19CSE303: Embedded Systems

LABSHEET 2: ASSEMBLY LANGUAGE PROGRAMMING OF 8085 MICROPROCESSORS

Name: Vinayak V Thayil **Roll Number:AM.EN.U4CSE21161**

1. ONE'S COMPLEMENT OF AN 8-BIT NUMBER

```
LDA 5100H ; Load Address Of Number In H-L Register Pair (Lower byte data).
CMA       ; Complement Accumulator
STA 5101H ; Store The Result
HLT       ; Terminate Program
```

INPUT		OUTPUT	
Memory location	Data	Memory location	Data
5100H	56	5101H	199

2. 2'S COMPLEMENT OF AN 8-BIT NUMBER

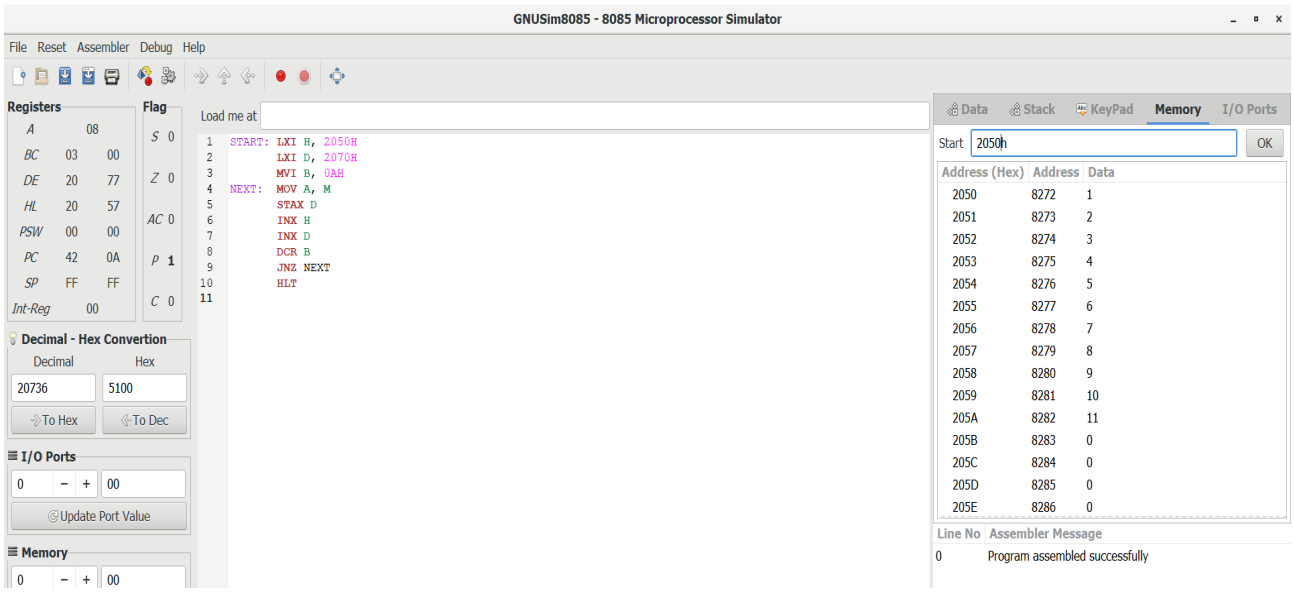
```
LDA 5100H ; Load Address Of Number In H-L Register Pair (Lower byte data).
CMA       ; Complement Accumulator
INR A     ; Increment
STA 5101H ; Store The Result
HLT       ; Terminate Program
```

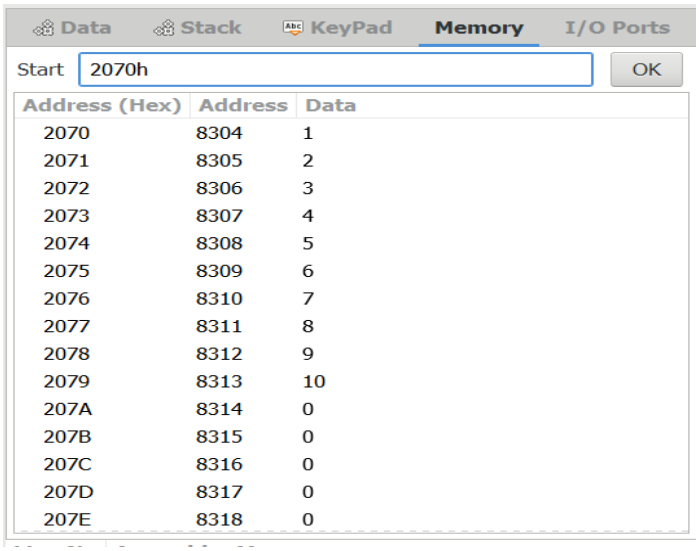
INPUT		OUTPUT	
Memory location	Data	Memory location	Data
5100H	56	5101H	200

3. Show the output in 8085 simulator for the following programs. Attach the screenshot of final status of PC and relevant registers.

   a) Copy 10 numbers stored from the location 2050H to the new location 2070H

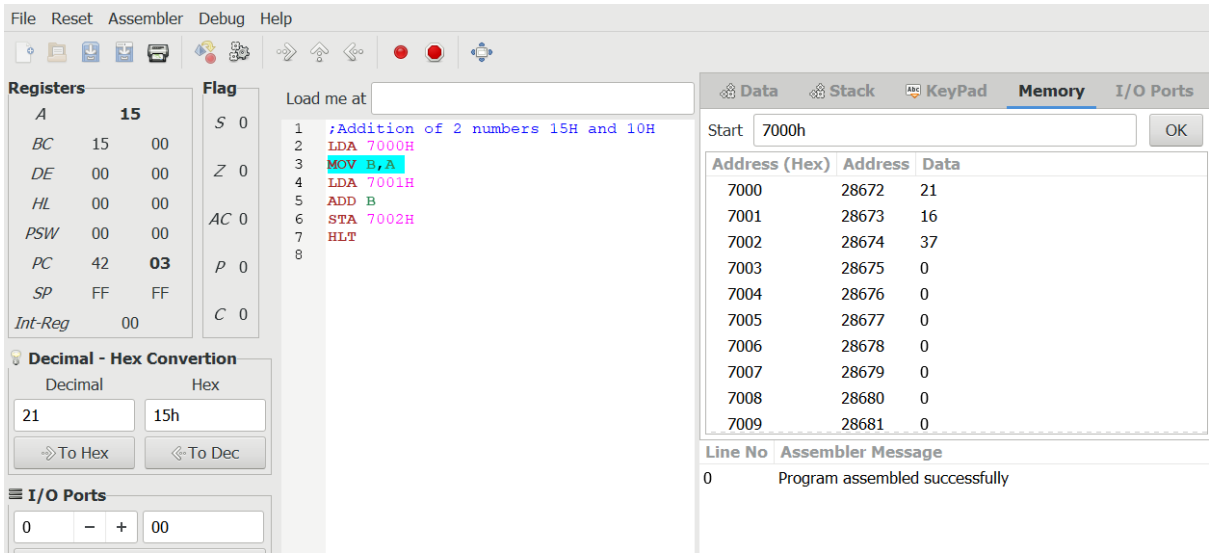
```
START: LXI H, 2050H
       LXI D, 2070H
       MVI B, 0AH
NEXT:  MOV A, M
       STAX D
       INX H
       INX D
       DCR B
       JNZ NEXT
       HLT
```





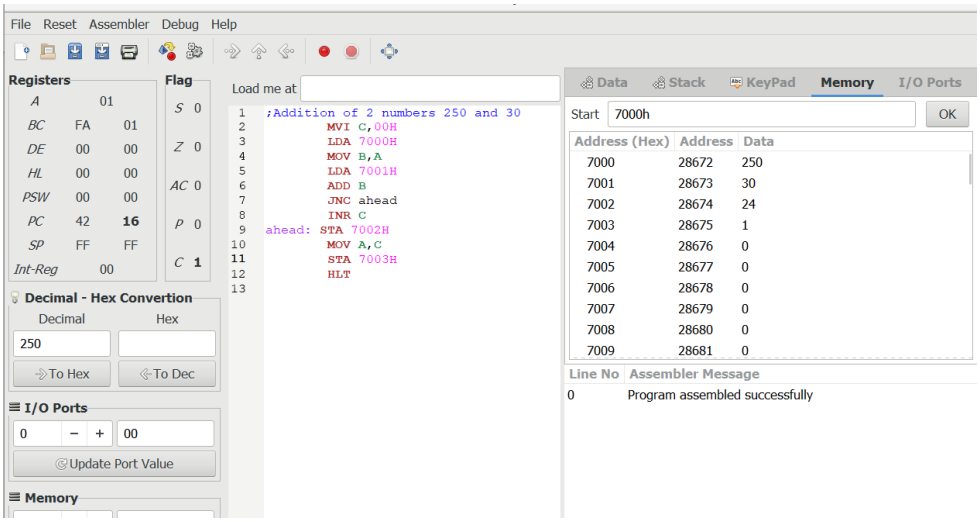
b) 8-bit Addition without carry

```
;Addition of 2 numbers 15H and 10H
LDA 7000H
MOV B,A
LDA 7001H
ADD B
STA 7002H
HLT
```



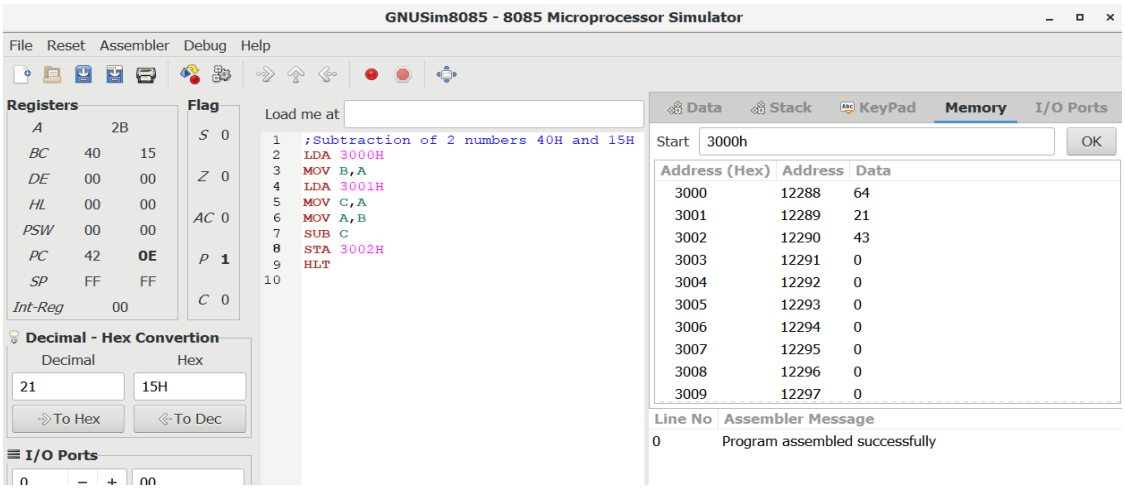
c) 8-bit Addition with carry

```
;Addition of 2 numbers 250 and 30
        MVI C,00H
        LDA 7000H
        MOV B,A
        LDA 7001H
        ADD B
        JNC ahead
        INR C
ahead: STA 7002H
        MOV A,C
        STA 7003H
        HLT
```



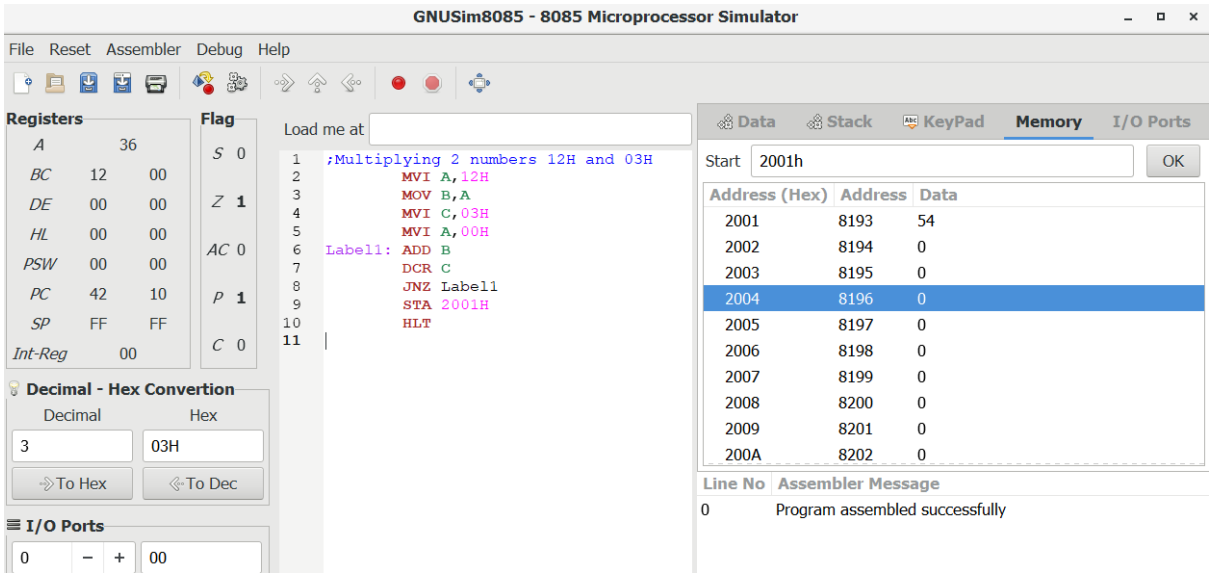
d) 8-bit Subtraction

```
;Subtraction of 2 numbers 40H and 15H
LDA 3000H
MOV B,A
LDA 3001H
MOV C,A
MOV A,B
SUB C
STA 3002H
HLT
```



e) 8-bit Multiplication

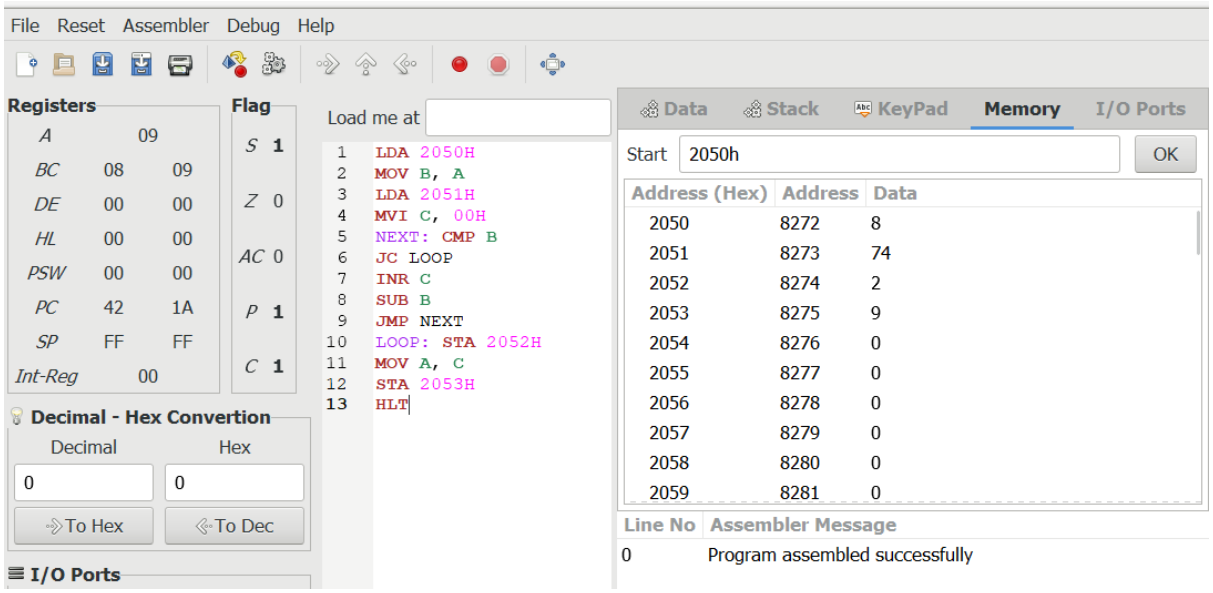
```
;Multiplying 2 numbers 12H and 03H
        MVI A,12H
        MOV B,A
        MVI C,03H
        MVI A,00H
Label1: ADD B
        DCR C
        JNZ Label1
        STA 2001H
        HLT
```




```
;Multiplying 2 numbers 12H and 03H
        MVI A,12H
        MOV B,A
        MVI C,03H
        MVI A,00H
Label1: ADD B
        DCR C
        JNZ Label1
        STA 2001H
        HLT
```

f) 8-bit Division

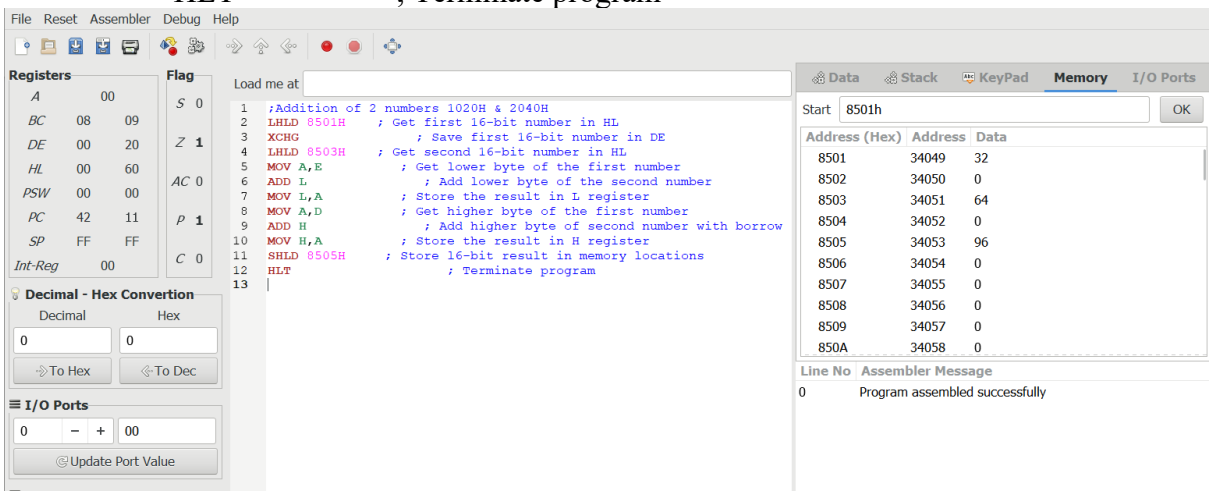
```
;Division of 2 numbers : Dividend -> 4AH & Divisor -> 08H
        LDA 2050H  ; Load Divisor in Accumulator
        MOV B,A    ; Copy Divisor to Register B
        LDA 2051H  ; Load Dividend in Accumulator
        MVI C, 00H ; Initialize register C for Quotient
Next: CMP B        ; Compare the no. in B with the no. in Accumulator
        JC Loop    ; Jump if B>A to Loop
        INR C      ; Increment Register C (Quotient)
        SUB B      ; Subtract B from A
        JMP Next   ; Repeat the above steps till A becomes smaller than B
Loop: STA 2052H    ; Store the remainder at memory address 2052
        MOV A,C    ; Move the contents of C to Accumulator
        STA 2053H  ; Store the Quotient at memory address 2053
        HLT        ; HLT
```




```
LDA 2050H
MOV B, A
LDA 2051H
MVI C, 00H
NEXT: CMP B
JC LOOP
INR C
SUB B
JMP NEXT
LOOP: STA 2052H
MOV A, C
STA 2053H
HLT
```

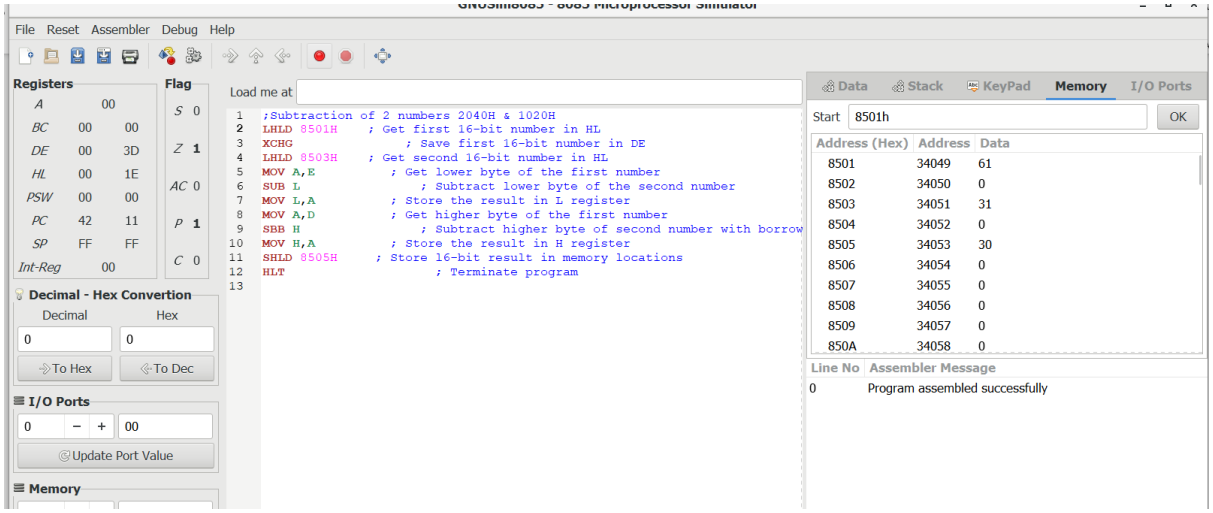
g) 16-bit Addition

```
;Addition of 2 numbers 1020H & 2040H
LHLD 8501H    ; Get first 16-bit number in HL
XCHG          ; Save first 16-bit number in DE
LHLD 8503H    ; Get second 16-bit number in HL
MOV A,E       ; Get lower byte of the first number
ADD L         ; Add lower byte of the second number
MOV L,A       ; Store the result in L register
MOV A,D       ; Get higher byte of the first number
ADD H         ; Add higher byte of second number with borrow
MOV H,A       ; Store the result in H register
SHLD 8505H    ; Store l6-bit result in memory locations
HLT           ; Terminate program
```



h) 16-bit Subtraction

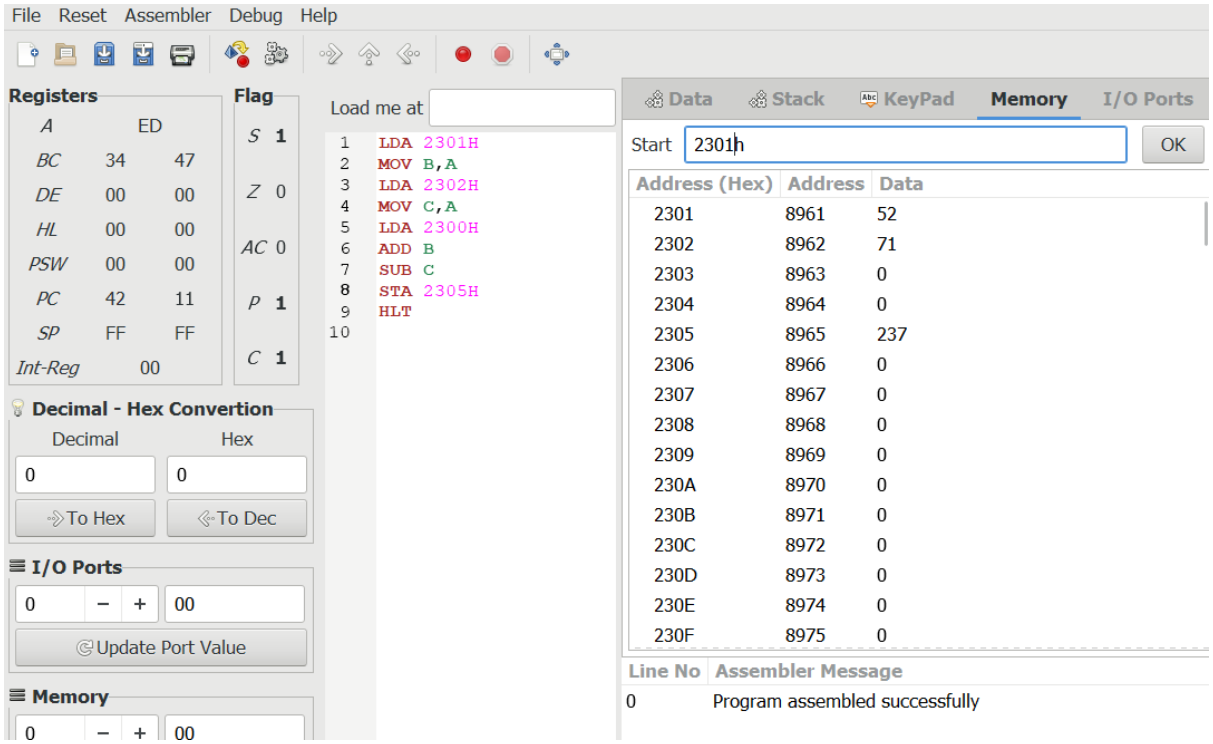
```
;Subtraction of 2 numbers 2040H & 1020H
LHLD 8501H    ; Get first 16-bit number in HL
XCHG          ; Save first 16-bit number in DE
LHLD 8503H    ; Get second 16-bit number in HL
MOV A,E       ; Get lower byte of the first number
SUB L         ; Subtract lower byte of the second number
MOV L,A       ; Store the result in L register
MOV A,D       ; Get higher byte of the first number
SBB H         ; Subtract higher byte of second number with borrow
MOV H,A       ; Store the result in H register
SHLD 8505H    ; Store l6-bit result in memory locations
HLT           ; Terminate program
```



4. Write an assembly program in mnemonic code for 8085 processor to calculate

(A+B)-C where the value of

- A stored the value 34H at address 2300H
- B stored the value 47H at address 2301H
- C stored the value ABH at address 2302H



5. Write an ALP (Assembly Level Language) to add 10 numbers stored in the consecutive memory locations starting from 2000H. Write the code using conditional jump instruction. Store the final result in 200AH.