Assignment #1 – Binary, Hex, and C/C++
Due: Friday, 10/05/12, 11:59pm

1. (12.5 pts) Convert each of the following decimal numbers into its equivalent binary and hexadecimal number: (You MUST show your work to get credit!!!):
   a. 7
   b. 40
   c. 78
   d. 132

2. (12.5 pts) Convert each of the following binary numbers into its equivalent hexadecimal and decimal number: (You MUST show your work to get credit!!!):
   a. 1001
   b. 1101
   c. 101011
   d. 1011011

3. (75 pts) For each programming assignment, you will carry out Polya’s 4 steps for solving problems:
   - Understanding the problem. (Recognizing what is asked.)
   - Devising a plan. (Responding to what is asked.)
   - Carrying out the plan. (Developing the result of the response.)
   - Looking back. (Checking. What does the result tell me?)

   (15 pts) You are provided a problem statement with each programming assignment, and you will be required to turn in a written document (as a pdf) addressing Polya’s steps to solving a problem with step 3 being the C code you write to carry out/implement your plan. With this said, your written document must include these three sections:

   **Understanding the Problem**
   In your own words, explain what YOU think the problem is asking you to do. In this section, document your uncertainties about the problem and anything else that you feel was unclear or vague. This is to ensure that YOUR understanding matches MY understanding of the problem.

   **Devising a Plan/Design**
   At a minimum, provide an algorithm/pseudo code you designed to help solve the problem. In addition, include pictures/flow charts you used to help you devise your plan, as well as any other design decisions you made such as how to manage your time, how to decompose the problem, where to start first, etc. You can scan any handwritten work and attach it to the document as needed.

   **Looking Back/Self-Reflection**
   Report any checking/self-reflection you did while solving the problem. For instance, how did you make sense of the output from the implementation? This
includes things such as using a calculator to make sure the output is correct, testing to make sure your code executes correctly and behaves the way you expect under specific circumstances, using external sources of information such as the internet to make sense of the results, etc. Also, include a statement about what you learned from the assignment.

(60 pts) **Problem Statement:** Write a C and C++ program that assigns the largest and smallest short, int, and long, both signed and unsigned, from standard headers/libraries (limits.h for C and climits for C++) to variables of the correct data type. List of largest/smallest macros: [http://www.cplusplus.com/reference/clibrary/climits/](http://www.cplusplus.com/reference/clibrary/climits/)

Print the values in decimal (base 10), hexadecimal (base 16), and octal (base 8) using dec, hex, and oct with cout in C++ and %d, %x, and %o with printf() in C.

Now, print the decimal (base 10) values of adding one to each variable containing the maximum numbers and subtracting one from the variables containing the minimum values (even zero for unsigned).

Lastly, directly compute the largest and smallest short, int, and long, both signed and unsigned, using pow(), which is in the math.h for C and cmath for C++. You only need to print the decimal values for these numbers.

(5 pts of each program, which is 10 points of your 60 total programming points)
In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments!

```c
/******************************************
** Program: largest_int.c
** Author: Your Name
** Date: 10/03/2012
** Description:
** Input:
** Output:
******************************************/
```

5. Electronically submit your written work and C program by the assignment due date, using TEACH. **NOTE:** The easiest way to upload your program from ENGR to TEACH is to map a network drive to your home directory on ENGR. See instructions: [http://engineering.oregonstate.edu/computing/personal/110](http://engineering.oregonstate.edu/computing/personal/110)

***For Mac users, here are two links suggested by students in the class:

[http://engineering.oregonstate.edu/computing/personal/258](http://engineering.oregonstate.edu/computing/personal/258)

Programs needed are osx fuse and Macfusion

Graphical scp/sftp client for transferring files between ENGR and your computer: [http://cyberduck.ch/](http://cyberduck.ch/)