## Introduction

The purpose of this assignment is to reinforce concepts around looping, data validation, and control structures, to provide practice with signed arithmetic operations, and to bring the status register flags into focus (CLO 3, 4).

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

## What you must do

#### **Program Description**

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

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.
   1. Validate the user input to be in [-200, -100] or [-50, -1] (inclusive).
   2. Notify the user of any invalid negative numbers (negative, but not in the ranges specified)
   3. Count and accumulate the valid user numbers until a non-negative number is entered. Detect this using the SIGN flag.  
      (The non-negative number and any numbers not in the specified ranges are discarded.)
5. Calculate the (rounded integer) average of the valid numbers and store in a variable.
6. Display:
   1. the count of validated numbers entered  
      NOTE: if no valid numbers were entered, display a special message and skip to (f)
   2. the sum of valid numbers
   3. the maximum (closest to 0) valid user value entered
   4. the minimum (farthest from 0) valid user value entered
   5. the average, rounded to the nearest integer
      1. -20.01 rounds to -20
      2. -20.5 rounds to -20
      3. -20.51 rounds to -21
      4. -20.99 rounds to -21
   6. a parting message (with the user’s name)

#### **Program Requirements**

1. The program **must** be fully documented and laid out according to the [CS271 Style Guide](https://canvas.oregonstate.edu/courses/1810861/files/85561736/download?wrap=1)[Download CS271 Style Guide](https://canvas.oregonstate.edu/courses/1810861/files/85561736/download?download_frd=1). This includes a complete header block for identification, description, etc., section comments for logical sections of code (modules) and block comments for code blocks.
2. The main procedure must be modularized into commented logical sections (procedures are not required this time)
3. The four value limits must be defined as constants.
4. The user input loop should terminate depending on the value of the SIGN flag.
5. The Min, Max, Count, Sum, and Average must all be stored in named variables as they are calculated.
6. Min, Max, Count, and Sum should be re-calculated every time the user enters a valid number. Average may be calculated after the last valid number.

#### **Notes**

1. This is an integer program. Even though it may make more sense to use floating-point computations, you are required to use **signed integer computations**.
2. You do not need arrays to complete this program - and you do not yet have the tools to use arrays. Trying to use arrays will just make it harder!
3. Check the [Course Syllabus](https://canvas.oregonstate.edu/courses/1810861/files/85708476/download?wrap=1)[Download Course Syllabus](https://canvas.oregonstate.edu/courses/1810861/files/85708476/download?download_frd=1)for late submission guidelines.
4. Find the assembly language instruction syntax and help in the [CS271 Instructions Guide](https://canvas.oregonstate.edu/courses/1810861/files/85562428/download?wrap=1)[Download CS271 Instructions Guide](https://canvas.oregonstate.edu/courses/1810861/files/85562428/download?download_frd=1).
5. To create, assemble, run,  and modify your program, follow the instructions on the course [Syllabus Page](https://canvas.oregonstate.edu/courses/1810861/assignments/syllabus)’s "Tools" tab.

## Resources

Additional resources for this assignment

* [Project Shell with Template.asm](https://canvas.oregonstate.edu/courses/1810861/files/85561500/download?wrap=1)[Download Project Shell with Template.asm](https://canvas.oregonstate.edu/courses/1810861/files/85561500/download?download_frd=1)
* [CS271 Style Guide](https://canvas.oregonstate.edu/courses/1810861/files/85561736/download?wrap=1)[Download CS271 Style Guide](https://canvas.oregonstate.edu/courses/1810861/files/85561736/download?download_frd=1)
* [CS271 Instructions Reference](https://canvas.oregonstate.edu/courses/1810861/files/85562428/download?wrap=1)[Download CS271 Instructions Reference](https://canvas.oregonstate.edu/courses/1810861/files/85562428/download?download_frd=1)
* [CS271 Irvine Procedure Reference](https://canvas.oregonstate.edu/courses/1810861/files/87258362/download?wrap=1)[Download CS271 Irvine Procedure Reference](https://canvas.oregonstate.edu/courses/1810861/files/87258362/download?download_frd=1)

## What to turn in

Turn in a single .asm file (the actual Assembly Language Program file, not the Visual Studio solution file).  File must be named "Proj3\_ONID.asm" where ONID is your ONID username. Failure to name files according to this convention may result in reduced scores (or ungraded work). When you resubmit a file in Canvas, Canvas can attach a suffix to the file, e.g., the file name may become Proj3\_ONID-1.asm. Don't worry about this name change as no points will be deducted because of this.

## Example Execution

User input in **boldface italics**.

Welcome to the Integer Accumulator by General Kenobi  
What is your name? **Grievous**  
Hello there, Grievous  
  
  
Please enter numbers in [-200, -100] or [-50, -1].  
Enter a non-negative number when you are finished to see results.  
Enter number: **-15**  
Enter number: **-550**   
Number Invalid!  
Enter number: **-36**  
Enter number: **-110**  
Enter number: **0**  
You entered 3 valid numbers.  
The maximum valid number is -15  
The minimum valid number is -110  
The sum of your valid numbers is -161  
The rounded average is -54  
  
We have to stop meeting like this. Farewell, Grievous

## Extra Credit (Original Project Definition must be Fulfilled)

To receive points for any extra credit options, you **must** add one print statement to your program output **per extra credit** which describes the extra credit you chose to work on. You ***will not receive extra credit points*** unless you do this. The statement must be formatted as follows...

--Program Intro--  
\*\*EC: DESCRIPTION  
  
--Program prompts, etc—

#### **Extra Credit Options**

1. Number the lines during user input. Increment the line number only for valid number entries. (1pt)
2. Calculate and display the average as a decimal-point number , rounded to the nearest .01. This calculation would need to be done in addition to the normal program functionality, with the result printing after the rounded integer average. Do not use the Floating Point Unit. (2pt)