Roll Number:____________ Section: _________

## National University of Computer and Emerging Sciences, Lahore Campus

| | | | |
|---|---|---|---|
| **Course:** | **Computer Organization and Assembly Language** | **Course Code:** | **EE213** |
| **Program:** | **BS(Computer Science)** | **Semester:** | **Fall 2020** |
| **Duration:** | **90 Minutes** | **Total Marks:** | **45** |
| **Paper Date:** | **20-Oct-2020** | **Page(s):** | **6** |
| **Section:** | **All** | **Section:** | |
| **Exam:** | **Midterm I** | **Roll No:** | |

**Instruction/Notes:**

- Exam is Open book, Open notes.
- Properly comment your code.
- You **CANNOT** use an instruction **NOT** taught in class.
- If there is any ambiguity, take reasonable assumption. Questions during exam are not allowed.
- Write your answer in the space provided. You **can take extra sheets BUT they WON'T BE ATTACHED WITH THE QUESTION PAPER OR MARKED.**

### Question 1 [30 Marks]: Short Questions

**i.** **[2 marks]** How is the following data stored in memory?
array1: dw 0x0A0B, 1, 0x0C0D

**ii.** **[4 Marks]** Consider the following data, write the HEX value of ax/ah after each instruction.

```
; numbers are declared in HEX (h)
nums:     dw 0A0Bh,
          db 0Ch,
          dw 0D0Eh,
          dd 0102030405060708h

mov ax, [nums]              ;  ax =0x________________________

mov ah, [nums+2]            ; ah =0x_________________________

mov ax, [nums+3]            ; ax =0x_________________________

mov ah, [nums+9]            ; ah =0x_________________________
```

**iii.** **[4 Marks]** Write a two-line code for **Shift Logical Right (SHR) a 32-bit number by 1 bit**.
For example, 32-bit number 0x22446688 should be 0x11223344 after SHR

iv. **[5 Marks]** Mark each of these instructions Valid or Invalid. In case of Invalid, give one-line reason.

| | | Valid/ Invalid | Reason |
|---|---|---|---|
| **a.** | Mov ax, [num1+bx] | | |
| **b.** | Mov word [bx+bp], 10 | | |
| **c.** | Mov [bx+si], ax | | |
| **d.** | Mov [bx-si], al | | |
| **e.** | Mov [num1], [num2] | | |

v. **[4 Marks]** Execute the code given below and write down the values of the AX (in hex), Carry, Sign, Zero, and Overflow flags where required.

```
mov al, 7Fh

add al,1                    ; AL = ___80h_____(h) CF = _0_  SF = _1__     ZF         =

_0__OF = __1___

mov ax, 0XFFFE

add ah,1                    ; AL = ____80h___(h) CF = _1__ SF = _0__   ZF = _1___OF=

_0____

add al,1                    ; AL = _00FF__(h) CF = __0_  SF = _1__   ZF = __0__OF =

___0__
```

vi. **[3 Marks]** After running the following code what would be the **Binary value** of **CL**, **CF** and **AL** in the end?

```
mov cl, 1111 1111 b ; value is given in binary, spaces are for readability
mov al, 1101 1101 b
and cl, 1101 1111 b
```

```
    and al, 1
    ror al, 1
```

**CF:_____1___________** **AL:__10000000__________**

**CL:__11011111________**

**vii.** **[3 Marks]** Show the **decimal value of AX, BX and CX** after the execution of the following code.

```
mov cx, 5
mov ax, 3
mov bx, 0
add ax, 1
push cx
mov cx, 2
L2:
        add bx, 1
        sub cx, 1
        cmp cx, 0
        jne L2
pop cx
sub cx, 1
```

**AX = ________________________**

**BX= ________________________**

**CX= ________________________**

**AX=_______8__________**

**BX=____10__________**

**CX=___0____________**

**viii.** **[5 Marks]** For the code given below, identify the error in code. Highlight the exact line(s)/region(s) having errors. Correct the error (in same space) such that the code works correctly according to requirements mentioned in comments.

```
    jmp start
; Sum takes two parameters and adds them in AX
Sum: push bp
        mov bp, sp

        pop cx                  ;trying to read parameter 2 in cx
        pop dx                  ;trying to read parameter 1 in dx

        ;trying to add two parameters (0xA and 0xB) in ax
        mov ax, 0
        add ax, cx
        add ax, dx

        Ret 2           ;Callee trying to clear stack

start:  mov ax, 0xA
        push ax         ;Parameter 1

        mov ax, 0xB
        push ax         ;Parameter 2

        call Sum
          ; ax should contain sum of 0xA and 0xB here
mov ax, 0x4c00
int 0x21
```

**Question 2 [15 Marks]:** Write an assembly language program that processes a *Given_Array* (of bytes), calculates **pairwise sum** of its elements and saves the result in *PairwiseSumArray*. First element is paired with the last element, the 2nd element is paired with the 2nd last element and so on. See the Sample Run below for detail.

**Sample Run:**

| **Example 1, even sized array** | **Example 2, odd sized array** |
|---|---|
| **ArrSize:** 8 | **ArrSize:** 9 |
| **Given_Array:** 10, 2, 3, 4, 50, 62, 70, 8 | **Given_Array:** 10, 2, 3, 4, 77, 50, 62, 70, 8 |
| **PairwiseSumArray:** 18, 72, 65, 54 | **PairwiseSumArray:** 18, 72, 65, 54, 77 |
| **Description:** Elements of Pairwise Sum Array are calculated as: (10+8 = 18), (2+70 = 72), (3+62 = 65), (4+50 = 54) | **Description:** Result is the same as Example 1, the middle element, with no pair, is copied at the end as it is. |

**Note:** You are not required to write subroutines. You may keep both the arrays (i.e. Given_Array and PairwiseSumArray) of ArrSize and ignore extra space at the end of PairwiseSumArray.

```
[org 0x0100] ; Write your code below
```

Roll Number:____________ Section: _________

```
end:  mov ax, 0x4c00   ; terminate the program
      int 0x21
```

**Rough Work**