Name: ____________________ Roll Number: ____________________ Section: ____________________

National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Computer Organization and Assembly Language**	**Course Code:**	**EE2003**
		Semester:	**Spring 2022**
Program:	**BS (Computer Science)**	**Total Marks:**	**30**
Duration:	**60 Minutes**	**Weightage:**	**15**
Paper Date:	**21-Mar-2022**	**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.**
- All other rules pertaining to examinations as per NUCES policy apply.

Question 1 [20 Marks]: Short Questions

i. **[3 Marks]** What is the value of ZF, SF and CF after the execution of the test instruction in the code given below?

```
[org 0x100]
          jmp start
num1:   dw  231Bh, 2337h, 124Ch

start:      clc
            mov ax, [num1]
            add ax, [num1+2]
            add ax, [num1+4]
            mov bx, 0100h
            test ax, bx

          mov ax, 0x4c00
          int 0x21
```

Answer:

1. ZF = 1
2. SF = 0
3. CF = 0

ii. **[5 marks]** The value of code segment (cs) and stack segment (ss) register is 4228h while the value of different registers is as follows:

bx: 2000h, ip: 0100h, di: 0400h, bp: 1111h, si: 0110h

Write the physical address of the following memory access. Also point out which type of wraparound is there if occurred, segment or whole memory?

	Memory Location	Physical Address in hex	Wraparound Type (if occurred)
a	[cs:bp + si]	434A1h	None

Show your working in this box:

a. Effective Address = bp+si = 1111h + 0110h = 1221h (No Wrapped around occurred)
Physical Address = CS * 10h + Offset (Effective Address) = 42280h + 01221h = 434A1h (No Wrapped around occurred)

iii. **[2 Marks]** Mention the addressing modes in each of the following instructions.

		Mode
a.	mov al, [bx+di]	Base + Index mode
b.	mov ax, [num1]	Direct mode
c.	mov ax, [bp+di+400]	Base + Index + offset mode
d.	mov ax, [bx+4000]	Base Register Indirect + offset Mode

iv. **[4 Marks]** Mark each of these instructions Valid or Invalid. In case of Invalid, give one-line reason.

		Valid/ Invalid	**Reason**
a.	xchg [34BFh], [3452h]	Invalid	
b.	push al	Invalid	
c.	mov ip, [4562h]	Invalid	

d.	mov [bx+si], dx	Valid	

v. **[4 Marks]** For the code given below, write the decimal values stored in memory label var1 after the execution of the program. You also have to briefly explain the working of this program.

```
[org 0x100]
        jmp start

array:  dw -1, 7, 9, -2, 2, 0
var1:   dw 0

proc:   mov bx, array
        mov cx, 6
        mov dx, 0
        stc

A1:     adc dx, [bx]
        add bx, 2
        sub cx, 1
        jnz A1

        mov [var1], dx

        ret

start:  call proc

        mov ax, 0x4c00
        int 0x21
```

Answer:

Memory label var1 will have a value of 10h. It basically adds all the elements to the CF, which has an initial value of 1.

vi. **[2 Marks]** Following data is stored in memory label num1:

num1: dw 231Bh, 2337h, 124Ch

You have to complete the following table and show the data placement in memory.

num1	1Bh
num1+1	23h
num1+2	37h
num1+3	23h

num1+4	4Ch
num1+5	12h

Question 2 [10 Marks]: Write an assembly language program to perform pairwise scan operation on an array such that 1st element is paired with the 2nd element, the 3rd element is paired with the 4th element and so on. Assume that last element of the array is -1 an indicator to stop the array iteration. If 1st element of a pair is even, then divide the 2nd element by 4 through bit manipulation and store the quotient in place of the 2nd element. However, if 1st element of the pair is odd, then multiply 2nd element by 2 through bit manipulation and store the result in the location of the 2nd element. In case the array contains odd number of elements, then save the last element as it is.

See a sample run below for detail.

Sample Run:

Example 1, even sized array (excluding the last element)	**Example 2, odd sized array (excluding the last element)**
Input Array: 3, 5, 10, 9, 12, 16, -1	**Input Array:** 3, 5, 10, 9, 12, 16, 23, -1
Output Array: 3, 10, 10, 2, 12, 4, -1	**Output Array:** 3, 10, 10, 2, 12, 4, 23, -1

Answer:

```
[org 0x0100]

jmp start

data:   dw 1, 8, 4, 2, 3, -1
start:  mov bx,0

loop1:  cmp word [data+bx], -1              ;check to stop loop
        jz end

        cmp word [data+bx+2], -1
        jz end

        mov ax, [data+bx]                   ; check next element either even or odd
        shr ax, 1
        jc odd

even:   shr word [data+bx+2], 2             ; divide by 4 in case of an even number
        jmp next

odd:    shl word [data+bx+2], 1             ; multiply by 2 in case of an odd number

next:   add bx, 4                           ; iterate to next elements
```

```
        jmp loop1

end:    mov ax, 0x4c00

        int 0x21
```

Name: ____________________ Roll Number: ____________________ Section: ____________________

Name: ____________________ Roll Number: ____________________ Section: ____________________