Question 1 [8 Marks]

Find the value of AL for the following set of instructions. Perform calculations on the box given or on the back of this page also mention proper part number. No marks for direct answer.

		Calculations:
Part a: Mov al, -88 Shr ax,1	AL= 54h 0101 0100	1010 1000 0101 0100 ⎵ ⎵ 5 4
Part b: Mov ax, -88 Shr ax,1	AL= D4h 1101 0100	1111 1010 1000 0111 1111 1101 0100 D 4
Part c: Mov ax, -88 Sar ax,1	AL= D4h	
Part d: Mov al, -88 Sar ax,1	AL= 54h 0101 0100	
Part e: Mov ax, -88 Shr al,1	AL= 04h 0101 0100	
Part f: Mov al, -88 Shr al,1	AL= 54h 0101 0100	
Part g: Mov ax, -88 Sar al,1	AL= D4h 1101 0100	
Part h: Mov al, -88 Sar al,1	AL= D4h	

A 8

```
  4 2 1 8
0 1 0 1 1 000        FF FF
    5    8            58

  2 | 88             FFA7
  2 | 44 - 0          8 4   2 1  1111 1111 1010 1000
  2 | 22 - 0         1 0 1 0 1 0 0 0
  2 | 11 - 0            A        8
  2 | 5 - 1
  2 | 2 - 1
    | 1 - 0
```

Question 2[8 Marks]

1111   1010   1000
0101   0100

1.
2. Write a code for SHIFT LEFT operation of 64 bit number **num1**. Perform this shift
3. **num2** times. Note that your code should work for any values of num1 and num2.
4. values given below. Perform shift at word level.
5.
6.

```asm
.model small
.stack 0100h
.data
; data is defined here
num1 dq 11223344h
num2 word 0002h
```

;dq means define Quadword, it allocates 8 bytes
;dw means define Word, it allocates 2 bytes

```asm
.code
mov ax,@data
mov ds, ax

mov      cx, [num2]
L1:
    mov   ax, word ptr [num1]
    shl   ax, 1
    mov   word ptr [num1], ax

    mov   ax, word ptr [num1+2]
    rcl   ax, 1
    mov   word ptr [num1+2], ax

    mov   ax, word ptr [num1+4]
    rcl   ax, 1
    mov   word ptr [num1+4], ax

    mov   ax, word ptr [num1+6]
    rcl   ax, 1
    mov   word ptr [num1+6], ax
Loop L1

mov  ah, 04ch
int  21h
```

Question 3[10 Marks]

Consider following code and fill the **FILL STACK**. After executing complete code, update registers. Also show how **STACK** will look after execution. Stack segment starts at address **0100H** whereas data starts **(Array offset)** at **000DH**. *NOTE: Consider Line number as Instruction address. Clearly mention returning address on which line number.*

```
1  .model small
2  .stack 0100h
3  .data
4  array db 2,3,4
5  .code
6  jmp main
7  SumArray proc
8      push bp
9      mov bp,sp
10     mov si, [bp+6]
11     mov cx, [bp+4]
12     cmp cx,0
13     je L1
14     add al,[si]
15     dec cx
16     inc si
17     push si
18     push cx
19     call SumArray
20     L1:
21         pop bp
22     ret 4
23 SumArray endp
24 main proc
25     mov ax,@data
26     mov ds,ax
27     push dx
28     push di
29     push bx
30     mov ax, offset array
31     push ax
32     mov ax,lengthof array
33     push ax
34     mov ax,0
35     mov si,0
36     mov cx,0
37     call SumArray
38 main endp
```

FILL STACK

Address	Value
00C6	
00C8	
00CA	00E2
00CC	RA 20
00CE	0000
00E0	0010
00E2	00EA
00E4	RA 20
00E6	0001
00E8	000F
00EA	00F2
00EC	RA 20
00FE	0002
00F0	000E
00F2	0000
00F4	RA 38
00F6	0003
00F8	000D
00FA	0000
00FC	0000
00FE	0000

STACK AT END OF PROGRAM

Address	Value
00C6	
00C8	
00CA	
00CC	
00CE	
00E0	
00E2	
00E4	
00E6	
00E8	
00EA	
00EC	
00FE	
00F0	
00F2	
00F4	
00F6	
00F8	
00FA	0000
00FC	0000
00FE	0000

AX=	CX=	SI=	BP=
0009	0000	0010	0000

Question 4[10 Marks]

Use Booth's algorithm to multiply -4 (take it as Q) by 10 (take it as M). Show all steps. Consider your computer's data width is 8-bits.

A(Accumulator)	Q(Multiplier)	M(Multiplicand)	Q_{-1}	Operation performed (Add/Shift)
0000 0000	1111 1100	0000 1010	0	
0000 0000	0111 1110	''	0	SAR
0000 0000	0011 1111	//	0	SAR
1111 ·0110	0011 1111	//	0	A = A−m
1111 1011	0001 1111	//	1	SAR
1111 1101	1000 1111	//	1	SAR
1111 1110	1100 0111	//	1	SAR
1111 1111	0110 0011	//	1	SAR
1111 1111	1011 0001	//	1	SAR
1111 1111	1101 1000	//	1	SAR