Lab 11

1. Create a 2D array of size 5x5 and fill it with random numbers.

```
import java.util.Random;

public class Lab11 {

    public static void main(String[] args) {
        int[][] numbers = new int[5][5];

        for (int i = 0; i < numbers.length; i++) {
            for (int j = 0; j < numbers[i].length; j++) {
                numbers[i][j] = (int)(Math.random() * 10);
            }
        }

        for (int i = 0; i < numbers.length; i++) {
            for (int j = 0; j < numbers[i].length; j++) {
                System.out.print(numbers[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

2. Create a 2d array of size 3x3 and fill with integers from 0-8. Then copy the elements of this array to another 3x3 array that is rotated 90 degrees clockwise.

Here is the expected output:

```
Original array:
0  1  2
3  4  5
6  7  8
Rotated array:
6 3 0
7 4 1
8 5 2
```

```
public class Lab11 {

    public static void main(String[] args) {
        int[][] numbers = {
```

```
            {0, 1, 2},
            {3, 4, 5},
            {6, 7, 8}
        };

        int[][] rotated = new int[3][3];

        for (int i = 0; i < numbers.length; i++) {
            for (int j = 0; j < numbers[i].length; j++) {
                rotated[j][2 - i] = numbers[i][j];
            }
        }

        System.out.println("Original array:");
        for (int i = 0; i < numbers.length; i++) {
            for (int j = 0; j < numbers[i].length; j++) {
                System.out.print(numbers[i][j] + " ");
            }
            System.out.println();
        }

        System.out.println("Rotated array:");
        for (int i = 0; i < rotated.length; i++) {
            for (int j = 0; j < rotated[i].length; j++) {
                System.out.print(rotated[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

3. Create a 2D array of size 5x5 and fill it will random integers. Then pass the array as a parameter to a method that finds the the maximum value and it's position(row, column) in the array. Finally, return the three values to the main method and print them out.

```
import java.util.Random;

public class week11 {

    public static void main(String[] args) {
        int[][] numbers = new int[5][5];


        for (int i = 0; i < numbers.length; i++) {
            for (int j = 0; j < numbers[i].length; j++) {
                numbers[i][j] = (int)(Math.random() * 10);
            }
        }

        int[] max = findMax(numbers);
```

```
        printArray(numbers);
        System.out.println("Max value: " + max[0]);
        System.out.println("Row: " + max[1]);
        System.out.println("Column: " + max[2]);
    }

    public static int[] findMax(int[][] arr) {
        int[] max = {arr[0][0], 0, 0};

        for (int i = 0; i < arr.length; i++) {
            for (int j = 0; j < arr[i].length; j++) {
                if (arr[i][j] > max[0]) {
                    max[0] = arr[i][j];
                    max[1] = i;
                    max[2] = j;
                }
            }
        }

        return max;
    }

    public static void printArray(int[][] arr){
        for (int i = 0; i < arr.length; i++) {
            for (int j = 0; j < arr[i].length; j++) {
                System.out.print(arr[i][j]);
            }
            System.out.println();
    }
}
}
```

4. Create a 2D String array size 5x2. Then prompt the user to enter first and last names of 5 people. Store the first name in the first column and the last name in the second column. Finally, print out the full names of the people.

Hint: Use a for loop to prompt the user to enter the first and last names of 5 people. Use another for loop to print out the full names of the people.

Input:

```
Enter first name: John
Enter last name: Doe
Enter first name: Jane
Enter last name: Smith
Enter first name: Bob
Enter last name: Johnson
Enter first name: Alice
Enter last name: Brown
```

```
Enter first name: Mary
Enter last name: White
```

Output:

```
John Doe
Jane Smith
Bob Johnson
Alice Brown
Mary White
```

```java
import java.util.Scanner;

public class Lab11 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        String[][] names = new String[5][2];

        for (int i = 0; i < names.length; i++) {
            System.out.print("Enter first name: ");
            names[i][0] = input.nextLine();
            System.out.print("Enter last name: ");
            names[i][1] = input.nextLine();
        }

        for (int i = 0; i < names.length; i++) {
            System.out.println(names[i][0] + " " + names[i][1]);
        }
    }
}
```

5. Finish the question from the last slide of the 2D lecture slides. How do you check if a player has won tic-tac-toe?

```java
import java.util.Scanner;

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

        int[][] board = {{0,0,0},{0,0,0},{0,0,0}};

        while(true){
            printBoard(board);
            int[] move = getMove();
```

```
            setElement(board,move[0],move[1],move[2]);
            if(gameOver(board)){
                printBoard(board);
                System.out.println("Game Over! Player " + move[2] + " has
won!");
                break;
            }

        }
    }

    public static boolean gameOver(int[][] board){

        //Check Rows
        for(int i = 0; i < board.length; i++){
            if(board[i][0] == board[i][1] && board[i][1] == board[i][2] &&
board[i][0] != 0){
                return true;
            }
        }
        //Check Columns
        for(int i = 0; i < board.length; i++){
            if(board[0][i] == board[1][i] && board[1][i] == board[2][i] &&
board[0][i] != 0){
                return true;
            }
        }
        //Check Diagonals
        //top-left to bottom-right
         if(board[0][0] == board[1][1] && board[1][1] == board[2][2] &&
board[0][0] != 0){
                return true;
            }
        //top-right to bottom-left
         if(board[0][2] == board[1][1] && board[1][1] == board[2][0] &&
board[0][2] != 0){
                return true;
            }

        return false;

    }

    public static int[] getMove(){
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the row, column, and value");
        int row = input.nextInt();
        int column = input.nextInt();
        int value = input.nextInt();
        int[] move = {row, column, value};
        return move;
    }

    public static void setElement(int[][] board, int row, int column, int
```

```
value){
        board[row][column] = value;
    }

    public static void printBoard(int[][] board){
        for(int i = 0; i < board.length; i++){
            for(int j = 0; j < board[i].length; j++){
                System.out.print(board[i][j]);
            }
            System.out.println();
        }

    }

}
```