Lab8: 2d Array Tasks

Test your code as you go.
Remember to do null checks. The tasks with *'s on them are the ones where null checks are required, and the ones without you may omit if too difficult to add.

*1. Create a function called arrayToString() that takes an integer array and returns a string, formatted with commas and curly braces.
EX: {1, 2, 3}

```
//Insert Code with comments here:
```

*2. Create a function called arrayToString() that takes a 2D integer array and returns a string, formatted as below.

```
{{1, 2, 3},
 {3, 4, 5, 6},        // Notice the space at the front
 {7, 8, 9, 9, 9}}
```

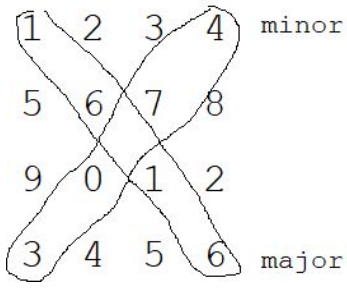
```
//Insert Code with comments here:
```

3. Create a function that takes a 3x3 2d int array and checks for TicTacToe (3 in a row). Assume blank = 0, x = 1, and o = -1.

```
//Insert Code with comments here:
```

*4. Create a function that calculates the total sum of all elements in a 2d array.
Do not assume it is a rectangular 2d array.

```
//Insert Code with comments here:
```



5. Create a function that calculates the sum of the major diagonal of a 2d array. May assume square 2d array.

Lab8: 2d Array Tasks

```
//Insert Code with comments here:
```

6. Create a function that calculates the sum of the minor diagonal of a 2d array. May assume square 2d array.

```
//Insert Code with comments here:
```

Create the following 3 functions to calculate the min and max element of a 2d array.

*7. Create a function that calculates the min of a 1d array.

```
public static int min(int[] a);
```

```
//Insert Code with comments here:
```

*8. Using 7, create a function that calculates the min of a 2d array.

```
public static int min(int[][] a);
```

```
//Insert Code with comments here:
```

*9. Not using 7, create a function that calculates the min of a 2d array.

```
public static int min2(int[][] a);
```

```
//Insert Code with comments here:
```

10. Create a function that rotates a square 2d array 90 degrees clockwise. Modify the original, don't create any new arrays.*

```
{{1 2 3}                {{7 4 1}
 {4 5 6}        =>       {8 5 2}
 {7 8 9}}                {9 6 3}}
// If this is too hard, you may assume the 2d array is 3x3.
```

```
//Insert Code with comments here:
```

11. Create a function that expands a 2d array. Each element of the 2d array expands into 4. Ex,

```
{{1,2,3},       => {{1,1,2,2,3,3},
 {4,5,6}}            {1,1,2,2,3,3},
                     {4,4,5,5,6,6},
                     {4,4,5,5,6,6}}
// You may assume no null values if too difficult.
```

Lab8: 2d Array Tasks

//Insert Code with comments here:

12. Create a function that prints out these patterns:

a.
```
1
12
123
1234
12345
```

b.
```
54321
4321
321
21
1
```

c.
```
5
44
333
2222
11111
```

//Insert Code with comments here:

13. Create a function that returns Pascal's Triangle with n rows, where n is an int argument.

```
int[][] pascalsTriangle(int n);
```

A spot has value 1 if it is on the edge of the triangle. Otherwise, it is the sum of the 2 elements above it.

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

It doesn't need to "look" nice, as long as the relevant information is stored systematically.

//Insert Code with comments here: