COMP241
Fall 2015
Programming Assignment 3
Assigned: 29 October, 2015
Deadline: 5 November, 2015 (early deadline: 3 November, 2015) at the beginning of class

Submission guidelines:

- Please post your solution on Moodle. It is due ***at the beginning of class.***
- +5 extra credit if you turn it in on November 3rd (also at the beginning of class)
- Use MS Visual Studio 2012 for the assignment. Submit your source code files (.cpp and .h files) separately for each problem.
- Include a comment with the course number, term, assignment, your name, and the date, at the top of each file.
- In addition to writing excellent code, you will be graded on the user experience you create. Any time there are user prompts and input/output, make sure your code is both informative and polite.
- Although I encourage collaboration if you want to work with a classmate, you must turn in your own work. Duplicate code and plagiarism will be considered a violation of the honor code and handled accordingly.

For the first two problems, you'll be writing recursive functions that operate on lists with an underlying linked structure. You'll need to have a working Unsorted List class for that, of course, so consider that a pre-requisite for the assignment. You can assume that all the lists are lists of integers.

If you don't have working List-linked-structure code handy, your best bet is probably to modify the List class from Programming Assignment #1. That one had an underlying array; shouldn't take much work to fix it up! Make sure you submit your Unsorted List class `.cpp/.h` files to make your solution work.

Problem #1
35 points

Write a recursive function that finds the node with the smallest element in a linked list.

Demonstrate that your function works with a mini test-suite in your driver: Repeatedly create a list, print it, find and print the smallest element. Create your list with elements that you create in the source code; don't prompt the user.

Problem #2
35 points

Write a recursive function that sorts the elements in an unsorted list. The original list should end up modified; don't create/return a whole new list.

Demonstrate that your function works with a mini test-suite in your driver: Repeatedly create a list, print it, sort and print again. Create your list with elements that you create in the source code; don't prompt the user.

Problem #3 (not a linked-list question!)
30 points

Write a recursive method in C++ that returns the number of 1s in the binary representation of its integer parameter, n. Example: if n = 23, then its binary representation is 10111 and your function would return 4.

Demonstrate that your function works with a mini test-suite in your driver: Repeatedly call your function and prints its argument and result, with various possible inputs.