M.I.T. LAB Assignment – 03

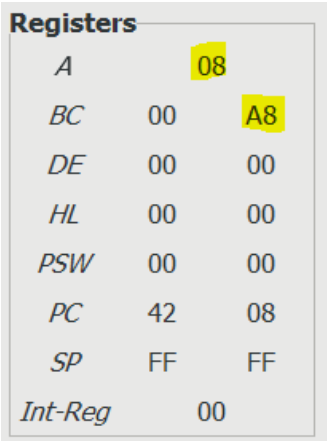
U19CS012

(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.

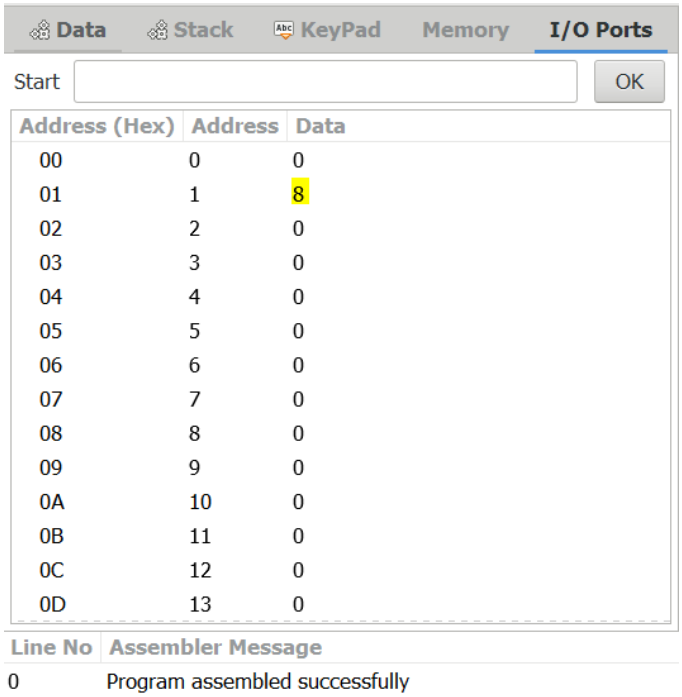
Notepad Code:

```
;Q-(1) Display Lower Order Bits [D3-D0]

; Load A8H in Register C
MVI C, 0A8H

; Copy it to Accumulator
MOV A, C

; AND with "0F" to Mask Higher Order Bits i.e. [D7-D4]
ANI 0FH

; Display the Lower Bits[D3-D0] at Output Port
OUT 01H

HLT
```

Registers and Memory:



Registers		
A	08	
BC	00	A8
DE	00	00
HL	00	00
PSW	00	00
PC	42	08
SP	FF	FF
Int-Reg	00	



Data | Stack | KeyPad | Memory | I/O Ports

Start OK

Address (Hex)	Address	Data
00	0	0
01	1	8
02	2	0
03	3	0
04	4	0
05	5	0
06	6	0
07	7	0
08	8	0
09	9	0
0A	10	0
0B	11	0
0C	12	0
0D	13	0

Line No	Assembler Message
0	Program assembled successfully

(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.

Notepad Code:

```
;(2) WAP 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

; Load 8EH in Register D
MVI D, 8EH
; Load F7H in Register E
MVI E, 0F7H
; Copy D to Accumulator
MOV A,D
; Mask High-Order (D7-D4) of D
ANI 0FH
; Copy it to Register D
MOV D,A
; Copy E to Accumulator
MOV A,E
; Mask High-Order (D7-D4) of E
ANI 0FH
; XOR between Masked high-order bits of D and E
XRA D
; Display at Output Port at Location 01h
OUT 01H
HLT
```

Registers and Memory: XOR ((1110) ^ (0111)) = (1001) = 9

Registers

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

Data | Stack | KeyPad | Memory | I/O Ports

Start OK

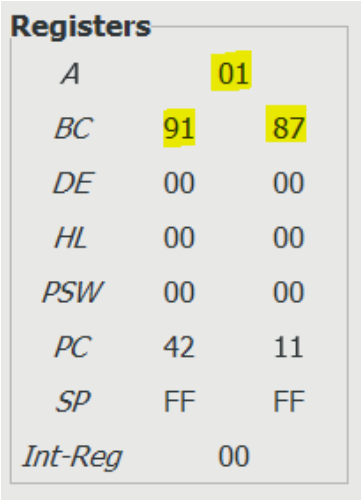
Address (Hex)	Address	Data
00	0	0
01	1	9
02	2	0
03	3	0
04	4	0

(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.

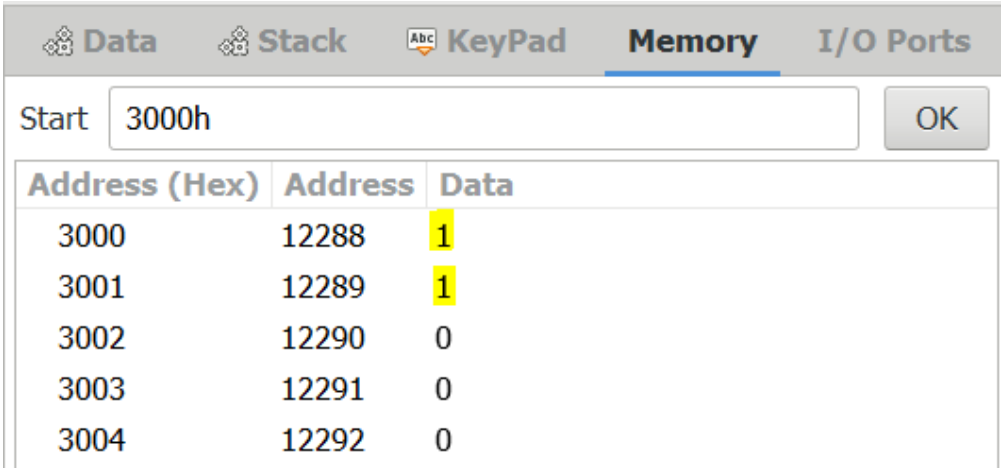
Notepad Code:

```
;(3) WAP 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.

; Load 91H in Register B
MVI B, 91H
; Load 87H in Register C
MVI C, 87H

MOV A, B
ANI 01H        ; Mask all Bits of Except D0 Bit ie (0000 0001) = (01)H
STA 3000H

MOV A, C
ANI 01H  ; Mask all Bits of Except D0 Bit ie (0000 0001) = (01)H
STA 3001H

HLT
```

Registers and Memory:



Registers		
A	01	
BC	91	87
DE	00	00
HL	00	00
PSW	00	00
PC	42	11
SP	FF	FF
Int-Reg	00	



Data | Stack | KeyPad | Memory | I/O Ports

Start 3000h OK

Address (Hex)	Address	Data
3000	12288	1
3001	12289	1
3002	12290	0
3003	12291	0
3004	12292	0

(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.

Notepad Code:

```
;(4) WAP 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.

; Load Number FFH in Register B
MVI B, 0FFH

; Increment Register B
INR B ; B = 00H

; Comment/Uncomment Based on Making Carry Flag = Set/Unset
STC     ; Set Carry Flag Cy = 1
;CMC        ; Clear Carry Flag Cy = 0

; Jump on Carry CY=1
JC Disp1
; If Carry CY=0, Display B
JNC Disp2

Disp1: MVI A,01h
       OUT 01h
       JC exit1
Disp2: MOV A,B
       OUT 01H
       JNC exit1

exit1: NOP

HLT
```

<u>Registers and Memory:</u>

<u>(1) Flag is Set (Cy=1)</u>

Registers

A	01	
BC	00	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	19
SP	FF	FF
Int-Reg	00	

Flag

S	0
Z	1
AC	1
P	1
C	1

Data | Stack | KeyPad | Memory | I/O Ports

Start 01h OK

Address (Hex)	Address	Data
01	1	1
02	2	0
03	3	0
04	4	0
05	5	0

(2) Flag is Not Set (Cy=0)

Registers			Flag	
A	00		S	0
BC	00	00	Z	1
DE	00	00	AC	1
HL	00	00	P	1
PSW	00	00	C	0
PC	42	1A		
SP	FF	FF		
Int-Reg	00			

Data | Stack | KeyPad | Memory | I/O Ports

Start 01h OK

Address (Hex)	Address	Data
01	1	0
02	2	0
03	3	0
04	4	0
05	5	0

(5) Write a program to mask lower bit of an 8-bit number.

Notepad Code:

```
;(5) Write a program to mask lower bit of an 8-bit number.

; Load HL Pair with Data 3000H
LDA 3000H
; Copy the Number to Reg B [For Reference]
MOV B, A
; AND Immediate F0H = (1111 0000) with Accumulator
ANI 0F0H        ; Lower bit Masked
; Store Result at Memory Location 3001H
STA 3001H
HLT
```

Registers and Memory:

Input: 135 = (87)H

Output: (87^(1111 0000))H = (80)H = 128

Registers		
A	80	
BC	87	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	0A
SP	FF	FF
Int-Reg	00	

Data | Stack | KeyPad | Memory | I/O Ports

Start 3000h OK

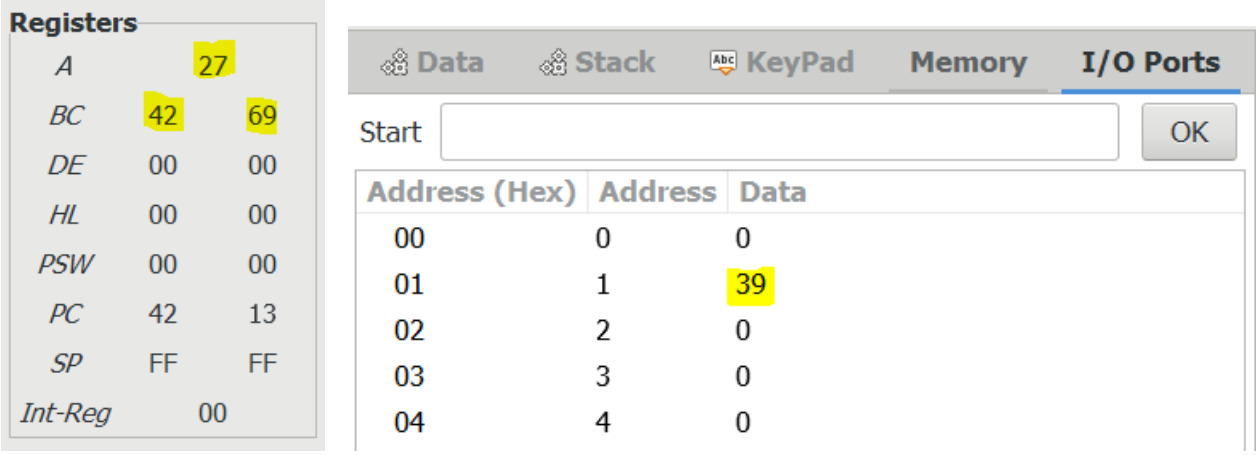
Address (Hex)	Address	Data
3000	12288	135
3001	12289	128
3002	12290	0
3003	12291	0
3004	12292	0

(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.

Notepad Code:

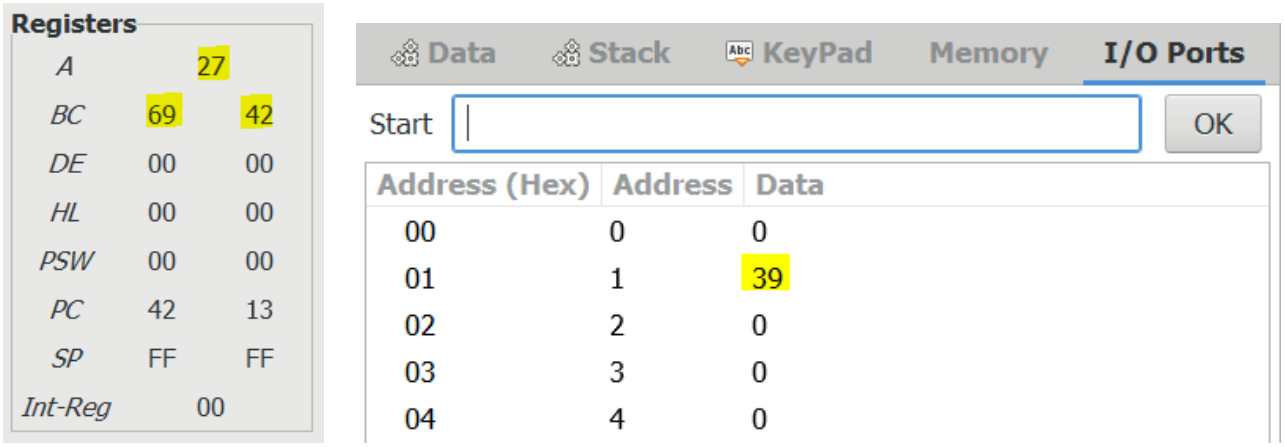
```
;(6) WAP to Load two unsigned numbers in reg. B and reg. C, respectively.
; Subtract C from B.
; If the result is in 2's complement,
; convert the result in absolute magnitude and display it at PORT 1,
; otherwise, display the positive result.
; Execute the program with the following sets of data.

; Load Value from Location 3000H for Reg B
LDA 3000H
MOV B,A
; Load Value from Location 3001H for Reg C
LDA 3001H
MOV C,A

MOV A, B
SUB C

; If Carry is Not Set, Jump to Output the Result
JNC print

; Otherwise Take 2's Complement
; Complement the Accumulator [1's Complement]
CMA
; Add 1 to (1's Complement)
ADI 01H

print: OUT 01H

HLT
```

Set1: B=42H -> 66, C=69H -> 105
Output: absolute(66-105) = 39 = (27)H



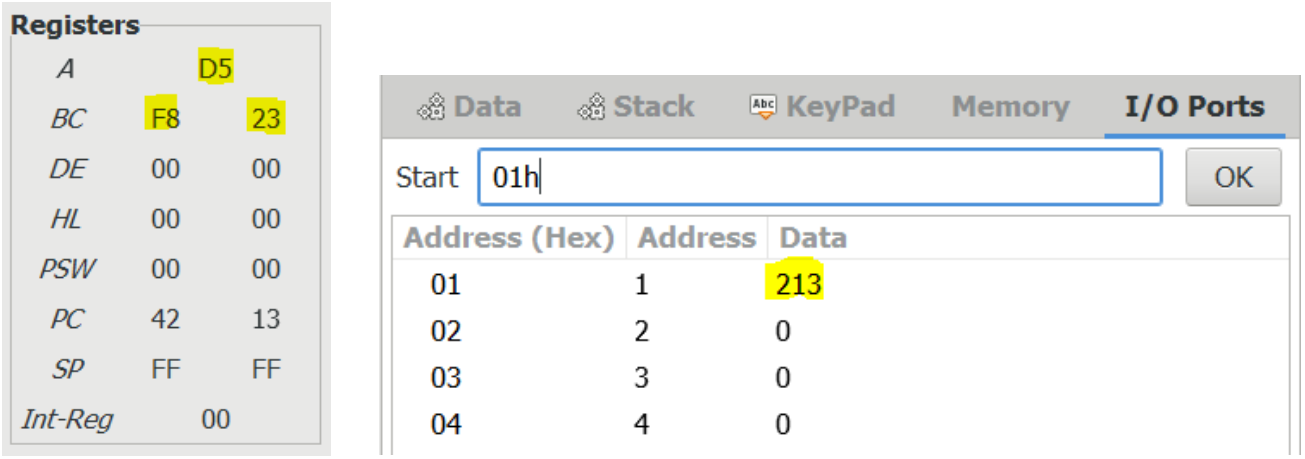
Set2: B=69H -> 105, C=42H -> 66

Output: absolute(105-66) = 39 = (27)H



Set3: B=F8H -> 248, C = 23H -> 35

Output: absolute(248-35) = 213



SUBMITTED BY: BHAGYA VINOD RANA [*U19CS012*]