1. Problems:

The first problem that I had was I wrote all of my for loops in the wrong format so I had to go back and change every one.

An obstacle that I faced was in counting and making sure that my loops were always working out to the position of the array that I intended to look at. What made this confusing was starting the array at 0. I worked around this by drawing it out on a piece of paper specific examples.

Another frequent obstacle I faced was terminating a certain loop after a specific position and making sure that the information that I obtained from that loop (the array2 was found completely in array1 or not) was not lost.

The last obstacle that I faced was changing around positions in the array and making sure that they did not lose and values while trying to write one to one position and one to another.

1. Test cases:
   * “” empty strings in all the tests to see if the functions handle them properly.
   * Test empty arrays with every function to see if they are handled properly.
   * Negative n values in any of the n positions to ensure it returns -1.
   * appendToAll
     1. “” appending to the end of the array to see if it does not change the array.
     2. Appending “!” to the array to test if it does append.
   * lookup
     1. Look up a value that is within the array once.
     2. Look up a value that is within the array multiple times to see if it returns the first position it appears.
     3. Look up a value that is not within the array.
   * positionOfMax
     1. Test an array that is made up all the same string and see if it returns the first position.
     2. Test an array with the max position at the end.
     3. Test an array with the max position at the front.
   * rotateLeft
     1. Test where the last position is “shifted.”
     2. Test where the first position is “shifted.”
   * countRuns
     1. Test an array that is made up all the same string and see if it returns 1.
     2. Test an array with every string different.
     3. Test an array with consecutive values.
   * flip
     1. Test an array with an odd number of values.
     2. Test an array with an even number of values.
   * differ
     1. Test two arrays that have nothing in common.
     2. Test two arrays that are identical.
     3. Test two arrays in which one is identical but is a few positions short.
     4. Test two arrays in which the discrepancy occurs within the string.
   * subsequence
     1. Test an array in which the a2 does not exist in a1 at all.
     2. Test an array in which the a2 does not exist in a1 completely.
     3. Test an array in a2 exists at the beginning of a1.
     4. Test an array in a2 exists in the middle of a1.
   * lookupAny
     1. Test an array that has no values in common between a1 and a2.
     2. Test an array that has one value in common between a1 and a2.
     3. Test an array that has more than one value in common between a1 and a2 and see if it returns the first position.
   * separate
     1. Test an array that is all larger than the separator.
     2. Test an array that is all smaller than the separator.
     3. Test an array that the separator is in the middle of the array.