Assignment #3 (10 points)

1. Using the binary search code given in class, write a BinarySearchEvaluator class, with the following specifications:
   - Member variables array and size, where array is an array of integers, and size is the size of the array
   - Member variable numCalls counts the number of times the binarySearch method is called
   - Member methods:
     - constructor: takes argument used to set the value of size, then creates an array to hold the specified number of int values; sets numCalls to 0; populates the array with values from 1 to size
     - binarySearch (from notes): finds the value specified in its first argument and increments numCalls each time it is called
     - reset method: resets numCalls to 0
     - main method to test the others; should run binarySearch algorithm several times, using randomly generated values to search for; reports value of numCalls after each search (and also reports whether or not search was successful)

2. Write a program that uses a recursive method to find the greatest common divisor of two numbers (a and b) using the following algorithm:

   Procedure gcd (inputs: a, b with a < b)
   
   if (a=0) then gcd(a,b) = b
   else gcd(a,b) = gcd(b mod a, a)

3. Although we discussed the linear search algorithm as an iterative problem, it could also be solved recursively. Write a class similar to your first one, but this time make the following changes:
   - Instead of a binarySearch method, create a linearSearch method that either:
     - returns the index of the target (if found)
     - returns -1 if we’ve reached the end of the array with no target found
     - or, if there’s more array to search and target isn’t found: call linearSearch again with a smaller version of the array