Assignment #1 – Binary, Hex, and C Programming
Due: Thursday, 4/12/12, 11:59pm

1. (16 pts) Convert each of the following decimal numbers into its equivalent binary and hexadecimal number: (You MUST show your work to get credit!!!):
   a. 6
   b. 44
   c. 72
   d. 131

2. (16 pts) Convert each of the following binary numbers into its equivalent hexadecimal and decimal number: (You MUST show your work to get credit!!!):
   a. 100
   b. 1011
   c. 101010
   d. 1001110

3. (8 pts) In your own words, write an algorithm for making an Easter Dinner on Sunday. Assume that you have at least 30 family members and friends coming over to your house, so you’ll want to produce enough food that will feed everyone! Yummy, yum!!!

4. (60 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.

(40 pts) **Problem Statement:** Write a program to determine the ranges of **short**, **int**, and **long**, both **signed** and **unsigned**, by printing appropriate values from standard headers/libraries (see page 257) and by direct computation. Print the integer ranges in decimal (base 10), hexadecimal (base 16), and octal (base 8) (see page 244). Lastly, assign variables the max value for each type and add one to each variable. Print the resulting number for each type in decimal, hexadecimal, and octal to the console.

(5 pts) In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments!

```
/*******************************************/
** Program: largest_int.c**
** Author: Your Name**
** Date: 4/10/2012**
** 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

[http://cyberduck.ch/](http://cyberduck.ch/)