CS 271 Computer Architecture and Assembly Language
Programming Assignment #3
Due Sunday, Oct. 28th (11:59 PM)
Submit at http://engr.oregonstate.edu/teach before midnight. Submit a backup copy to Blackboard.

Objectives: more practice with

1. Implementing data validation
2. Implementing an accumulator
3. Integer arithmetic
4. Defining variables (integer and string)
5. Using library procedures for I/O
6. Implementing control structures (decision, loop, procedure)

Description:

Write and test a MASM program to perform the following tasks:

1. Display the program title and programmer's name.
2. Get the user's name, and greet the user.
3. Display instructions for the user.
4. Repeatedly prompt the user to enter a number. Validate the user input to be less than or equal to 100. Count and accumulate the valid user numbers until a negative number is entered. (The negative number is discarded.)
5. Calculate the (rounded integer) average of the non-negative numbers.
6. Display:
    i. the number of non-negative numbers entered (Note: if no non-negative numbers were entered, display a special message and skip to *iv.*)
    ii. the sum of non-negative numbers entered
    iii. the average rounded to the nearest integer
    iv. a parting message (with the user's name)

Requirements:

1. The *main* procedure must be modularized into commented logical sections (procedures are not required this time)
2. The program must be fully documented. This includes a complete header block for identification, description, etc., and a comment outline to explain each section of code.
3. The upper limit should be defined as a constant.
4. The usual requirements regarding documentation, readability, user-friendliness, etc., apply.
5. Submit your text code file (*.asm*) by the due date to both TEACH and Blackboard.

Notes:

1. There are no new concepts in this programming assignment. It is given for extra practice, to keep MASM fresh in your mind while we study internal/external data representation.
2. This is an integer program. Even though it would make more sense to use floating-point computations, you are required to do this one with integers.

Extra-credit options (original definition must be fulfilled):

1. Number the lines during user input.
2. Calculate and display the average as a floating-point number, rounded to the nearest .001.
3. Do something astoundingly creative.