M.I.T. LAB Assignment – 01

U19CS012

(1) Store the data byte 25H into memory location 3000H

Notepad Code:

```
2   ; Q-(1) - Store the data byte 25H into memory location 3000H
3
4   ; Add 25h [37 in Decimal] to Accumulator [Initially Empty]
5   ADI 25h
6
7   ; Store the Result of Accumuator at Location 3000h
8   STA 3000h
9
10  ; Result : We will Find 37 [25h] At Location 3000h
11
12  ;Interrupt
13  hlt
```

Registers

A	25	
BC	00	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	06
SP	FF	FF
Int-Reg	00	

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

Start 3000h OK

Address (Hex)	Address	Data
3000	12288	37
3001	12289	0
3002	12290	0
3003	12291	0
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0

Line No | Assembler Message

(2) Write a program to add two 8-bit numbers. Store the result at one memory location.

Notepad Code:

```
; Direct Method

; Intiatilize the Register B with 8-bit [Number 1]
MVI B, 01h
; Intiatilize the Register D with 8-bit [Number 2]
MVI D, 01h
; Add B to Accumulator
ADC B
; Add D to Accumulator
ADC D
; Store Result from Accumulator at Location 3004h
STA 3004h
```

Registers

A	02	
BC	01	00
DE	01	00
HL	00	00
PSW	00	00
PC	42	0A
SP	FF	FF
Int-Reg	00	

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

Start 3004h OK

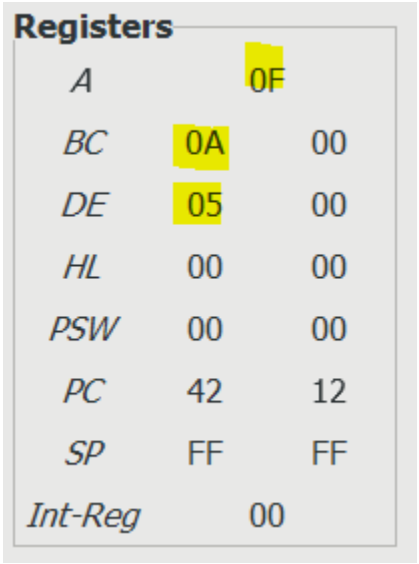
Address (Hex)	Address	Data
3004	12292	2
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0
300E	12302	0
300F	12303	0
3010	12304	0
3011	12305	0

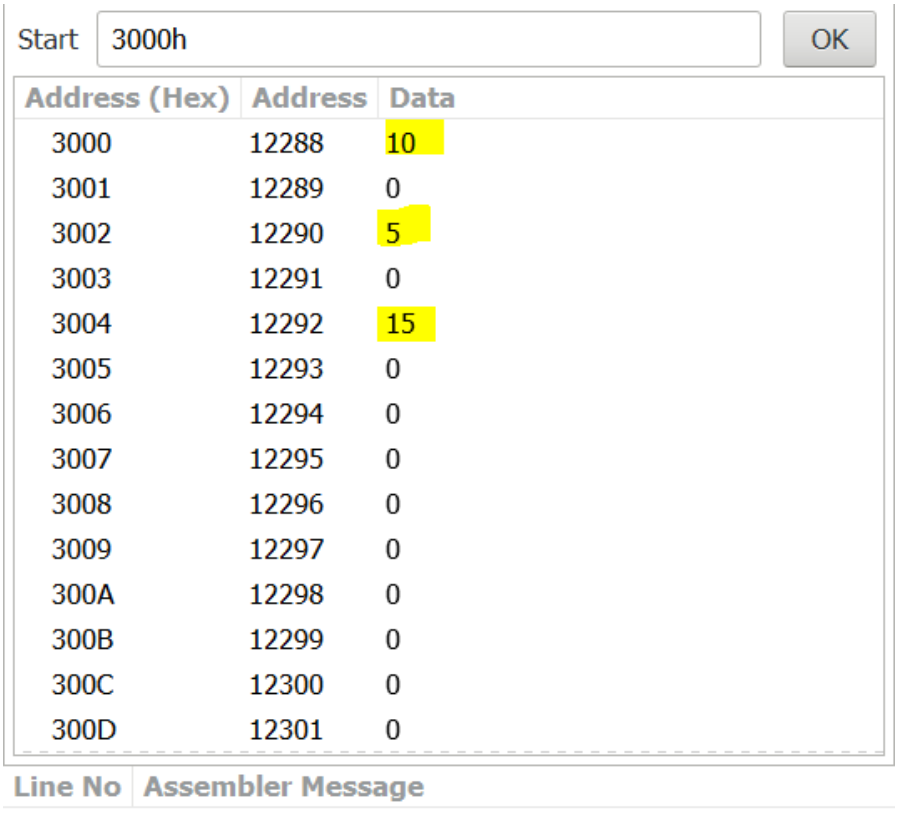
Line No | Assembler Message

Notepad Code:

```
5   ; Method (2) : Data from Register at Address 3000h & 3002h
6
7   MVI A,00
8   ; Intiatilize the Register B with 8-bit [Number 1] at Location 3000h
9   LDA 3000h
10  MOV B, A
11  ; Intiatilize the Register D with 8-bit [Number 2] at Location 3002h
12  LDA 3002h
13  MOV D, A
14  ; Reset the Accumulator
15  MVI A,00
16  ; Add B to Accumulator
17  ADC B
18  ; Add D to Accumulator
19  ADC D
20  ; Store Result from Accumulator at Location 3004h
21  STA 3004h
22
23  hlt
```

Let's Input B = **10** [(0A)h] & D = **5** [(05)h] -> Output : **15** [(0F)h]





(3) Write a program to subtract two 8-bit numbers. Store the result at one memory location.

Notepad Code:

```
; Data from Register at Address 3000h & 3002h

MVI A,00
; Intiatilize the Register B with 8-bit [Number 1] at Location 3000h
LDA 3000h
MOV B, A
; Intiatilize the Register D with 8-bit [Number 2] at Location 3002h
LDA 3002h
MOV D, A
; Reset the Accumulator
MVI A,00
; Add B to Accumulator
ADD B
; Sub D to Accumulator
SUB D
; Store Result [B-D] from Accumulator at Location 3004h
STA 3004h

hlt
```

Let's Input B = **10** [(0A)h] & D = **5** [(05)h] -> Output : **5** [(05)h]

Registers

A	05	
BC	0A	00
DE	05	00
HL	00	00
PSW	00	00
PC	42	12
SP	FF	FF
Int-Reg	00	

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

Start 3000h OK

Address (Hex)	Address	Data
3000	12288	10
3001	12289	0
3002	12290	5
3003	12291	0
3004	12292	5
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0

Line No	Assembler Message
0	Program assembled successfully

Let's Input B = **5** [(05)h] & D = **10** [(0A)h] -> Output : **5-10 = -5 = -5+256 = 251** [(FB)h]

Registers

A	FB	
BC	05	00
DE	0A	00
HL	00	00
PSW	00	00
PC	42	12
SP	FF	FF
Int-Reg	00	

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

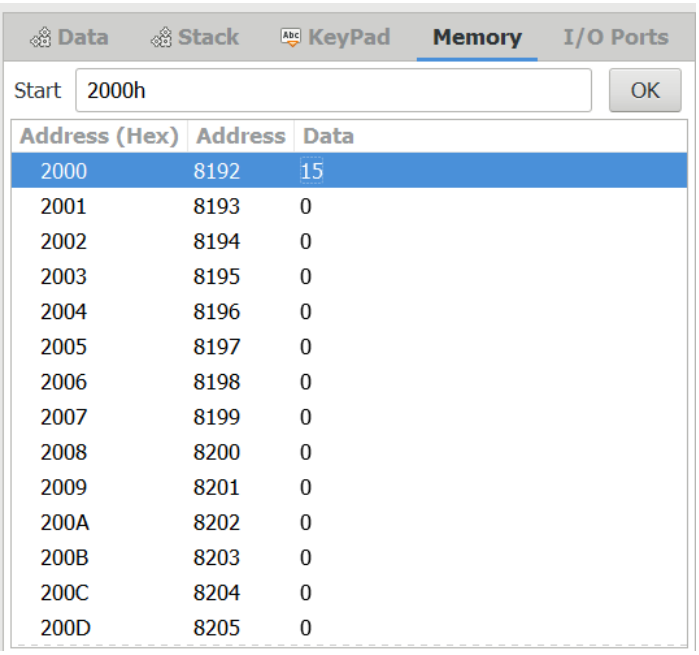
Start 3000h OK

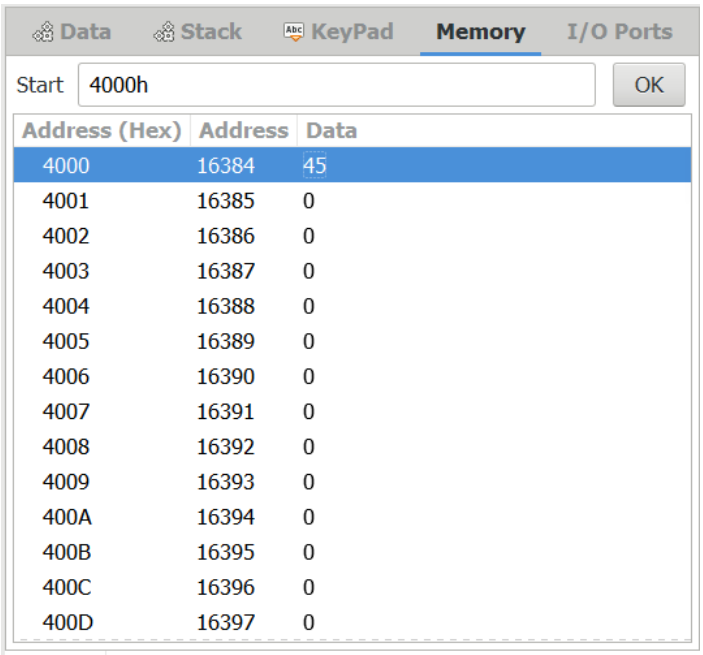
Address (Hex)	Address	Data
3000	12288	5
3001	12289	0
3002	12290	10
3003	12291	0
3004	12292	251
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0

(4) Exchange the contents of memory locations 2000H and 4000H

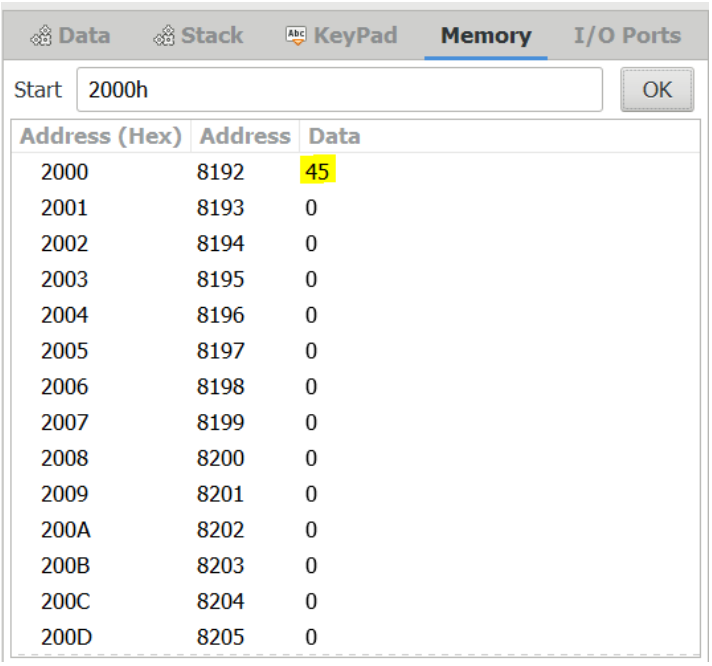
```
3  ; Intialize Accumulator to 00
4  MVI A,00
5  ; Store the Contents at Location 2000H in B register [Temporary Var]
6  LDA 2000h
7  MOV B, A
8  ; Store the Contents at Location 4000H at Location 2000H
9  LDA 4000h
10 STA 2000h
11 ; Store the Contents from B Register [Temp] to Location 4000H
12 MVI A, 00
13 ADD B
14 STA 4000h
15
16 hlt
```

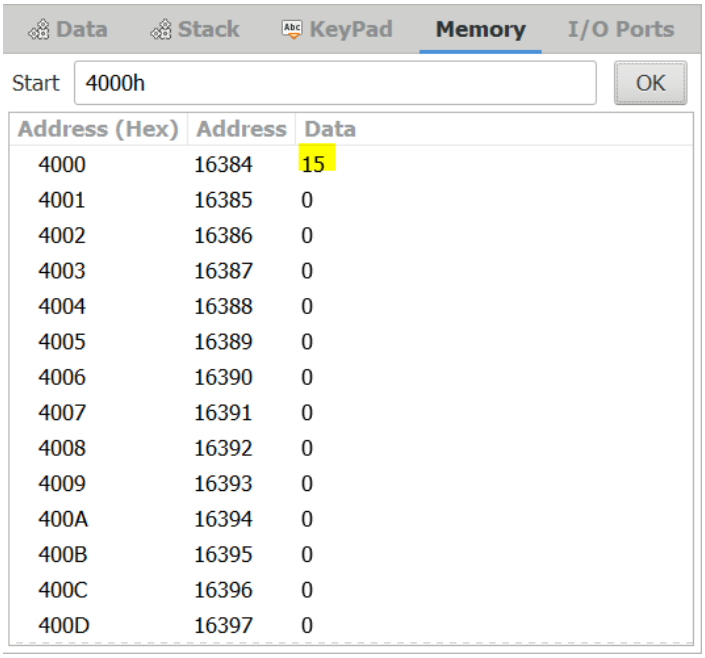
INPUT: 2000h = 15 & 4000h = 45





OUTPUT: 2000h = 45 & 4000h = 15





(5) Write a program to add two 16 bit numbers. Numbers are stored in four consecutive memory location as 8-bit numbers. (Use instruction ADC)

```
3   MVI A,00
4   ; Intiatilize the Register B with 16-bit [Number 1](lsb) at Location 3000h
5   LDA 3000h
6   MOV B, A
7   ; Intiatilize the Register C with 16-bit [Number 1](msb) at Location 3001h
8   LDA 3001h
9   MOV C, A
10  ; Intiatilize the Register D with 16-bit [Number 2](lsb) at Location 3002h
11  LDA 3002h
12  MOV D, A
13  ; Intiatilize the Register E with 16-bit [Number 2](msb) at Location 3003h
14  LDA 3003h
15  MOV E, A
16  ; Reset Accumulator
17  MVI A,00
18  ; Add (lsb) of Two Numbers
19  ADC B
20  ADC D
21  ; Store the LSB Result at location 3005h
22  STA 3005h
23
24  ; Reset Accumulator
25  MVI A,00
26  ; Add (msb) of Two Numbers
27  ADC C
28  ADC E
29  ; Store the MSB Result at location 3005h
30  STA 3006h
31
32  hlt
```

Let's Suppose

Number 1 = (msb) **10** 10 (lsb) [**4112** in Decimal]

Input 1: Addr(3000h) = 10h = 16 && Addr(3001h) = **10**h = 16

Number 2 = (msb) 01 10 (lsb) [**272** in Decimal]

Input 2: Addr(3002h) = 10h = 16 && Addr(3003h) = **01**h = 1

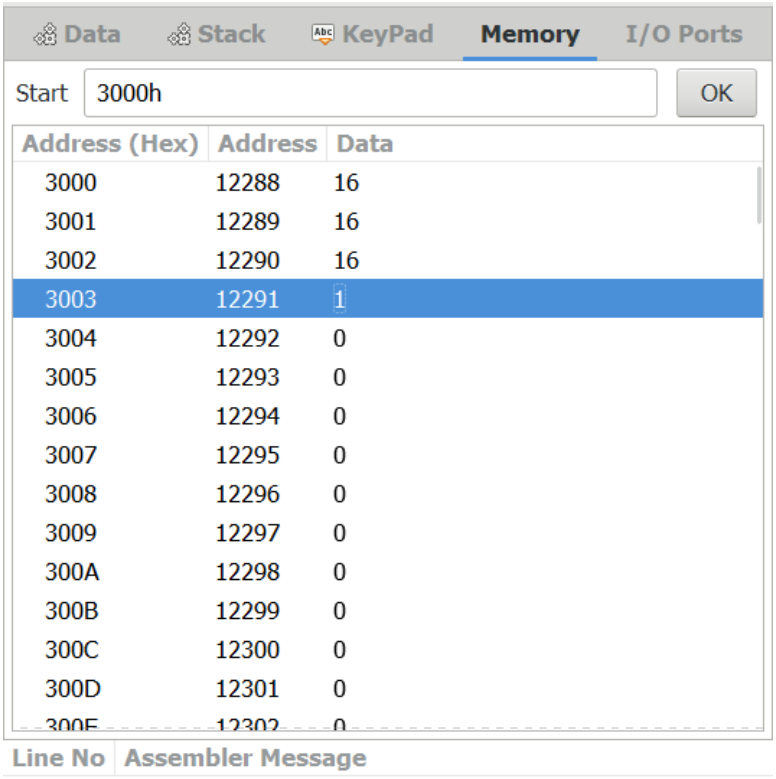
Expected Output = 4112 + 272 = **4384** = (msb) **11** 20 (lsb) [Hexadecimal]

Expected Output:

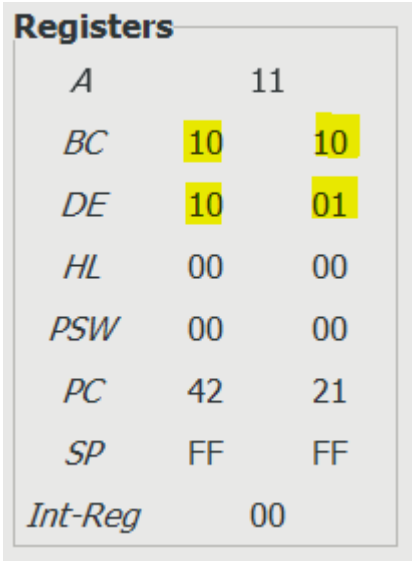
(LSB)Addr(3005h) = 20h = 32

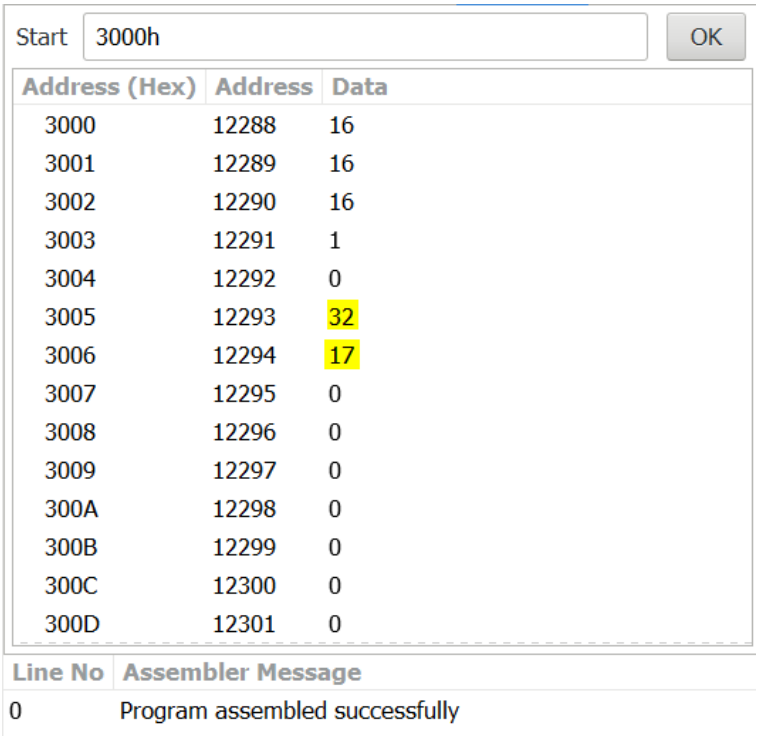
(MSB)Addr(3006h) = **11h** = 17

INPUT :



OUTPUT:





(6) Write a program to subtract two 16 bit numbers. Numbers are stored in four consecutive memory location as 8-bit numbers. (Use instruction SBB)

```
;Q-(6) Write a program to subtract two 16 bit numbers.
; Numbers are stored in four consecutive memory location as 8-bit numbers. (Use instruction SBB)
MVI A,00
; Intiatilize the Register B with 16-bit [Number 1](lsb) at Location 3000h
LDA 3000h
MOV B, A
; Intiatilize the Register C with 16-bit [Number 1](msb) at Location 3001h
LDA 3001h
MOV C, A
; Intiatilize the Register D with 16-bit [Number 2](lsb) at Location 3002h
LDA 3002h
MOV D, A
; Intiatilize the Register E with 16-bit [Number 2](msb) at Location 3003h
LDA 3003h
MOV E, A

; Subtract (lsb) of Two Numbers
MOV A,B ; A <- B
SBB D ; A <- A-D
; Store the LSB Result at location 3005h
STA 3005h

MVI A, 00
; Subtract (msb) of Two Numbers
MOV A,C ; A <- B
SBB E ; Subtract with Borrow
; Store the MSB Result at location 3005h
STA 3006h
hlt
```

Let's Suppose

Number 1 = (msb) **10** 10 (lsb) [**4112** in Decimal]

Input 1: Addr(3000h) = 10h = 16 && Addr(3001h) = **10**h = 16

Number 2 = (msb) 01 10 (lsb) [**272** in Decimal]

Input 2: Addr(3002h) = 10h = 16 && Addr(3003h) = **01**h = 1

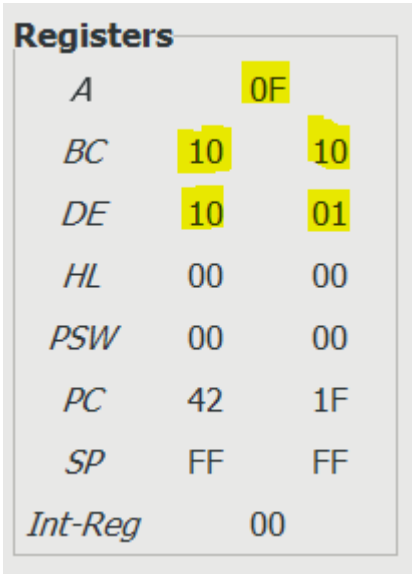
Expected Output = 4112 - 272 = **3840** = (msb) **0F** 00 (lsb) [Hexadecimal]

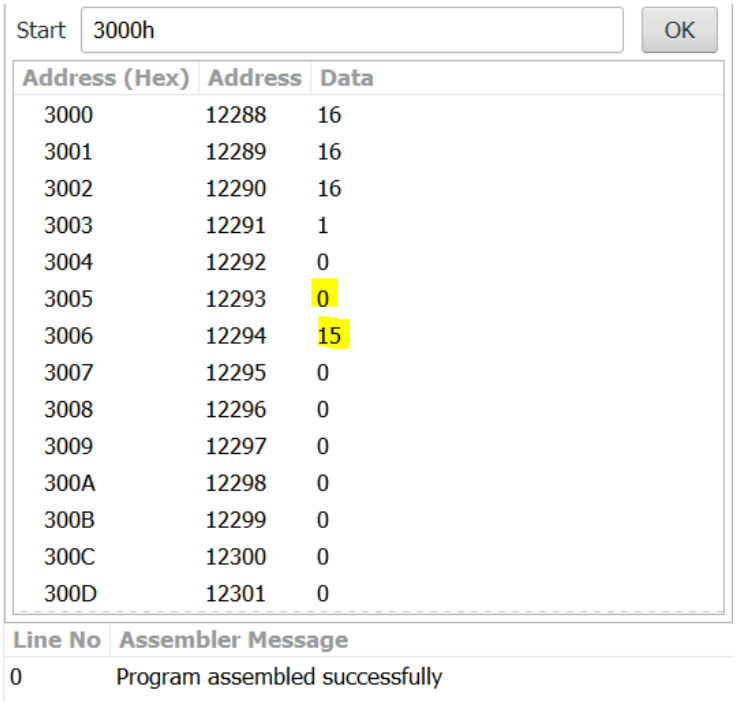
Expected Output:

(LSB)Addr(3005h) = 00h = 0

(MSB)Addr(3006h) = **0Fh** = 15

OUTPUT:





Let's Suppose

Number 1 = (msb) **10** 10 (lsb) [**4112** in Decimal]

Input 1: Addr(3000h) = 10h = 16 && Addr(3001h) = **10**h = 16

Number 2 = (msb) 01 11 (lsb) [**273** in Decimal]

Input 2: Addr(3002h) = 10h = 1 && Addr(3003h) = **01**h = 17

Expected Output = 4112 - 273 = **3839** = (msb) **0E** FF (lsb) [Hexadecimal]

Expected Output:

(LSB)Addr(3005h) = 0Eh = 14

(MSB)Addr(3006h) = **FFh** = 255

Registers

A	FF	
BC	10	10
DE	01	11
HL	00	00
PSW	00	00
PC	42	1F
SP	FF	FF
Int-Reg	00	

Start 3000h OK

Address (Hex)	Address	Data
3000	12288	16
3001	12289	16
3002	12290	1
3003	12291	17
3004	12292	0
3005	12293	14
3006	12294	255
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0
300C	12300	0
300D	12301	0

Line No	Assembler Message
0	Program assembled successfully

Submitted By:

BHAGYA VINOD RANA

U19CS012