# EE213 Computer Organization and Assembly Language
# Semester Assignment - Fall 2017

Open Date: Thursday, November 30, 2017

Due Date: Friday, December 15, 2017 (23:00)

**Q1**: As you now have experienced assembly language, elaborate in your own words, what differences have you found in 32 and 16 bit architectures, your elaboration should include the discussion over registers, busses, stack, variables, memory addresses, etc.

**Q2**: (Inline Assembly) Write an assembly language subroutine that receives two input parameters: the offset of an array and the array's size. It must return a count of the longest increasing sequence of integer values. For example, in the following array, the longest increasing sequence begins at index 3 and has a length of **4** {14, 17, 26, 42}:

```
[-5, 10, 20, 14, 17, 26, 42, 22, 19, -5]
```

Construct the array by taking inputs from the user in CPP part of your program. Call your ASM subroutine by passing the arguments, and print the values returned by the subroutine (either in asm or cpp part of the program).

**Q3**: What does it mean by triple encryption/Decryption? Construct a program where different methods would have been implemented to achieve triple encryption such that each procedure has its own encryption/decryption formula. Show your results in the form of output text files (at every stage of encryption/decryption).

**Q4**: Construct a mini project that should:

- ✓ Take some text file as input (e.g. *input.txt*) having, read it character by character and then results in a text file (*output.txt*) that should hold combination frequencies for each pair of characters.
- ✓ It is necessary to submit the algorithm (in pseudocode) along with solution.
- ✓ You can make use of inline assembly OR multi-module programming, where algorithm must have been implemented in assembly language.
- ✓ E.g. given that input.txt has following single paragraph:

  *"A brown burnt bear on a brushed bridge"*

  The *output.txt* will contain the following frequencies:

| | **A** | **B** | **R** | **O** | **W** | **N** | **U** | **T** | **E** | **O** | **S** | **H** | **D** | **I** | **G** | **Total** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **A** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| **B** | 0 | | 3 | | | | 1 | | 1 | | | | | | | 5 |
| **R** | 0 | | | 1 | | 1 | 1 | | | | | | | 1 | | 5 |
| **O** | 0 | | | | 1 | 1 | | | | | | | | | | 2 |
| **W** | 0 | | | | | 1 | | | | | | | | | | 1 |
| **N** | | | | | | | | 1 | | | | | | | | 3 |
| **U** | | | 1 | | | | | | | | 1 | | | | | 2 |
| **T** | | | | | | | | | | | | | | | | 1 |
| **E** | 1 | | | | | | | | | | | | 1 | | | 3 |
| **O** | | | | | 1 | 1 | | | | | | | | | | 2 |
| **S** | | | | | | | | | | | | 1 | | | | 1 |
| **H** | | | | | | | | | 1 | | | | | | | 1 |
| **D** | | | | | | | | | | | | | | | 1 | 2 |
| **I** | | | | | | | | | | | | | 1 | | | 1 |
| **G** | | | | | | | | | 1 | | | | | | | 1 |

**Q5**: Construct and display a 5 * 5 word array by taking inputs from the user.

I. Show sum, average, minimum, maximum of each row in the results.
II. Show sum, average, minimum, maximum of each column in the results
III. Show sum, minimum, maximum, average of the table.

**Q6:** Answer the following:

**I.** Provide opcodes for the following MOV instructions:

```
.data
myByte BYTE ?
myWord WORD ?
.code
mov ax,@data
mov ds,ax
mov es,ax                       ; a.
mov dl,bl                       ; b.
mov bl,[di]                     ; c.
mov ax,[si+2]                   ; d.
mov al,myByte                   ; e.
mov dx,myWord                   ; f.
```

**II.** Provide Mod R/M bytes for the following MOV instructions:

```
.data
array WORD 5 DUP(?)
.code
mov ax,@data
mov ds,ax
mov BYTE PTR array,5            ; a.
mov dx,[bp+5]                   ; b.
mov [di],bx                     ; c.
mov [di+2],dx                   ; d.
mov array[si+2],ax              ; e.
mov array[bx+di],ax             ; f
```

**III.** Assemble the following instructions by hand and write the hexadecimal machine language bytes for each labeled instruction. Assume that val1 is located at offset 0. Where 16-bit values are used, the bytes must appear in little endian order:

```
.data
val1 BYTE 5
val2 WORD 256
.code
mov ax,@data
mov ds,ax                       ; a.
mov al,val1                     ; b.
mov cx,val2                     ; c.
mov dx,OFFSET val1              ; d.
mov dl,2                        ; e.
mov bx,1000h                    ; f.
```

**Q7:** Briefly describe the following terminologies:

- ✓ Stack and Stack Frames,
- ✓ .model Directive: Language Specifiers, Stack Distance
- ✓ String Primitive Instructions: **MOVS**: Move string data; **CMPS**: Compare strings; **SCAS**: Scan string; **STOS**: Store string data; **LODS**: Load accumulator from string

- ✓ Interrupts: Interrupts and Interrupt Services Routine, Interrupt Vector and Interrupt Vector Table, IRQ
- ✓ ISA: CISC, RISC

******************