

60-266 – Assignment #2

DUE DATE is: *Tuesday, October 22, 2019*. To be submitted via Blackboard by 11:59PM.

WARNINGS: You must only use instructions and directives discussed from Lecture 1 ([chapt_01.pptx](#)) to Lecture 6 ([chapt_04-c.pptx](#)).

Programming Exercise 1 (10 points): [call it Ass2-Q1.asm]

Write an ASM program that reads an integer number N and then displays the first N values of the Fibonacci number sequence, described by:

$$\text{Fib}(0) = 0, \text{Fib}(1) = 1, \text{Fib}(N) = \text{Fib}(N-2) + \text{Fib}(N-1)$$

Thus, if the input is $N = 10$, your program Ass1-Q1.exe should display the following single line:

Fibonacci sequence with $N = 10$ is: 0 1 1 2 3 5 8 13 21 34 55

Programming Exercise 2 (50 points): [call it Ass2-Q2.asm]

Write an ASM program that prompts the user to enter a string of at most 128 characters and then displays the string in reverse order, with: each upper-case letter converted to its corresponding lower-case letter, and each lower-case letter converted to its corresponding upper-case letter. The program should also display the number of uppercase letters after displaying the output string, as well as the total number of characters in the string. For instance, a sample execution of “Ass2-Q2.exe” with the input string “An Input Line!” is shown below

```
C:\Programming\asm>Ass1-Q2
```

```
Enter a string of at most 128 characters: An Input Line!
```

```
Here it is, with all lowercases and uppercases flipped, and in reverse order:
```

```
!ENII TUPNi Na
```

```
There are 8 upper-case letters after conversion.
```

```
There are 14 characters in the string.
```

```
C:\Programming\asm>
```

HINT: Solving this question in the following sequential order will be much easier; though you can solve the way you want. It is always much easier to solve a problem when you break it down into smaller problems; don't care if your code is long.

- 1) First, read the string from the keyboard into a memory variable. Count the number of characters while reading the string.
- 2) Second, convert initial lowercases to uppercases and initial uppercases to lowercases; be careful here. You can also do this while reading the string.
- 3) Third, count the final number of uppercases; after converting all lowercases to uppercases. You can also do this while reading the string.
- 4) Fourth, print the resulting string in reverse order, by using indirect addressing (or indexed addressing, if you wish). Also, be careful here since there are two possible ways: you can either reverse the initial string first then display the resulting reversed string, or you can directly display the initial string in reverse order.
- 5) Display the two counts.

If the user enters more than 128 characters, only the first 128 characters must be processed (the rest are ignored but make sure that you cannot write outside the memory you have allocated for storing the string).

Also, try to make use of data-related operators as much as possible, such as OFFSET, SIZEOF, TYPE, LENGTHOF, DUP or PTR, in order to make your program as flexible as possible (and as short/efficient as possible); see Chapt_04-c.