CSE 331 16wi Homework 0

due: Friday, Jan. 8, 2016 by **10 am**. ***No late submissions will be accepted***.

Write an algorithm to partition the elements in an integer array as described below and argue that your algorithm is correct. You ***should not*** run your algorithm or debug it on a computer. The idea of the exercise is to get it right by thinking and analyzing, not by executing, debugging, or experimenting.

Use Java syntax for your code, but do not write an entire Java program; just one method. Because you are not using a computer we will not grade on tiny syntax mistakes like forgetting a semicolon, but do your best to write a clear and correct algorithm.

Your algorithm should work as follows:

- The input is an array *b* containing *n* integer values in locations *b*[0] to *b*[*n*-1] where you may assume that *n* >= 1.
- Your algorithm should rearrange the original elements of the array so that all the negative (<0) elements in the original array, if any, appear at the front of the array starting in location *b*[0]; all of the 0 elements in the original array, if any, are moved to the middle; and all the positive (>0) elements, if any, follow that up to position *b*[*n*-1]. The array does not need to be sorted, just partitioned so the negative, zero, and positive elements are grouped together.
- You should not use any arithmetic or logical operations to compute new values for array elements or to assign constant values (like 0) to array elements. Use assignment statements to copy or swap the original values as needed. You may write *swap*(*x*,*y*) as a shorthand for interchanging the contents of variables *x* and *y* (including array elements like *b*[*i*]). You may use a small number of additional integer variables in your solution but no additional arrays. The idea is to rearrange the elements of the original array in place.
- Your solution should run in *O*(*n*) time, preferably without having to examine or traverse the array elements multiple times.
  Use the catalyst drop-box linked from the CSE 331 home page to turn in a pdf file containing your answer. The file should contain:

1. Your algorithm.
2. An argument that your algorithm is correct. You may use any informal, but precise, style you wish. If you were unable to completely solve the problem, for instance to create an algorithm that runs in *O*(*n*) time, you should describe those problems. Be brief.
3. The total amount of time you spent working on this problem.
   Be sure that your name appears at the top of your solution.

Note: Your solution file must be in pdf format, but it is ok to submit a scanned copy of a handwritten document as long as it is **legible** when printed and no larger than 2-3 MB or so.