National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Computer Organization and Assembly Language	Course Code:	EE213
Program:	BS(Computer Science)	Semester:	Fall 2018
Duration:	60 Minutes	Total Marks:	
Paper Date:		Weight	15%
Section:	ALL	Page(s):	
Exam Type:	Mid-1		

Student : Name:_____ **Roll No.**_____ **Section:**_____

Instruction/Notes:

1. Exam is Open book, Open notes.
2. Properly comment your code.
3. You **CANNOT** use an instruction **NOT** taught in class.
4. Write your answer in the space provided. You **can take extra sheets BUT they WONT BE ATTACHED WITH THE QUESTION PAPER OR MARKED.**

Q1. Given the code below, answer the following questions **[5x3 Marks]**:

```
        [org 0x100]
1.      mov si, -1
2.      mov di, [size]
3.
4.      l1:
5.        add si, 1
6.        cmp si, di
7.        je end
8.        mov al, [arr+si]
9.        shr al, 1
10.       jnc l1
11.
12.     l2:
13.       sub di, 1
14.       cmp si, di
15.       je end
16.       mov al, [arr+di]
17.       shr al, 1
18.       jc l2
19.       cmp si, di
20.       jnl end
21.       mov al, [arr+si]
22.       mov ah, [arr+di]
23.       mov [arr+di], al
24.       mov [arr+si], ah
25.       jmp l1
26.     end:
27.
28.     mov ax, 0x4c00
29.     int 21h
30.     arr: db 3, 10, 2, 0, 7, 5, 8
31.     size: dw 7
```

i) If the label 'size' is at 0x0149 what is the total size of program in bytes (decimal value) excluding data (i.e. arr label?

66 bytes

ii) What changes will be made to 'arr' after the program execution? Fill the table below:

Before program:	arr:	3	10	2	0	7	5	8
After program:	arr:	8	10	2	0	7	5	3

iii) Rewrite the above code from line 4 to 10 without using shr.

```
    mov dl, 0x01

    l1:
     add si, 1
     cmp si, di
     je end
     mov al, [arr+si]
     test dl, al                    ;mask
     jz l1
     ;shr al, 1
     ;jnc l1
```

Q2. Implement 32 Bit Subtraction and write answer back into memory. **[10 Marks]**
Assume that your Architecture only supports these instructions.
Mov, shl ,shr ,rcl , ror , rol , add , and, not
Note: No Credit would be given if any other instruction is used.

```
[org 0x0100]

mov ax, [num1]
mov [result], ax
mov bx, [num2]
not bx
add bx, 1
add [result], bx

mov dx, 0
rcl dx, 1
add dx, 0xFFFF

mov ax, [num1+2]
mov [result+2], ax
mov bx, [num2+2]
not bx
add bx, 1
add [result+2], bx

add [result+2], dx

mov ax, 0x4c00
INT 21h
num1: dd 0x0203f405
num2: dd 0xABCDEF01
result: dd 0
```

Q3. Fill the following table. These instructions are from same program and are not independent. Write the corresponding output for the given registers' and flags' values. **[10 Marks]**

AX=0x5CAA DX=0x3729 CX=0x235A

Instructions	Updated value after executing the instruction			Flag values after the instruction execution		
	AL	DL	CL	CF	OF	SF
xor al, dl	83	29	5A	0	0	1
add dl, dl	83	52	5A	0	0	0
sub cl, dl	83	52	08	0	0	0
sar al, cl	ff	52	08	1	0	1
adc al, dl	52	52	08	1	0	0