

# Arizona State University - CSE205

## Assignment #9

### Due Date

Friday, March 31st, at 5:30pm

***Important: This is an individual assignment. Please do not collaborate.***

*No late assignment will be accepted.*

*Make sure that you write every line of your code. Using code written by someone else will be considered a violation of the academic integrity and will result in a report to the Dean's office.*

### Minimal Submitted Files

You are required, but not limited, to turn in the following source files:

Assignment9.java

### Requirements to get full credits in Documentation

1. The assignment number, your name, student ID, lecture number/time, and a class description need to be included at the top of each file/class.
2. A description of each method is also needed.
3. Some additional comments inside of methods (especially for the "main" method) to explain codes that are hard to follow should be written.

You can look at the Java programs in the text book to see how comments are added to programs.

**You are not allowed to use the Scanner class in this assignment and any assignment after this one. You will need to use InputStreamReader and BufferedReader (they are in java.io package) to process input and also take care of IOException.**

### New Skills to be Applied

In addition to what has been covered in previous assignments, the use of the following items, discussed in class, will probably be needed:

Recursion  
One-dimensional arrays

### Program Description

Assignment #9 will be the construction of a program that reads in a sequence of integers from standard input **until 0 is read**, and store them in an array (including 0). This is done using iteration (choose one of for, while, or do while loop). You may assume that there will not be more than 100 numbers.

Then compute the minimum number, compute the sum at even indexes (0, 2, 4, ...) count numbers that are divisible by 3 and compute the maximum number among the numbers that are less than the first number (You can assume that a user will enter at least one number before 0 is entered, thus at least two numbers will be entered including 0.) **using recursion**. Thus you will create recursive

methods `findMin`, `computeSumAtEvenIndexes`, `countDivisibleBy3`, and `findMaxOfLessThanFirst` in `Assignment9` class and they will be called by a main method.

Specifically, the following **recursive** methods must be implemented (These methods **should not contain any loop**):

```
public static int findMin(int[] numbers, int startIndex, int endIndex)
```

```
public static int computeSumAtEvenIndexes(int[] numbers, int startIndex, int endIndex)
```

```
public static int countDivisibleBy3(int[] numbers, int startIndex, int endIndex)
```

```
public static int findMaxOfLessThanFirst(int[] numbers, int startIndex, int endIndex, int firstNumber)
```

**If these methods are implemented using a Loop or any Static Variable, points will be deducted (from the test cases) even if your program passes test cases. DO NOT use any Static Variables.**

The program should output the results of those calculations to standard output. Your program will continue to read in numbers until the number 0 is entered. At this point, the calculations will be outputted in the following format:

The minimum number is 0

The sum of numbers at even indexes is 0

The count of numbers that are divisible by 3 is 0

The maximum number among numbers that are less than the first number is 0

Note that the result values will be different depending on test cases (not always 0).

**Do not output a prompt to query for the numbers.** The number 0 is included in the sequence of numbers and should be included in all of your calculations.

## Input

The following files are the test cases that will be used as input for your program (Right-click and use "Save As"):

[Test Case #1](#)

[Test Case #2](#)

[Test Case #3](#)

[Test Case #4](#)

## Output

The following files are the expected outputs of the corresponding input files from the previous section (Right-click and use "Save As"):

[Test Case #1](#)

[Test Case #2](#)

[Test Case #3](#)

[Test Case #4](#)