CSCE 121(505-508) **Homework 8**
Due 11:59 p.m., 4/8/14 [80 points]

This Homework assignment is meant to help you understand more about pointers.

All work should be turned into CSNet by the deadline. **In addition, please bring a hardcopy printout of your code to class the next day.** For programs, you need to turn in only the source code (not object or executable code). Your code will be tested using g++: you are welcome to develop in Visual Studio, but please make sure your code also runs in g++.

0) Create a text file, README, in which you:
   a. State the Aggie Honor statement, or else explain why you cannot do so.
   b. List any resources used, outside the textbook and discussions with the Instructor, TA, or Peer Teacher
   c. List any known problems with the assignments you are turning in. For example, if you know your code does not run correctly, state that. This does not need to be a long explanation.
   d. For places where you may have done some additional work, put in a brief summary detailing what you did.

1) [80 points] You are to write a program that:
   a. Prompts the user for a file name
   b. Reads in integers from that file
   c. Stores those numbers in a binary tree
   d. Outputs a sorted list of the numbers by outputting the binary tree

   There are better ways to sort numbers than this, but this will be a way of ensuring you can handle pointers correctly. Here are some details:
   - Binary trees will be discussed in more detail in lab. They are a very fundamental structure used in programming, and you can find many references to them. The basic idea is that you have a set of "nodes." Each node stores a value, and can have two children, one right and one left. Each child should be a node, itself. For this problem, a node should identify its children using pointers.
   - You should create a class for the node of a tree. You will want to have a "null" pointer to represent cases where there is no left/right child. Think about what data needs to be stored in this node class, as well as what functions (methods) need to be provided in it.
   - For a "sorted" tree, for any node, the values less than the value of that node should be in the left "branch", and the values greater than it in the "right" branch. Equal values could go on either side
   - You should read integers in the input file until the end of file is reached. Build a binary tree as you read numbers.
   - Once the file is finished, you should be able to output the tree by outputting the left branch, then the value, then the right branch. This should produce a list of all the integers, in order.
   - [10 points]. If an integer is repeated more than once, when you are outputting, rather than output that number multiple times, output the number, followed by an "x" and the number of times it was repeated.
     - For example, if the input file was 4 9 1 9 2 9 10 2, you would output: 1 2x2 4 9x3 10
     - Note: get the rest of the program working before trying to modify it to do this!

2) EXTRA CREDIT [50 points]. Write a "20 questions" style game. The way this works is that the user should gradually "teach" the computer to be able to guess items.
   a. The game is played where one player (the user) things of some thing. The other player (the computer) asks yes/no questions to try to determine what the user is thinking of. Eventually, that player will guess when it seems there is only one thing the person could be thinking of. You can get the computer to play this game by implementing a basic binary tree.
   b. Initially, the computer should guess "Is it a rock?". Assuming that the user was thinking of something different, the user will reply "No" and then be asked what the user was thinking of, and what a yes/no question is that would distinguish it from a rock. The next time the game is played, that question will be asked, and the computer will either guess rock or that item. As the game is played repeatedly, the computer will gradually learn to distinguish more and more.
   c. Here is an example of what a game might look like:

```
Type Q to quit.
Is it a rock? N
What were you thinking of? cat
```

```
Type a yes/no question that would let me distinguish
     rock from cat:
     Is it alive?
Is the answer for cat Yes or No? Y
OK, let's play again...
Is it alive? Y
Is it a cat? N
What were you thinking of? dog
Type a yes/no question that would let me distinguish
     cat from dog:
     Does it say meow?
Is the answer for dog Yes or No? N
OK, let's play again...
Is it alive? Y
Does it say meow? Y
Is it a cat? Y
I guessed it!
OK, let's play again...
Is it alive? Q
```