

CS 271 Computer Architecture and Assembly Language

Programming Assignment #1

Objectives:

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

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 two numbers.
4. Calculate the sum, difference, product, (integer) quotient and remainder of the numbers.
5. Display a terminating message.

Requirements:

1. The main procedure must be divided into sections:
 - introduction
 - get the data
 - calculate the required values
 - display the results
 - say goodbye
2. The results of calculations must be stored in named variables before being displayed.
3. 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.
4. Turn in your submission to Canvas by the due date.

What to turn in:

1. Your source code files (*.asm*) that can be compiled by Visual Studio.
2. A video of a quick overview of your code and a quick demonstration of your program by compiling and running through it.
3. Do NOT put them into a zip file. Please leave them out separately.

Notes:

1. Read the article “Getting Started with MASM and Visual Studio 2019” on the navigation.
2. You are not required to handle negative input or negative results.
3. Find the **assembly language instruction syntax** in the textbook.
4. Find help on using **Irvine library procedures** in the textbook.

Example execution (user input is in *italics*):

Elementary Arithmetic
by Wile E. Coyote

Enter 2 numbers, and I'll show you the sum, difference,
product, quotient, and remainder.

First number: **37**

Second number: **5**

$37 + 5 = 42$

$37 - 5 = 32$

$37 \times 5 = 185$

$37 \div 5 = 7$ remainder 2

Impressed? Bye!

Optional challenges:

1. Repeat until the user chooses to quit.
2. Validate the second number to be less than the first.
3. Calculate and display the quotient as a floating-point number, rounded to the nearest .001.