Objectives:
- Y86 assembly language

REMINDERS and GRADING CRITERIA:
- Read these sections from lab1... they haven’t changed.

LAB DESCRIPTION

Mandatory filename → lab4.ys

PROBLEM:
Write an assembly language program using Y86 assembly to add, subtract and multiply two values in an array.
The program will need to add, subtract and multiply the first two values in the array, then do the same for the second and third values in the array, then do the same for the third and 4th values in the array, etc.

You are required to define the following at the top of your program:
(1) the number of values in your array
(2) the actual array values

NumEl: .long 0x4  # number of array elements/values
Array: .long 0x5
       .long 0xc
       .long 0x3
       .long 0x7
Done: .long 0xFFFFFFFF  # optional

INPUT:
- The number of elements input is a non-zero positive value.
- There are no data errors that need to be checked as all the data will be assumed correct.
- Your program should be able to handle an unknown number of array values and still work. That is, you cannot assume that you know how many input values there will be; however, the exact number of input values will be designated correctly as shown above.
- Instead of counting through the number of elements given, you can check for the end of the input data by using -1 as the value to designate the end of the input; or check for the “Done” label location.
- You can use any name you like for the labels instead of what is given, but they should be descriptive.

HINT: Since Y86 does not have a multiplication statement, you want to remember that 2*4 is the same as 2+2+2+2.

CONSTRAINTS:
- You can assume that the result of any input value will not overflow.
- You are NOT ALLOWED to use the “call” instruction in this program. This is NOT a stack based programming problem.
OUTPUT:

- You need an output area of memory in your program which is required to be at the bottom of your program with a descriptive label associated with it.
- The output is the addition, subtraction and multiplication (in that order) of each pair of numbers in the array (in order) as specified in the problem section above.

For the data given in the problem section above, the output (designated below in decimal) should be:

- 5+12
- 5-12
- 5*12
- 12+3
- 12-3
- 12*3
- 3+7
- 3-7
- 3*7

LAB SUBMISSION

Read this section from lab1... it hasn't changed except for:

The labname for this lab is: lab4
Be sure to submit the following files: lab4.xls and a README file