

מבוא לתכנות מונחה עצמים

סטודנט 1: דליה וויליאם

סטודנט 2: גיא רחמים

```
package sm2Lab1;

//Dalya William & Guy Rahamim
//Home Assignment 1.
import java.util.Random;

public class HomeAssignment1
{

    public static void main(String[] args)
    {

        {

            final int FOUR=4;
            final int MAX=21;

            // initializing an instance of the Random
            // and a 4x4 array.
            Random rand = new Random();
            int[][] array = new int[FOUR][FOUR];

            //Ex.5
            //initializing array with random integers
            //ranging from 0 to 20
            for (int i = 0; i < array.length; i++)
            {
                for (int j = 0; j <
array.length; j++)
```

```

        {
            array[i][j] =
rand.nextInt(MAX);
        }
    }

    //Ex.4
    show(array);
    System.out.println();

    //eEx.1
    checkMatrix(array);
    System.out.println();

    //Ex.2
    symmetricMatrix(array);
    System.out.println();

    //Ex.3
    positiveMatrix(array);
    }
}

// -----Question 1 -----//
// a function that check is a matrix is row sorted,
// column sorted or both.
public static void checkMatrix(int[][] array)
{
    boolean sortedRows = true, sortedCols = true;
    for (int i = 0; i < array.length; i++)
    {
        for (int j = 0; j < array.length - 1; j++)
        {

```

```

        bigger // if an element of a row is
                // than the one to its right,
                // the matrix is not row sorted.
        array[i][j]) if (array[i][j + 1] <=

                    sortedRows = false;

        bigger // if an element of a column is
                // than the one below it, the
        matrix // is not column sorted.
        array[j][i]) if (array[j + 1][i] <=

                    sortedCols = false;

                }

        }

        if (sortedCols && sortedRows)
            System.out.println("Matrix is perfectly sorted");
        else if (sortedCols)
            System.out.println("Matrix is column sorted");
        else if (sortedRows)
            System.out.println("Matrix is row sorted");
        else
            System.out.println("Matrix is nothing in
perticular");
    }

// -----Question 2 -----//
public static void symmetricMatrix(int[][] array)
{
    // if matrix is not square, it is defiantly not symmetric.

```

```

        if (array.length != array[0].length)
        {
            System.out.println("matrix is not squered,
therefore not symmetrical");
            return;
        }
        for (int i = 0; i < array.length; i++)
        {
            for (int j = 0; j < array.length; j++)
            {
                // if opposite elements of the
matrix are not equal,

                // the matrix is not symmetric.
                if (array[i][j] != array[j][i])
                {

                    System.out.println("matrix is not symmetrical!");

                    return;
                }
            }
        }
        // if the for loop managed to complete, the matrix is
symmetric.

        System.out.println("Matrix is symmetrical! hooray!");
    }

```

// -----Question 3 -----//

// a function that check if a matrix is "positive"

```
public static void positiveMatrix(int[][] array)
```

```
{
```

```
    // initializing variables.
```

```
    int sumEvenRows = 0, sumOddColumns = 0;
```

```

        for (int i = 0; i < array.length; i++)
        {
            for (int j = 0; j < array[0].length; j++)
            {
                // checking for even columns
                if (j % 2 == 1)
                    sumEvenRows +=
array[i][j];

                // checking for odd rows
                if (i % 2 == 0)
                    sumOddColumns +=
array[i][j];
            }
        }

        // if sumEvenRows is bigger than sumOddColumns, that
matrix is positive.

        if (sumEvenRows > sumOddColumns)
        {
            System.out.println("Matrix is positive!");
            return;
        } else
            System.out.println("Matrix is not positive. :(");
    }

// -----Question 4 -----//
// a function that takes in a matrix and prints its contents.
public static void show(int[][] array)
{
    for (int i = 0; i < array.length; i++)
    {

```

```
        for (int j = 0; j < array[0].length; j++)
        {
            System.out.print(array[i][j] +
"\t");

        }
        System.out.println();
    }
}
```

### Output example 1:

|    |    |    |    |
|----|----|----|----|
| 20 | 20 | 13 | 17 |
| 5  | 12 | 0  | 15 |
| 8  | 1  | 5  | 15 |
| 2  | 19 | 0  | 0  |

Matrix is nothing in perticular

matrix is not symmetrical!

Matrix is not positive. :(

### Output example 2:

|    |    |    |    |
|----|----|----|----|
| 14 | 5  | 16 | 11 |
| 6  | 14 | 16 | 17 |
| 15 | 18 | 17 | 1  |
| 12 | 7  | 3  | 5  |

Matrix is nothing in perticular

matrix is not symmetrical!

Matrix is not positive. :(

### Output example 3:

|   |    |    |    |
|---|----|----|----|
| 7 | 5  | 4  | 2  |
| 1 | 15 | 15 | 18 |
| 3 | 5  | 5  | 4  |
| 9 | 11 | 15 | 13 |

Matrix is nothing in perticular

matrix is not symmetrical!

Matrix is positive!

