

**MIT Practicals**  
**Assignment 4**  
**Krunal Rank (U18CO081)**

**Question 1: Write a program to load the data byte A8H in register C. Mask the high-order bits(D7- D4), and display the low-order bits (D3-D0) at an output port.**

```
1
2 ;<Question 1>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: mvi c,00a8h
11        mov a,c
12        ani 000fh
13        out 00h
14 hlt|
```

**Registers**

A	08	
BC	00	A8
DE	0E	07
HL	00	00
PSW	00	00
PC	42	0B
SP	FF	FF
Int-Reg	00	

Address (Hex)	Address	Data
00	0	8

**Question 2: Write a program to load the data byte 8EH in register D and F7H in register E. Mask the high-order bits (D7-D4) from both the data bytes, Exclusive-OR the low-order bits (D3-D0) and display the answer.**

```

1
2 ;<Question 2>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: mvi a,0fh
11         mvi d, 008eh
12         mvi e, 00f7h
13         ana d
14         mov d,a
15         mvi a, 0fh
16         ana e
17         xra d
18         out 00h
19
20
21 hlt

```

#### Registers

A	09	
BC	00	00
DE	0E	F7
HL	00	00
PSW	00	00
PC	42	12
SP	FF	FF
Int-Reg	00	

Address (Hex)	Address	Data
00	0	9

**Question 3: Write a program to load the bit pattern 91H in register B and 87H in register C. Mask all the bits except D0 from registers B and C.**

```
1
2 ;<Question 3>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: mvi b,91h
11        mvi c,87h
12        mvi a,0001h
13        ana b
14        mov b,a
15        mvi a,0001h
16        ana c
17        mov c,a
18
19
20 hlt
```

**Registers**

A	01	
BC	01	01
DE	00	00
HL	00	00
PSW	00	00
PC	42	10
SP	FF	FF
Int-Reg	00	

Question 4: Write a program to clear the CY flag, to load number FFH in register B, and increment B. If the CY flag is set, display 01 at the output port, otherwise, display the contents of register B.

```

1
2  ;<Question 4>
3
4  jmp start
5
6  ;data
7
8
9  ;code
10 start: stc
11         cmc
12         mvi b,00ffh
13         inr b
14         jnc nocarry
15         mvi a,01h
16         out 00h
17         jmp exit
18 nocarry: mov a,b
19         out 00h
20 exit:    hlt

```

Registers		
A	00	
BC	00	80
DE	00	00
HL	00	00
PSW	00	00
PC	42	13
SP	FF	FF
Int-Reg	00	

Address (Hex)	Address	Data
00	0	0

Remember that the question asked for Incrementing it, not adding 1.

**Question 5: Write a program to mask the lower bit of an 8 bit number.**

```
1
2 ;<Question 5>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: mvi a,31h
11        ani 00feh
12        hlt
```

**Registers**

A	30	
BC	00	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	08
SP	FF	FF
Int-Reg	00	

Question 6: Write a program:

Load two unsigned numbers in register B and register C respectively.

Subtract C from B. If the result is in 2's complement, convert the result in absolute magnitude and display it at PORT 1, otherwise, display the positive result. Execute the program with the following sets of data:

Set1:B=42H,C=69H

Set2:B=69H,C=42H

Set 3: B=F8H,C = 23H

```
1
2  ;<Question 6>
3
4  jmp start
5
6  ;data
7
8
9  ;code
10 start: mvi b,42h
11         mvi c,69h
12         mov a,b
13         sub c
14         jm neg
15         out 01h
16         jmp exit
17 neg:    xri 00ffh
18         adi 01h
19         out 01h
20 exit:    hlt
```

Registers		
A	27	
BC	42	69
DE	00	00
HL	00	00
PSW	00	00
PC	42	18
SP	FF	FF
Int-Reg	00	

Address (Hex)	Address	Data
00	0	0
01	1	39

```

1
2 ;<Question 6>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: mvi b,69h
11         mvi c,42h
12         mov a,b
13         sub c
14         jm neg
15         out 01h
16         jmp exit
17 neg:    xri 00ffh
18         adi 01h
19         out 01h
20 exit:    hlt

```

Registers		
A	27	
BC	69	42
DE	00	00
HL	00	00
PSW	00	00
PC	42	18
SP	FF	FF
Int-Reg	00	

Address (Hex)	Address	Data
00	0	0
01	1	39

```

1
2 ;<Question 6>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: mvi b,00f8h
11         mvi c,23h
12         mov a,b
13         sub c
14         jm neg
15         out 01h
16         jmp exit
17 neg:    xri 00ffh
18         adi 01h
19         out 01h
20 exit:    hlt

```

Registers		
A	2B	
BC	F8	23
DE	00	00
HL	00	00
PSW	00	00
PC	42	18
SP	FF	FF
Int-Reg	00	

Address (Hex)	Address	Data
00	0	0
01	1	43