## Recursive Binary Search

**The goal for this exercise** is to review both recursion, and binary search, by implementing a recursive binary search.

In the previous lesson, you created a function to search through an array, using a binary search. Copy that code, and modify is, so that it's recursive.

In order to do this, you'll create a function named FindIntegerBinary, that has the following prototype:

**public** bool FindIntegerBinaryRecursive(int target,   
 int [] array);

This method will call another, very similar method, which has the following first line:

**private** bool FindIntegerBinaryRecursive(int target,

int[] array,

int lowestIndex,

int highestIndex);

It is this second method that will end up calling itself repeatedly.

Both methods should be placed in the SearchingAndSorting class. Test your code by adding code to Main() as you need to.

**What you need to do for this exercise:**

1. Implement the FindIntegerBinaryRecursive method, within the SearchingAndSorting class.
2. The following is NOT required (which is why it’s crossed out).  
   I may want to add it back in a future quarter, though, which is why it’s still here. ~~Within the Test\_Find\_Integer\_Binary\_Measured class (which is located in the~~ **~~PCE\_StudentCode~~** ~~project, inside the~~ **~~PCE\_09.cs~~** ~~file), you should implement any tests that you feel are useful, inside the TestRecursiveBinarySearch method.~~ 
   1. ~~You may need to set another project as the start up project (such as the~~ **~~PCE\_StudentCode~~** ~~project), by right-clicking on that project, and selecting the “Set As Startup Project” menu option. Within the test runner project, you may (or may not) need to change the code in the Program.cs file, so that TestRecursiveBinarySearch is actually called.~~