# Challenge - DiagonalAdd

1 Hour

READ ALL INSTRUCTIONS. Don't forget the *Notes* and *Tips*.

## The Assignment

In this challenge you will write a Java program, called *DiagonalAdd*. You have one hour. (Keep track of your time!)

### DiagonalAdd

Write the class `DiagonalAdd` in the file `DiagonalAdd.java`. The main method of the `DiagonalAdd` class must prompt the user for input with the String "Enter the side length:". The program must then create a square 2D array (an array with an equal number of rows and columns) where the length of the side is equal to the user's input. Your program must automatically fill the array with numbers from zero up to one less than the square of the side length.

Example 1: If the side length is `2` the array should be of size 2x2 and look like:

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}$$

Example 2: If the side length is `3` the array should be of size 3x3 and look like:

$$\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix}$$

The program must print out this array, and the sum of all numbers in the *principal diagonal*, which starts in the top left corner, and ends in the bottom right. In *Example 1* above, the sum would be equal to 3 because 0 and 3 are the elements in the principal diagonal and 0+3=3. In *Example 2*, the sum would be equal to 12 because 0, 4 and 8 are the elements in the principal diagonal and 0+4+8=12. Your program must print output according the examples below.

### Expected Output

Look at these two examples to see how your console should look after the program executes. Your program must not print the `<in>` and `<out>`, those are there to show you which lines are input, and which lines are output.

**Example 1 - the user enters `3`**

```
<out>        Enter the side length:
<in>         3
<out>        The array created is:
<out>        [[0, 1, 2], [3, 4, 5], [6, 7, 8]]
<out>        The sum of the diagonal is:
<out>        12
```

**Example 2 - the user enters `4`**

```
<out>        Enter the side length:
<in>         4
<out>        The array created is:
<out>        [[0, 1, 2, 3], [4, 5, 6, 7], [8, 9, 10, 11], [12, 13, 14, 15]]
<out>        The sum of the diagonal is:
<out>        30
```

## Submitting to the Autograder

1. Use the Autograder to your advantage. Submit your code as many times as you like.

2. Make sure your program's output matches the expected output shown above.

3. Make sure that your program compiles and runs without any errors before submitting.

4. Create a `zip` file containing the `DiagonalAdd.java` file.

5. Upload the `zip` file to Athena.

6. The Autograder closes at 12:10.

## Tips

- Remember that your class names needs to be exactly the same as the name of the files in which they were defined.
- Do not include any package declarations in your submission file
- The `System.out.println` method automatically appends a newline character to the string it prints.
- You can use `scanner` for user input and you can read numbers using `sc.nextInt()`
- remember what data type is required to index an array and take note of whether you need to explicitly or implicitly convert between data types that you need.
- You can print out whole 2D arrays by importing `java.util.Arrays` and passing `Arrays.deepToString(ArrayName)` to the `println()` function.