15 19:17
k1.java - Visual Studio Code
File Edit Selection View Go Run Terminal Help
public class Seats Untitled-1
k1.java 2
home > surya > Desktop > k1.java > k1
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL
bash
Java ...
No Notifications
Ln 101, Col 2 Spaces: 4 UTF-8 LF Java JavaSE-11

```
for (int i = 1; i < intMetrix.length; i++) {
if (intMetrix[i][intSize - i] != intMetrix[0][intSize])
return 2;
}
return intMetrix[0][0];
}
}
```

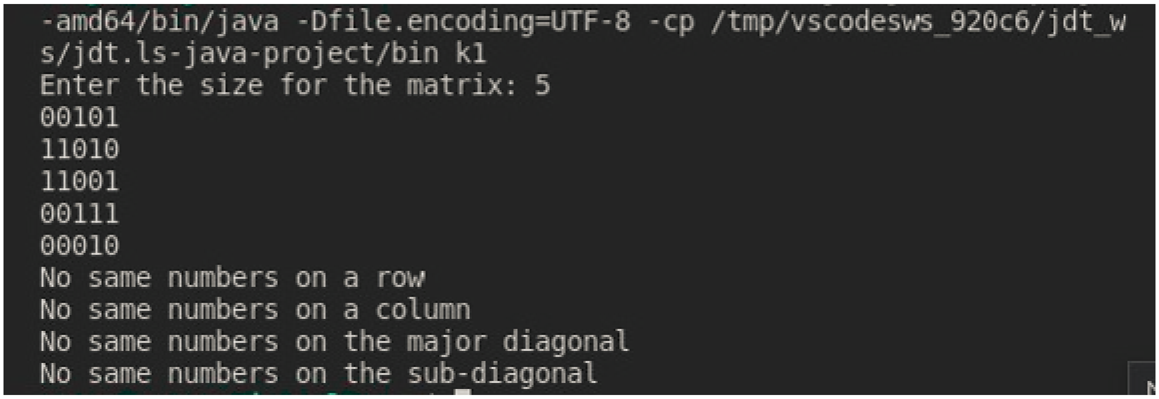
```
surya@surya-VirtualBox:~$  /usr/bin/env /usr/lib/jvm/java-11-openjdk
-amd64/bin/java -Dfile.encoding=UTF-8 -cp /tmp/vscodesws_920c6/jdt_w
s/jdt.ls-java-project/bin k1
Enter the size for the matrix: 5
00101
11010
11001
00111
00010
No same numbers on a row
No same numbers on a column
No same numbers on the major diagonal
No same numbers on the sub-diagonal
surya@surya-VirtualBox:~$
```



Result

compiled and executed in 7.027 sec(s)

```
Enter the size for the matrix: 5
00110
11111
01110
10110
01011
All 1s on row 1
All 1s on column 3
No same numbers on the major diagonal
No same numbers on the sub-diagonal
```



```
-amd64/bin/java -Dfile.encoding=UTF-8 -cp /tmp/vscodesws_920c6/jdt_w
s/jdt.ls-java-project/bin k1
Enter the size for the matrix: 5
00101
11010
11001
00111
00010
No same numbers on a row
No same numbers on a column
No same numbers on the major diagonal
No same numbers on the sub-diagonal
```