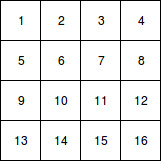
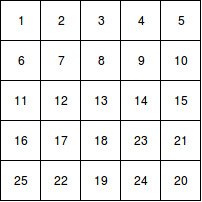
**Squarelotron**

The main assignment deals with more array manipulations.

We want you to program a bunch of operations for this object called a Squarelotron.

A Squarelotron consists basically of a matrix of numbers. This matrix can be decomposed as square rings which can flip independently in two different ways: Upside-Down and through the Main Diagonal.

[](https://www.draw.io/?scale=2#G0B6qIzWvdjOlWb1RoaHpaWjRGWlk)[](https://www.draw.io/?scale=2#G0B6qIzWvdjOlWQXN0QUptbmpVM1E)

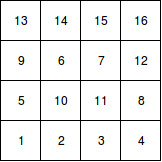
     a            b

For example, consider the following Squarelotrons:

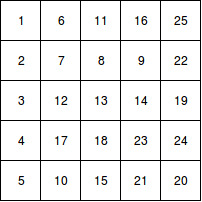
Squarelotron (a) has two rings. The outer ring contains the numbers 1, 2, 3, 4, 5, 8, 9, 12, 13, 14, 15, 16, while the inner ring contains the numbers 6, 7, 10, 11.

Squarelotron (b) has two rings and a center piece. The outer ring contains the numbers 1, 2, 3, 4, 5, 6, 10, 11, 15, 16, 21, 25, 22, 19, 24, 20, while the inner ring contains 7, 8, 9, 12, 14, 17, 18, 23. The number 13 is by itself in the center.

A Upside-Down Flip of the outer ring of Squarelotron (a) yields:

[](https://www.draw.io/?scale=2#G0B6qIzWvdjOlWZW5EUEF5YkxBVnM)

A Flip through the Main Diagonal of the outer ring of squarelotron b) yields:

[](https://www.draw.io/?scale=2#G0B6qIzWvdjOlWNzU2ZTZ6d1dvWjA)

Since the squarelotron is a physical object, it can be simply rotated. For example, if the top row of the 4x4 squarelotron contains (1, 2, 3, 4) and the squarelotron is rotated right by 90 degrees, then the rightmost column of the squarelotron will contain (1, 2, 3, 4). This is not considered a "flip."

Note that while we have only shown you 4 by 4 and 5 by 5 squarelotrons, the flip concept and rotation concept extends to any n by n squarelotron.

We will say that the maximum size of the squarelotron is 8 by 8.

We want to write a program that will help us flip and rotate squarelotrons.

We also want you to write this program in a test driven manner. For each of the methods we describe below, please begin by writing the stub of a method and then writing unit tests for it (in SquarelotronTest.java)

Make a class called Squarelotron with the following instance variables

1. int[][] squarelotron.
2. int size

**Constructors:**

Squarelotron should have a constructor.

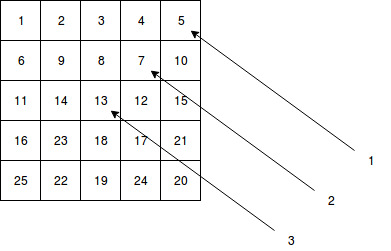
**Squarelotron(int n)** - It fills the 2-dimensional array with the numbers 1 to n squared, in order. It also sets the size instance variable to be n.

Note that you could do this with a switch statement for case 4,5,6,7 and 8 individually. That would not be considered wrong but you might want to pause and think about whether there is some kind of pattern that you can exploit which may be generalizable.

**Methods:**

In each of the following methods, the ring should be a number and we number from the outermost ring being the number 1.

See the illustration below for clarity

[](https://www.draw.io/?scale=2#G0B6qIzWvdjOlWNzU2ZTZ6d1dvWjA)

**Squarelotron upsideDownFlip(int ring)**

This method performs the Upside-Down Flip of the squarelotron, as described above, and returns the new squarelotron. The original squarelotron should not be modified (we will check for this).

**Squarelotron mainDiagonalFlip(int ring)**

This method performs the Main Diagonal Flip of the squarelotron, as described above, and returns the new squarelotron. The original squarelotron should not be modified (we will check for this).

**void rotateRight(int numberOfTurns)**

The argument numberOfTurns indicates the number of times the entire squarelotron should be rotated 90° clockwise. Any integer, including zero and negative integers, is allowable as the argument. A value of -1 indicates a 90° counterclockwise rotation. This method modifies the internal representation of the squarelotron; it does not create a new squarelotron.