COMP 3350 Project #3

Possible points: 100 Due: October 9, 2020 **11:59pm CST**

Goals:

- Defining and accessing Arrays.
- Dealing with Registers and instructions.
- Defining with Loops
- Debugging and running your assembly code.

Requirements:

- Read the design section and write a program. Submit source file (.asm) to Canvas.
- Show your work to get full credit. **ZERO** point without steps for a result.
- Please start early. ZERO point for late submission. After the **11:59pm** on the due day, you can't submit your assignment anymore.
- Check deliverables section below. ZERO point for hand-written or scanned homework.
- See my last announcement if you do not like to use visual studio. I provide an free and lightweight alternative.

Deliverables:

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

Rebuttal period:

- You will be given a period of 3 **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, and then place this value in another array with the location shifted by a certain amount. The array may be of any length from 2 to 100. Your program must be flexible enough to produce the correct solution regardless of the array size. You have to provide documentation for your program in the form of comments.

(10 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.

(10 Points) Create a BYTE array with the label ‘output’. This array should be the same length as ‘input’.

(10 Points) Create a DWORD variable with the label ‘shift’. ‘shift’ should hold a single value. The value of ‘shift’ must be less than the length of ‘input’.

(50 Points) The program should then read each of the values from the array ‘input’ and place the values into the ‘output’ array but the location should be shifted by the amount in the ‘shift’ variable. If the shift would cause a value to be outside of the bounds of ‘output’, then the values should “wrap around” to the front of ‘output’.

Example:

My ‘input’ array is 5, 0A, 2, 6, 0C, 9, 4

‘shift’ is 3

The proper solution for ‘output’ is 0C, 9, 4, 5, 0A, 2, 6

As you can see the value ‘5’ is the 1st value in the ‘input’ array. The value ‘5’ then shifts 3 to the 4th value in the ‘output’ array. Also, note that the value ‘0C’ is the 3rd value in the ‘input’ array. After a shift of 3, this would take the value ‘0C’ out of bounds for the ‘output’ array (it is the same length as the ‘input’ array). The value ‘0C’ must “wrap around” to the front of the ‘output’ array. This also holds true for ‘9' and ‘4’.

Remember that your program must be flexible enough to handle an array of any length.

Just because you test it with an array of length 6 does not mean that I will test it with an array of length 6. I could test with an array of length 2 or 100 or any number in between.

(20 points) Documentation:

At the top of your ".asm" file, give a brief description of the program, author name and last modified date in the form of comments.

For the code, use your judgment to choose between explaining each line or related group of lines. The purpose of these comments is to make the code readable for other programmers or for you in the future.