CS270 Programming Assignment 3
"LC3 Assembly Programming"

Program due Monday, October 17, 2011 (via Checkin by 4:59pm)
Revised 10/06/2011

Goals

In this assignment, you will write an LC-3 assembly program to compute a sequence of numbers according to a specified recurrence relation. The goals are:

- Learn how to write a non-trivial LC-3 program, and
- Expand your understanding of the LC-3 architecture and tools.

The Assignment

You will need to use the LC3 tool-chain to complete the assignment. Download the Unix version of the LC-3 simulator from http://www.cs.colostate.edu/~cs270/links.php, extract the zip file and build it:

```
$ unzip lc3tools_v12.zip
$ cd lc3tools
$ ./configure
$ make
```

Take a look at the descriptions provided in the Wikipedia pages for Fibonacci and Lucas numbers as examples of sequences:
http://en.wikipedia.org/wiki/Fibonacci_number
http://en.wikipedia.org/wiki/Lucas_number

For this assignment, you will write an assembly program that takes an integer N (0 <= N < 22) and extends a given 3-integer sequence (called a prefix) by N additional values. The prefix (and all subsequent values) are to be stored contiguously (i.e., in a block of words) in memory starting at an address labeled RESULT. You will want to play with and try out different prefixes when you simulate the program, so you should allocate an array (block of words) of size N+3. The rule to extend a prefix is:

$$F_n = F_{n-1} - F_{n-2} + F_{n-3}$$

- For example, suppose you choose the prefix (1, 2, 3), and N = 3, then the value at address RESULT is 1, the value at address RESULT + 1 is 2, and the value at RESULT + 2 is 3. Then you need to compute 3 – 2 + 1 = 2 and store this at address RESULT + 3, compute 2 – 3 + 2 = 1 and store this at address RESULT + 4, and compute 1 – 2 + 3 = 2 and store this at address RESULT + 5. The resulting array has six elements total and begins at address RESULT.

Rules:

- The resulting numbers must be stored into memory immediately after the program instructions and following any input variables (and additional variables you may use).
- The storage for the numbers (including the first three -- so a total of N+3 words which is no more than 24) must be allocated at assembly time, therefore, your argument to .BLKW will be N+3. You

might also use a series of .FILL statements.
    - You must label the base address of this memory using the label: **RESULT**
- The program must start at address location **0x3000**.
- The input number must be specified as the variable **N**.
    - If N > 21, set N to 21.
- Your code must be well commented (i.e., in general, every line has a comment)
- Before submission, set your prefix to (5, 3, 2)

Failure to follow these rules may result in loss of points. As a starting point, we provided you with a sample assembly file here:
http://www.cs.colostate.edu/~cs270/Assignments/PA3/example.asm

The example provided by Chris Wilcox in lecture 10/06/11 can be found here:
http://www.cs.colostate.edu/~cs270/Assignments/PA3/loop.asm

Submission Instructions

Create a directory called PA3 to store the assembly file, and name your file pa3.asm. Submit a tar.gz file generated from the following command (assuming you are running this command from inside PA3 directory):

```
$ cd ..; tar –czvf PA3.tar.gz PA3
```

Do not submit the assignment with lc3tools inside PA3.

Grading Criteria

Points will be awarded as follows:

- Detecting if N is greater than 21 and setting N to 21: 15 points
- Completing rest of the functionality: 75 points
- Style: 5 points
    - Style includes well-commented code and using fewer instructions.
- Following submission instructions: 5 points

Late Policy

This assignment will be accepted up to 24 hours past the due date (only one day late period for this assignment) with a 10% deduction. This assignment will not be accepted past this period. If you were unable to submit via Checkin for any reason, contact us via email and we may be able to help.