



**Lab 4: Decimal Converter**  
**Worth 70 points (65 lab + 5 report)**

**Lab Objectives:**

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 1:**

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 2:**

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 Lab3\_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 quit your program when finished.

See an Example below of how this might work. Give the program your own personal look and feel but try to keep it clear.

## 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. Inputs to the program are in red and bolded.

```
Welcome to the conversion program
Enter a decimal number or X to quit:
>12
Thanks, here it is in binary
00000000000001100
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 -----
```

## Lab Write-up Requirements

Discuss algorithm(s) you implemented and issues doing so. Discuss any particular constructs in assembly you made that made the problem easier to solve. Be sure to include how you used your flowchart in the actual coding of your program.

To alleviate file format issues we want lab reports in plain text. Feel free to use a word processor if you like to type it up but please submit a plain text file.

**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 on Canvas

- Lab3.asm
- Lab3\_report.txt
- Flowchart to TA/Tutor when demonstrating

## Check-off

You should demonstrate your lab to the TA/Tutor when finished to get it checked off and show them your flowchart.

## Point Breakdown

- 5 pts: Print greeting message and quit when prompted
- 20 pts: Converts character decimal string into a 2SC number
- 20 pts: Print as a binary string
- 10 pts: Work with negative input
- 5 pts: Good flow chart
- 5 pts: Have good comments in the code
- 5 pts: Writeup