ELEC 2220 Computer Systems
Homework #7
Due: Monday, June 8

For this and the remaining assignments, "Target" memory should be configured to use the on-chip flash memory of the microcontroller (beginning at address 0x08000000) for program code, and RAM (beginning at 0x20000000) for data. Running the program on the board for this assignment is recommended, but using the simulator is acceptable.

Write and test a short assembly language program to perform the following computation.

$$Leia = (Han - Luke) + (Yoda - 37)$$

A. Create a Data Area and allocate space for one 32-bit word for each of the four variables *Han*, *Luke*, *Yoda*, and *Leia* (in that order). These should be initialized to the following values:
   Han = 250, *Luke* = -150, *Yoda* = 800, *Leia* = -1

B. The program is to set up and use a separate memory pointer register for each variable, to read/write the variables.

C. In the memory window of the debugger, you would likely see random numbers on the board, which reflects the "random state" of RAM when the actual microcontroller powers on. If using the simulator, you will likely see all 0s. You can create the test values for Part a for your debug session as follows.

   With a debug initialization file:
   1. On the Debug tab of "Options for Target", select an "Initialization File", which contains commands to be performed by the debugger with it starts.
   2. My debug initialization file ("debug.ini") contains the following:
      PC=0x08000000   //Starting PC Counter Address
      xPSR=0x01000000   //Set T bit of Processor Status Register to Thumb Mode
      E INT 0x20000000 = 250,-150,800,-1   //Initial memory data values

   In the last line, "E" tells the debugger to "Enter" values into memory, as "INT" types (32-bit integers), beginning at address 0x20000000, with a list of four values to be inserted into successive words of memory.

   3. In lieu of a debug initialization file, the above commands can be entered in the command window of the debugger at any time during the debug session.

   Interactively, in the Memory 1 window, you can change memory contents as follows:
   a. Select the data memory starting address of 0x20000000 in the Memory 1 window.
   b. Right click in the Memory 1 window, and change its format to display memory contents as 32-bit signed decimal numbers, by selecting "Decimal" and also "Signed > Int".
   c. Right click on the first data value in memory to be changed (at address 0x20000000) and select "Modify Memory at 0x20000000". In the dialog box that appears, enter the list of values you want to place in memory words, using the signed decimal format you selected in Step 2, and beginning at that address. (Ex. 250, -150, 800, -1, etc.)
   d. Alternatively, you can double click on any memory value that you want to enter, which highlights it, and then change it to the desired value.

D. Open a Watch window and display the values of the four variables *Han*, *Luke*, *Yoda*, and *Leia*.

E. Execute the program one step at a time, studying the values of the registers, memory locations, and watch variables.

F. After executing the last instruction, do a screen capture of the debugger window. In the debugger's "Memory" window, highlight or circle the final value of variable *Leia*.)

G. ***Hand-calculate*** the result and then write your answer on the debug window printout, and indicate whether it matches the result produced by the program.

H. Modify the program to use pointer plus offset addressing. Create a single memory pointer in a register, and then use that value plus appropriate offsets for each variable (without changing the address in the pointer register), and repeat the execution and repeat steps C-F.

I. Submit your two assembly language source programs and the captured, annotated debugger windows. (If you can copy these into a single file, that would help grading.)