FAST- National University of Computer & Emerging Sciences, Karachi.
Department of Computer Science,
Mid Term I Examinations, Spring 2019.
27th February, 2019, 9:00 pm – 10:00 pm

Course Code: EE 213	**Course Name:** Computer Organization and Assembly Language
Instructor: Muhammad Danish Khan	
Student's Roll No:	**Section:**

Instructions:

- Attempt all questions.
- Return the question paper.
- Read each question completely before answering it. There are **3 questions on 3 pages.**
- In case of any ambiguity, you may make 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.
- 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**: 30 points

==========

Q No. 1 Briefly answer each of the following questions, examples are necessary where asked. [**10 Points**]

(i) How direct addressing is different from indirect addressing? Give an example.

(ii) Discuss the difference between OVERFLOW FLAG (OF) and CARRY FLAG (CF) with the help of an example.

(iii) Which two registers are used when microprocessor access some byte(s) of data and why?

(iv) List one special function for each of the data registers (EAX, EBX, ECX, EDX).

(v) How PTR is different from LABEL? Briefly explain with the help of an example.

Q No. 2

(i) Given the following array, using LOOP write some code that should swap the elements in specified order: *1st* with *2nd*, *3rd* with *4th*, *5th* with *6th*, *and 7th* with *8th*. [**5 Points**]

```
ARRAY1  SDWORD    12h, 11h, 14h, 13h,16h,15h, 18h, 17h, 19h, 20h
```

After Swapping: *ARRAY1 = 11h,12h,13h,14h,15h,16h,17h,18h, 19h, 20h*

```
.data
      ARRAY1 SDWORD 12h, 11h, 14h, 13h,16h,15h, 18h, 17h

      .CODE
      MAIN PROC
            mov    esi, OFFSET array1          ; starting OFFSET
            mov    ebx, TYPE array1            ; doubleword = 4 bytes
            mov    ecx, LENGTHOF array1        ; number of units in arrayD
            call   DumpMem                     ; display memory

      MOV    ECX, 4
      L1:    MOV    EAX, [ESI]
             XCHG   EAX, [ESI+4]
```

```
        MOV    [ESI], EAX
        ADD    ESI, 8
        LOOP   L1
    mov    esi, OFFSET array1          ; starting OFFSET
    mov    ebx, TYPE array1            ; doubleword = 4 bytes
    mov    ecx, LENGTHOF array1        ; number of units in arrayD
    call   DumpMem                     ; display memory
    ret
    EXIT
    MAIN ENDP
    END main
```

(ii) Assuming the following array, write some assembly code that should sum up all the EVEN NUMBERS in the array and stores the resulting value in a variable named *result*. You must use base-offset addressing mode for processing array elements. [**5 Points**]

```
ARRAY1  WORD 0, 1, 2, 3, 4 ..... 99
```

```
MAIN PROC
    MOV    ESI, 0
    MOV EAX, 0
    MOV    ECX, LENGTHOF ARRAY1/2
    L1:    ADD AX, [ARRAY1+ESI]
           ADD ESI, 4
    LOOP   L1
    MOV RESULT, AX

       MAIN ENDP
```

Q No. 3 Assume the following data segment (starting from 0000 FFFFh) for the following questions.

```
.data
    arr1        SBYTE       3 DUP(-127)
    arr2        WORD        2, 2 DUP(?)
                DWORD       2 DUP (7FE09A9h),$
```

0000 FFFF	-127	0001 0008	A9	0001 0011	00
0001 0000	-127	0001 0009	09	0001 0012	01
0001 0001	-127	0001 000A	FE	0001 0013	00
0001 0002	02	0001 000B	07		
0001 0003	00	0001 000C	A9		
0001 0004		0001 000D	09		
0001 0005		0001 000E	FE		
0001 0006		0001 000F	07		
0001 0007		0001 0010	10		

		main PROC	
1.	00FF C10C	MOV AL, [arr1+1]	;-127
2.	00FF C10D	MOV ESI, OFFSET[arr2 +6]	; 0001 0009
3.	00FF C10E	MOV DX, WORD PTR [arr2+7]	; FE09
4.	00FF C10F	ADD AL, AL	
5.	00FF C110	MOV ECX, 0Ch	
6.	00FF C111	JMP L1	
7.	00FF C112	INC DL	
8.	00FF C113	INC CL	
9.	00FF C114	L1: SUB CL,DL	; CL = 3

10.	00FF C115	MOV AL,DL ; AL = 09h
11.	00FF C116	L2: ADD AL, 2
12.	00FF C117	LOOP L2 ; AL = 0F
13.	00FF C118	MOV BYTE PTR [ESI],AL ; [0001 0009] = 0F 07 FE 17 0Fh
		main ENDP

(i) What will be the last element in the data segment?

0001 0010h

(ii) What will be the final value of AL?

0Fh

(iii) What will be the Status of CF, ZF, and OF after line 4 is executed?

CF: **SET(1)** ZF: **CLEAR(0)** OF: **SET(1)**

(iv) What is stored in EIP after line 6 is executed?

00FF C114h

(v) Draw Byte by Byte memory (with addresses) for **DWORD array** (unnamed) after execution of above code.

07FE170Fh, 07FE1709A9, 0001 0011

0001 0008	A9	0001 0011	00
0001 0009	09	0001 0012	01
0001 000A	FE	0001 0013	00
0001 000B	07		
0001 000C	A9		
0001 000D	09		
0001 000E	FE		
0001 000F	07		
0001 0010	10		

STAY BRIGHT