National University of Computer & Emerging Sciences, Karachi.
FAST School of Computing,
Mid Term I, Fall 2021.
October 13, 2021, 9:00 am – 10:00 am.

Course Code: EE 2003	**Course Name:** Computer Organization and Assembly Language
Instructors: Dr. Nouman M Durrani, Shoaib Rauf, Aashir Mahboob, Aamir Ali, and Qurat ul Ain	
Student's Roll No:	**Section:**

Instructions:

- Except for your Roll No and Section, DO NOT SOLVE anything on this paper.
- Return the question paper.
- Read each question completely before answering it. There are **3 questions on 2 pages.**
- In case of any ambiguity, you may make an assumption. But your assumption should not contradict any statement in the question paper.
- All the answers must be solved according to the SEQUENCE given in the question paper, otherwise, points will be deducted.
- This paper is subjective.
- Where asked for values, only provide the **hex-decimal** values.
- Problems needing iterations should be coded using iterative instructions. No points will be awarded otherwise.

Time Allowed: 60 minutes. **Maximum Points**: 32 points

==========

Q. No. 1 Briefly answer each of the following: [8 x 2 = 16 points]

(i) Explain the difference between the direct-offset and indexed operands with examples.

```
.data
 arrayB byte 10h,20h,30h,40h,50h,60h
.code
 mov al, [arrayB+0] ; Direct offset
        mov esi,4
        mov al, [arrayB+esi] ; Indexed offset
```

(ii) In which case, we use MOVSX instruction. Consider an example where a smaller value is moved to larger destinations.

```
mov v1,-9
movsx eax, v1
```

(iii) Discuss the hardware viewpoint of Signed and Unsigned integers.

(iv) Discuss the term label as an identifier and directive with examples.

(v) The LOOP instruction creates a counting loop. What happens when a loop instruction is encountered?

```
1. ecx -1; Substract ecx value by 1
2. cmp ecx,0; Check ecx value is equal to 0. If 0 then exit otherwise
loop continues
```

(vi) Why the concept of virtual machines is studied? At which level does assembly language appear at the virtual machine level? Consider L1 as the lowest level.

```
Virtual Machine: Virtual machine is a machine that translates and
then interperates the assembly language code into machine codes.
LEVEL 2.
```

(vii) How do we load and store the status flag bits? Give example instructions.

```
.data
  Saveflag byte ?
.code
Lahf
Mov saveflag, ah
Mov ah,saveflag
sahf
```

(viii) Compare the following x86 processors' mode of operations:

(a) Real address mode and

(b) Protected address mode

Real Mode

only 1 MB of memory can be accessed from 0 to FFFFF

Program can access any part of main memory

MS-DOS runs in real-address mode

Protected Mode

Each program can access a maximum of 4GB of memory

OS assign memory to each running program

Program are prevalent from accessing each other memory

Windows NT, 2000, XP and Linux used Protected mode

Q. No. 2 Consider the following initialization: [3 + 3 + 2 = 8 points]

```
X1    WORD  0E342H, 4 DUP(0Eh)
```

(i) Assign proper physical addresses (using a real address mode) to each byte stored in the data segment, and draw a memory map. (Assume DS= 2FF0h, and the starting offset is 2304h).

32204h

(ii) For the above data definition directives, give the content of the destination register after execution of each of the following instructions:

```
MOV  EAX, DWORD PTR X1          ;(a) EAX = 000E E3242h
MOV  BL, SIZEOF X1              ;(b) BL = 0Ah
MOV ESI, 4
MOV   BX, [X1+ESI]              ;(c) BX = 000Eh
```

(iii) Where indicated, write down the values of the Carry, Sign, Zero, and Overflow flags after each instruction has been executed:

```
MOV AX,7FF0H
ADD AL,10H   ; (a) CF = 1    SF = 0    ZF = 1    OF = 0
ADD AH,1     ; (b) CF = 0    SF = 1    ZF = 0    OF = 1
```

Q. No. 3 Write assembly language programs for the following problems: [4 + 4 = 8 points]

(i) Declare two variables “val1” of type BYTE and “val2” of type WORD, initialized with hexadecimal values 79h and 100h respectively. Find the multiplication of these variables using a loop instruction and store the result into a third variable “val3” of type DWORD.

```
.data
        val1 byte 79h
        val2 word 100h
        val3 dword ?
.code
        Main PROC
                movzx eax, val2
                movzx ecx, val1
                mov val3,0
                movzx edx,val2
        mult:
                add eax, edx
```

```
        mov val3, eax
        loop mult:

        mov eax,val3
        call WriteInt
        exit
    Main Endp
End Main
```

(ii) The Lucas sequence has the same recursive relationship as the Fibonacci sequence, where each term is the sum of the previous two terms, but with different starting values. If the starting values are 2 and 1, write an assembly language program that finds five missing elements in the following series.

2, 1, 3, 4, 7,__, __, __, __, __ .

```
.data
 x1 word 2,1,3,4,7,5 DUP(?)
 valA dword 5
.code
 Main PROC
        mov ecx, valA
        mov esi, 8
 Fibo:
        mov ax, [x1+esi]
        add esi,2
        mov bx, [x1+esi]
        add ax,bx
        add esi,2
        mov [x1+esi], ax
        sub esi,4
        loop Fibo

        exit
 Main Endp
End Main
```