Fall 2024 COMP 3350 Project #2
Possible points: 100 Due: Sep 27, 2024 11:59pm CST

Goals:

- Get you familiar with Microsoft Macro Assembler (MASM) on Visual Studio.
- Setting up an assembly program.
- Defining and accessing Arrays.
- Dealing with Registers and instructions.
- Debugging and running your assembly code.

Requirements:

- Read the design section and write a program. Submit source file (.asm) to Canvas.
- Always test your program before submission. You will lose at least 30 points if your code failed to build or run.
- Please start early. ZERO point for late submission. After the 11:59pm on the due day, you cannot submit your assignment anymore.
- You may hear "project hints/templates" from friends who took this class before. However, starting this semester, several topics have been updated, and the projects have been redesigned accordingly. Therefore, if you have solutions from your friends, do not copy and paste their code for the following reasons:

1. Their solutions are outdated and do not apply to the redesigned projects.
2. Without explicit permission from Li, you are not allowed to use them for any purpose.

If the TA or Instructor identifies such cases, your project will receive a ZERO and may be reported to the department or university at the Instructor's discretion.

Deliverables:

- Save your source of assembly program as a .asm document.
- Name document as a “Firstname_Lastname_Project2.asm”.
- Submit your “Firstname_Lastname_Project2.asm” through the Canvas system. You do not need to submit hard copies.

Rebuttal period:

- You will be given a period of 2 business days to read and respond to the comments and grades of your homework or project assignment. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.

Design:

The objective of this assignment is to create a program that will read a value from an array, add another value to this, and save the sum of those two values into a specific register. Please note that your program doesn't have to print out any results since TA will examine the values in your register.
(20 points) Create a BYTE array with the label 'input'. 'input' should have eight elements. You should place values 1,2,3,4,5,6,7,8 in each of the elements of this array.
(20 points) Create a BYTE variable with the label 'shift'. 'shift' should hold a single value 2.
(20 points) Set the values of the EAX, EBX, ECX, and EDX to 0. You then sum the value of this variable with each of the individual values in the array 'input'. The program should read each of the values from the array 'input' one at a time and add the value 'shift' to it.
The sum should be stored in the "correct" register:
(5 points) The first sums should be in high position of the AX register.
(5 points) The second sums should be in low position of the AX register.
(5 points) The third sums should be in high position of BX register.
(5 points) The fourth sums should be in low position of BX register.
(5 points) The fifth sums should be in high position of CX register.
(5 points) The sixth sums should be in low position of CX register.
(5 points) The seventh sums should be in high position of the DX register.
(5 points) The eighth sums should be in low position of the DX register.