60-266 – Assignment #1

DUE DATE: ***Sunday October 6 2019.*** To be submitted via Blackboard by 11:59PM.

WARNINGS: You must use only instructions and directives described in Chapt-1 to Chapt-4a.

Question #1 (10 points)

Write an ASM program that evaluates the following expression, using variables:

$$Z = (A - B) - (C - D)$$

1. Declare and initialize the memory variable *A* to 32-bit signed integer value -543210 and variable *B* to 16-bit signed integer value -3210.
2. Declare the memory variables *C* and *D* and read in their values from the keyboard as 32-bit signed integer value -43210 and 8-bit signed integer values -10 (not -210 as before), respectively.
    a. You should display a message asking for the value of the variable, for example: "What is the value of *C*?" then
    b. Read in the value of *C* from the keyboard and then
    c. Display the value you have read beside (ie, to the right of) the message "What is the value of *C*?". Then repeat for variable *D* in the next line.
3. Variable *Z* should be declared as a 32-bit signed integer.
4. Display the string "*Z* = (*A* - *B*) - (*C* - *D*)" alone in a single line.
5. Display the values of all the variables *A, B, C, D* together in the next line (in the order in which they appear in the expression); each separated by 3 spaces and a semicolumn and 3 spaces again.
6. Display an empty line.
7. Display the final result contained in variable *Z*, in binary, then in decimal, and then in hexadecimal; each in a separate line.

Question #2 (10 points)

Write an ASM program that reads a value for the variable **littleEndian** and then uses a sequence of MOV instructions to fill the array **bigEndian** with the bytes of **littleEndian** in reverse order. For instance, if **littleEndian** = 12345678h then **bigEndian** should be filled as 12h, 34h, 56h, 78h (not 78h, 56h, 34h, 12h as before).

```
.data
bigEndian      BYTE ?, ?, ?, ?
littleEndian   DWORD 12345678h
```

You must first read in an arbitrary value for **littleEndian** and then fill and display the content of **bigEndian** in a single line.

For example: if I read the value FEDCBA98h for **littleEndian**, then **bigEndian** will be first filled and then displayed as FEh, DCh, BAh, 98h (not 98h, BAh, DCh, FEh as before).

littleEndian must be read as a hexadecimal number; use the readHex library function. Likewise, the cells of **bigEndian** must be displayed in hexadecimal; use the writeHex library function. Both functions are described in the textbook.