Lab 6: Decimal Converter

Worth 50 points - 10 extra credit
All source due by Sunday 11:55pm, November 15th, 2015

Lab Objective:

Now that we are moving into programming in the lab, it is important for you to focus on good programming practices. When programming in assembly this is especially true as assembly is not a pretty or easily readable language. Being clear with your register usage is very important, also, having a clear plan for your program is essential, thus for all programs in this class you will be required to create flowcharts. Once a flowchart is created the problem has been solved, next you just map your solution to assembly code.

Part A:

Please review the Program Flow slides presented in class. The purpose of that lecture was to go over some of the basics of what a programmer does and the usage of flowcharting to solve a simple problem. Once the problem is solved it is easy to then implement it in whatever programming language you are required to use.

Part B:

You know how to get character input from the user. Extend this to write a program to input and output integer values. Here are the specifications:

- Print out a greeting message.
- Ask for a decimal number followed by a carriage return.
- Expect the user to only input valid digit characters (0-9) and possibly a leading negative sign (-).
- Convert the string of digits entered into a two's complement number (you will have to do something special for the negative sign).
- Print the number out in binary.
- Repeat until the user wants to quit.

Pretty straight forward but this will require the use of loops and procedure calls.

Lab Requirements:

You must do the following for this lab:

- Create a flow chart for your program first, submit this to the TA/tutor when you demonstrate your program, here is an example of one **Lab6_Flowchart.pdf**.
- Create the following procedures:
  1. A get decimal integer procedure, it will read a character string and convert to an integer.
  2. A print binary procedure, it will just print out the bits that are stored in a register as a string of 16 0's and 1's.

- Print out appropriate messages and prompts and have some way to quite your program when finished.

See the example below for how this might work. Give the program your own personal look and feel.

Example of possible output:

Your output does not have to look exactly like this, make it your own program. You just need to make it useable so that anyone sitting down would know how to use it. You can assume that only valid input is accepted.

```
Welcome to the conversion program
Enter a decimal number or X to quit:
>12
Thanks, here it is in binary
0000000000001100
Enter a decimal number or X to quit:
>-3
Thanks, here it is in binary
1111111111111101
Enter a decimal number or X to quit:
>X
Bye. Have a great day.
----- Halting the processor -----
```

Extra Credit Version

```
Welcome to the advanced conversion program

Enter a decimal number or X to quit:

>12
Thanks, here it is in dec, bin, hex, and oct
12
0000000000001100
000C
000014
>-3
Thanks, here it is in dec, bin, hex, and oct
-3
1111111111111101
FFFD
177775
>X
Bye. Have a great day.
----- Halting the processor -----
```

Extra Credit:

Once you complete ALL the above requirements and you still want to have even more fun, try to do some or all of the following additions:

- Check for invalid inputs, i.e. make sure they type in only '0' to '9' and '-', and only in the correct locations.
- Create additional printing procedures that print out your number in HEX, OCT or DEC (you need to actually do the conversion back, not just store the string entered).

See the grading template for information on the points possible for the extra credit.

Collaboration: ***You are allowed to discuss this lab with other students on this lab only from a high level, such as discussing your flowcharts, NOT by working on code together.***

Files to Submit:

- **Lab6.asm**
- **Lab6_report.pdf – needs to discuss the algorithm(s) you implemented and that you drew in the flow chart.**
- **Flowchart to TA/Tutor when demonstrating**

Check-off: In must demonstrate this lab to the TA/tutor, either before submission or after submission. To be considered on time you must have submitted everything by the due date.

Grading template:
See the template file.