LC3 Program 1:

Math Functions and Overflow Checking

Due: Monday, March 31

Prepared for: NCSU Engineering Programs at UNCA
Prepared by: Dakota Lazenby, TA

January 11, 2014
Project name: LC3 Programming Assignment 1
Math Function & Overflow Checking

Objective
The objective of this assignment is to test your ability to break a larger task or problem into programmable, algorithmic chunks; then to program the LC3 (in machine language or assembly language) to implement your design. In this project, you will be tested on recently covered material including loops, conditionals, and the basic structure of a low-level (machine or assembly language) program.

Task
Your task in this assignment will be to use a text editor and/or CodeLup.jar to write a program that performs a pre-determined math function and then performs error checking on certain portions of your code. You will use the LC3 PennSim simulator to execute your program.

Overview
Your program should calculate the answer to this math function:

This function should be implemented in your LC3 program using loops and conditionals to handle the multiplication, division, and error checking.

\[ (a \times X - \frac{Y}{b}) + Z = \text{answer} \]

Breakdown
The backbone design of this program has been left up to you. Because of the relatively narrow scope of this project, giving a detailed layout of how to write this program would give away the answer to the problem, and would prevent you all from having unique solutions. However, some specifications and hints have been provided below.

Specifications
Your program should start at x3000.
Variables X, Y and Z will be provided for you in memory at locations x3070, x3071, and x3072 respectively.
Your answer should be stored in memory location x3075.
You should store the values for a & b in memory at locations x3073, and x3074 respectively.
You may assume that all parameters: X, Y, Z, a, and b are either zero or positive, not negative.
You may assume that the parameters are such that neither the multiplication (a*X) nor the division (Y/b) causes overflow.
You may also assume that the division will be “nice,” meaning that there will be no division by zero.
You may neglect the remainder when performing division.
Because all parameter are positive, you will only need to check for overflow after the addition shown in the function.

Memory locations and their designated values.
  x3000  --> your program begins here
  ...
  ...
  X30??  --> your program ends here
  ...
  ...
  x3070  --> X (entered in memory before your program runs)
  x3071  --> Y (entered in memory before your program runs)
  x3072  --> Z (entered in memory before your program runs)
  x3073  --> a  (you store this value)
  x3074  --> b (you store this value)
  x3075  --> store your answer here or leave it as x0BAD if O.F. occurs (x0BAD will be stored in this memory location before your program runs.)

Lastly, the error checking case. When error checking (checking overflow) remember that a difference in MSBs of the addends will NEVER result in overflow, and overflow occurs when the MSBs of the two added numbers are the SAME but the MSB of the result is DIFFERENT.

When your program determines that overflow has occurred, it should leave the previously stored hexadecimal value of x0BAD in memory location x3075. If no overflow occurs, then your program should store the answer to the function in memory location x3075.

The value for a that you should use is decimal 10.

The value for b that you should use is decimal 6.

Please direct any questions, clarifications or concerns to your Instructor or your TA / Facilitators.