Introduction

The purpose of this assignment is to acquaint you with elementary MASM programming and integer arithmetic operations (CLO 3, 4).

1. Introduction to MASM assembly language
2. Defining variables (integer and string)
3. Using library procedures for I/O
4. Integer arithmetic

What you must do

**Program Description**

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

1. Display your name and program title on the output screen.
2. Display instructions for the user.
3. Prompt the user to enter three numbers (A, B, C) in strictly descending order.
4. Calculate and display the sum and differences: (A+B, A-B, A+C, A-C, B+C, B-C, A+B+C).
5. Display a closing message.

**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., and a comment outline to explain each section of code.
2. The main procedure **must** be divided into logical sections:
   1. introduction
   2. get the data
   3. calculate the required values
   4. display the results
   5. say goodbye
3. The results of calculations **must** be stored in named variables before being displayed.

**Notes**

1. You are not required to handle negative input or negative results. We will not test input that would generate a negative output or overflow.
2. Check the Course Syllabus for late submission guidelines.
3. 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).
4. 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

* [Program Solution Shell with Template.asm](https://canvas.oregonstate.edu/courses/1810861/files/85561500/download?wrap=1)[Download Program Solution 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 Procedures Reference](https://canvas.oregonstate.edu/courses/1810861/files/87258362/download?wrap=1)[Download CS271 Irvine Procedures 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).  Files should be named "Proj1\_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 Proj1\_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***.

         Elementary Arithmetic     by Wile E. Coyote  
Enter 3 numbers A > B > C, and I'll show you the sums and differences.  
First number: ***20***  
Second number: ***10***   
Third number: ***5***  
  
20 + 10 = 30  
20 - 10 = 10  
20 + 5 = 25  
20 - 5 = 15  
10 + 5 = 15  
10 - 5 = 5  
20 + 10 + 5 = 35  
  
Thanks for using Elementary Arithmetic!  Goodbye!

Extra Credit Options (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—

For example, for extra credit option #2, program execution would look like this:

        Elementary Arithmetic     by Wile E. Coyote  
\*\*EC: Program verifies the numbers are in descending order.  
  
Enter 3 numbers A > B > C, and I'll show you the sums and differences.  
First number: ***20***  
Second number: ***25***  
ERROR: The numbers are not in descending order!  
  
Impressed?  Bye!

**Extra Credit Options**

1. Repeat until the user chooses to quit. (1pt)
2. Check if numbers are not in strictly descending order. (1pt)
3. Handle negative results and computes B-A, C-A, C-B, C-B-A. (1pt)
4. Calculate and display the quotients A/B, A/C, B/C, printing the quotient *and* remainder (see **DIV** instruction). (2pt)