Assignment 1 – 1.1

* Page 26-27
* Write a program to solve the selection problem. Let k=N/2. Draw a table showing the running time of your program for various values of N (10, 100, 1000, 10000). Make sure you submit a working java program and the runtime in seconds to 5 decimal places for at least the 4 problem sizes above.

Assignment 1 – 1.5

* Write a recursive method that returns the number of 1's in the binary representation of N. Use the fact that this is equal to the number of 1's in the representation of N/2, plus 1, if N is odd. For example, starting at 5, 5 is odd, so ones++, 5/2 = 2, 2 is even, 2/2 = 1, 1 is odd, so ones++, so 5 should produce an answer of 2 (101 in binary). DO NOT use the built in method for converting an integer into a binary string. Make sure you are prompting the user for a number, then outputting how many 1's are in it's binary representation.

Assignment 1 – 1.15

* Define a Rectangle class that provides getLength and getWidth methods. Using the findMax routines in Figure 1.18, write a main that creates an array of Rectangle (5+) and finds the largest Rectangle, first on the basis of area, and then on the basis of perimeter. Note, you should create 2 Comparator classes, one for area, and one for perimeter, to use with the findMax method provided by the textbook. You shouldn't need to modify the findMax method.

Assignment 1 – Grading

* 1.1 - Program (2pts)  
  - Select middle largest number (if 10 numbers, get 5th largest)
* 1.1 - Runtimes (1pt)  
  - Add code to program for timing (only need to time solving, not filling the array with numbers in a random order), record runtimes of at least 4 different executions (10, 100, 1000, 10000)
* 1.5 - Program (3pts)  
  - Recursive method
* 1.15 - Program (4pts)  
  - Rectangle Class, 2 Comparators, findMax, 5+ Rectangle objects in an array