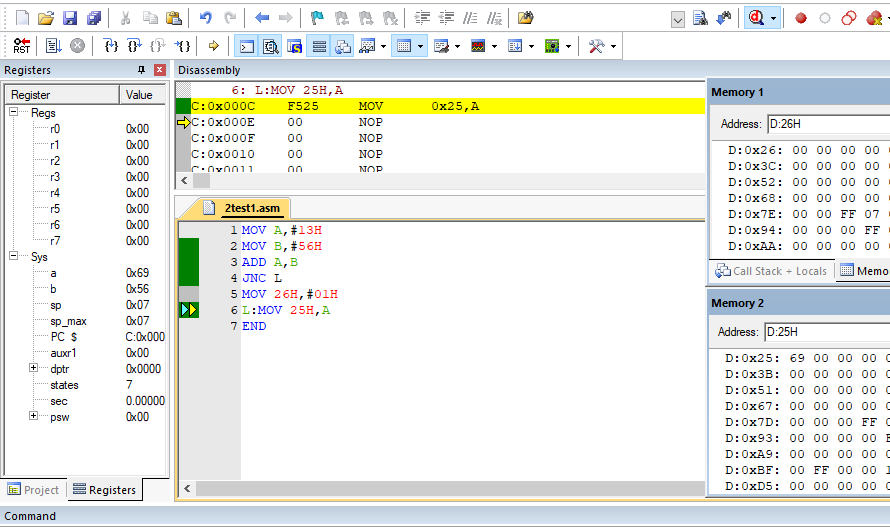
**PART A : 8-bit Arithmetic Operations: Write an 8051 Assembly program to do the following tasks**

**Add the given two 8-bit numbers <<D1>> and <<D2>> (With / Without carry). Then store the sum in the Internal RAM location 25h. Store the final carry, if any in the location 26h.**



WITHOUT CARRY:

MOV A,#13H

MOV B,#56H

ADD A,B

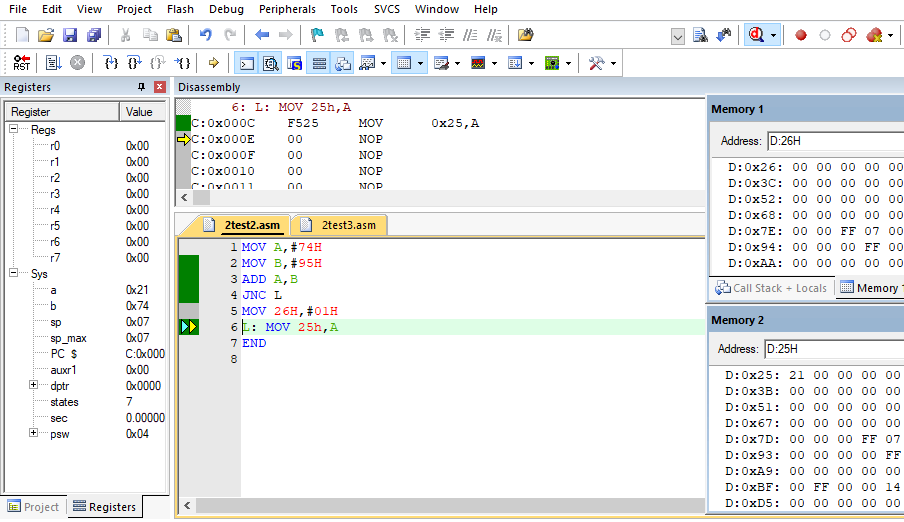
JNC L

MOV 26H,#01H

L: MOV 25h,A

END

WITH CARRY :

MOV A,#74H

MOV B,#95H

ADD A,B

JNC L

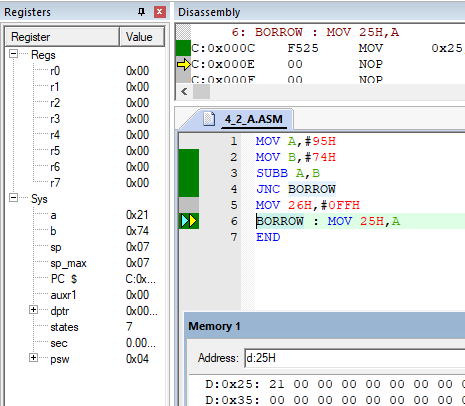
MOV 26H,#01H

L: MOV 25h,A

END

**Subtract the given 8-bit number <<D1>> from another 8-bit number <<D2>> (With / Without borrow) and store the difference in the Internal RAM location 25h.**

WITHOUT BORROW : **Output**



MOV A,#95H

MOV B,#74H

SUBB A,B

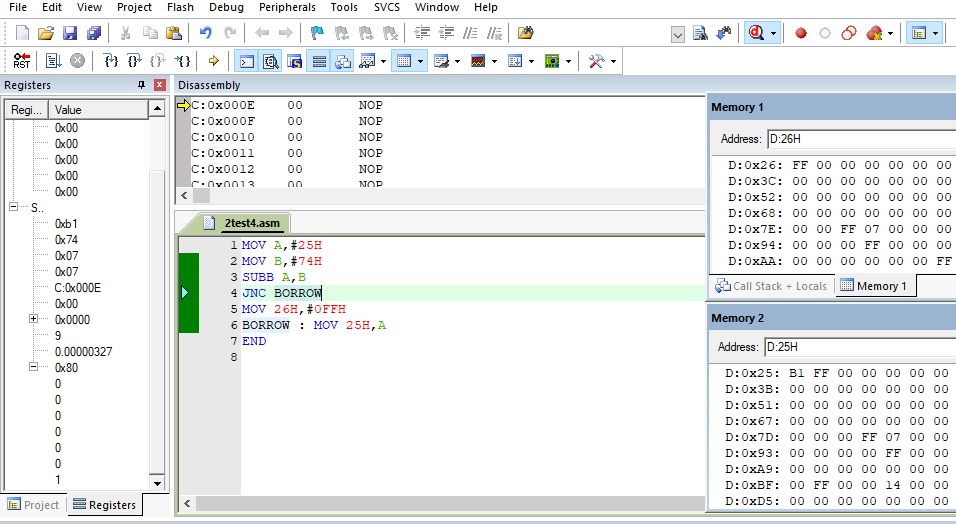
JNC BORROW

MOV 26H,#0FFH

BORROW : MOV 25H,A

END

WITH BORROW : **Output**



MOV A,#25H

MOV B,#74H

SUBB A,B

JNC BORROW

MOV 26H,#0FFH

BORROW : MOV 25H,A

END

**Multiply the given two 8-bit numbers <<D1>> and <<D2>> and then store the product into Internal RAM locations 50h (LB) and 51h (UB).**

MOV A,#25H

MOV B,#31H

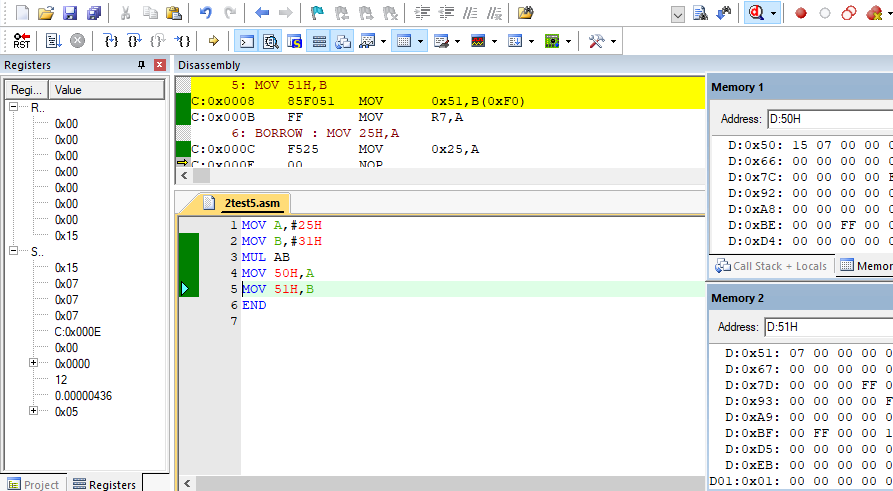
MUL AB

MOV 50H,A

MOV 51H,B

END

**Output**



**Divide the given 8-bit numbers <<D1>> by another 8-bit number <<D2>> and then store the quotient into Internal RAM location 60h and the reminder in 61h.**

MOV A,#31H

MOV B,#25H

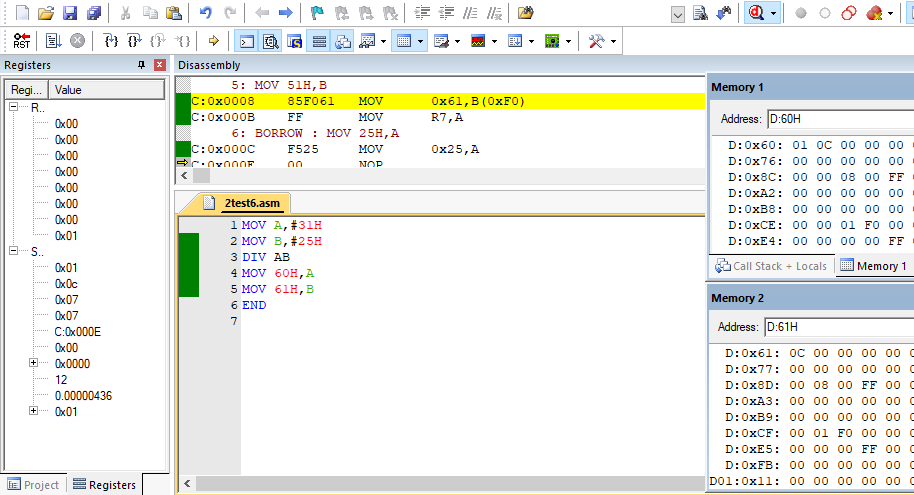
DIV AB

MOV 60H,A

MOV 61H,B

END

**Output**



**Square the given number <<D>>. Store the result in 30h and 31h.**

MOV A,#5H

MOV B,A

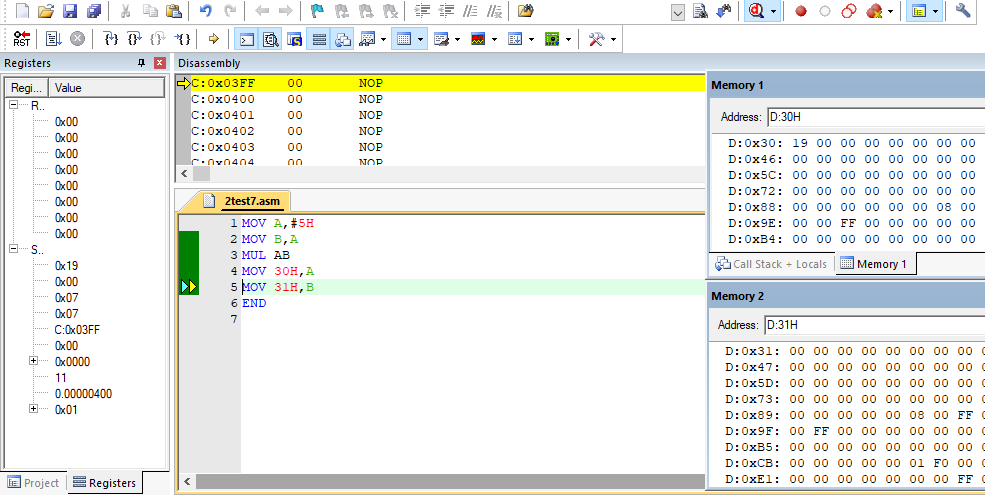
MUL AB

MOV 30H,A

MOV 31H,B

END

**Output**



**Cube the given 8-bit number <<D>>. Store the result in 40h and 41h.**

MOV A,#27H

MOV R0,A

MOV B,A

MUL AB

MOV R1,A

MOV R2,B

MOV A,R0

MOV B,R1

MUL AB

MOV R4,A

MOV R5,B

MOV A,R0

MOV B,R2

MUL AB

MOV R6,A

MOV R7,B

MOV A,R4

MOV 40H,A

MOV A,R5

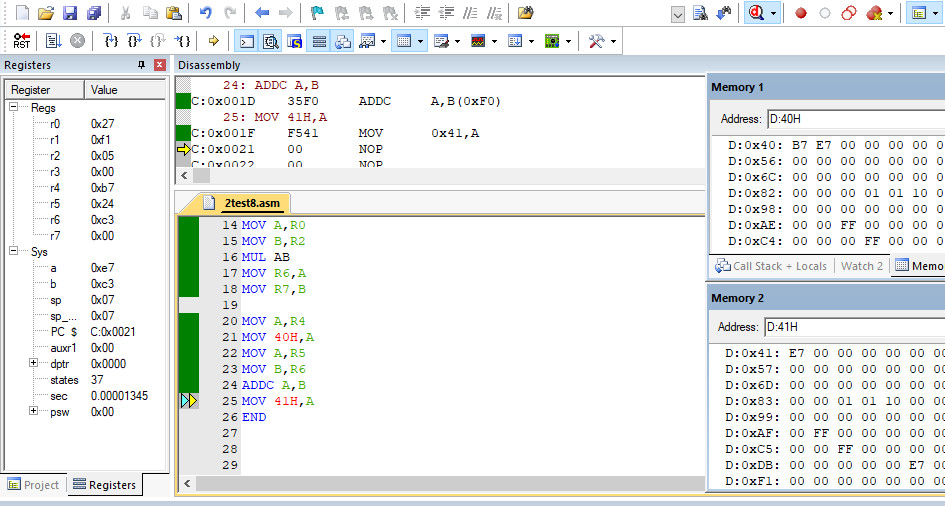
MOV B,R6

ADDC A,B

MOV 41H,A

END

**Output**



**PART B : 16-bit Arithmetic Operations: Write an 8051 Assembly program to do the following tasks**

**Add two 16-bit number <<D1>> and <<D2>> . Store the 16-bit Sum into 31h and 32h.**

;1234H + 5678H

MOV A,#34H

MOV R1,#78H

ADD A,R1

MOV 31H,A

MOV A,#12H

MOV R1,#56H

ADDC A,R1

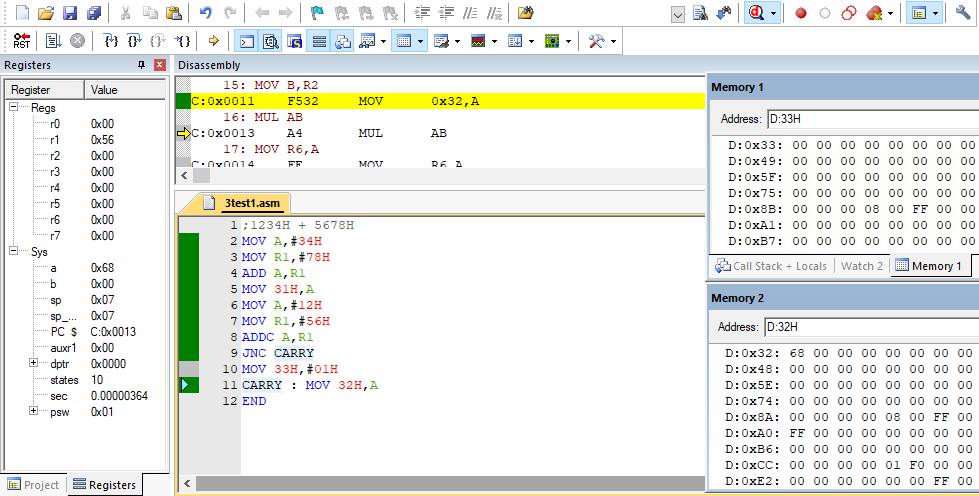
JNC CARRY

MOV 33H,#01H

CARRY : MOV 32H,A

END

**Output**



**Subtract two 16-bit number: <<D1>> from <<D2>> . Store the 16-bit Sum into 31h and 32h.**

;5871H - 1234H

MOV A,#71H

MOV R1,#34H

SUBB A,R1

MOV 31H,A

MOV A,#58H

MOV R1,#12H

SUBB A,R1

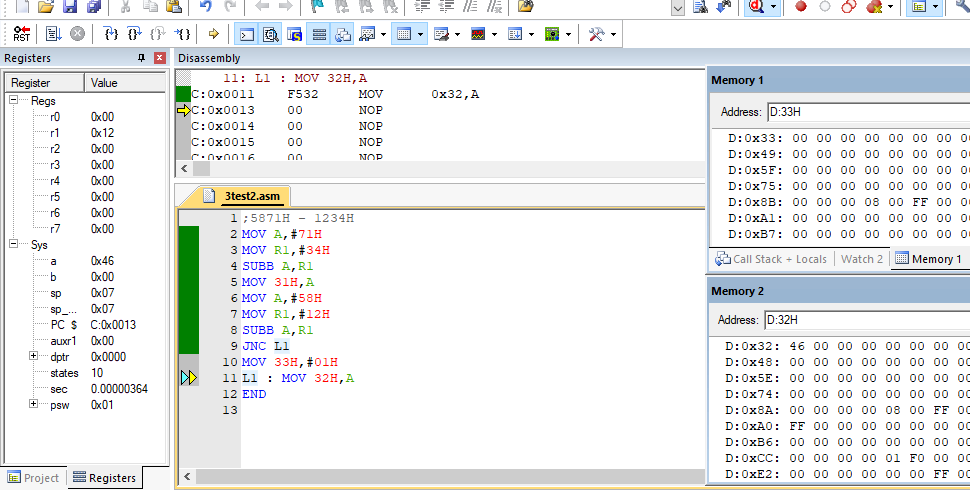
JNC L1

MOV 33H,#01H

L1 : MOV 32H,A

END

**Output**



**Multiplication of two 16-bit numbers <<D1>> and <<D2>>.**

;3456H \* 17H

MOV R0,#34H

MOV R1,#56H

MOV R2,#17H

MOV A,R1

MOV B,R2

MUL AB

MOV 31H,A

MOV R4,B

MOV A,R0

MOV B,R2

MUL AB

MOV R5,B

ADD A,R4

MOV 32H,A

MOV A,B

ADDC A,#00H

MOV 33H,A

END

**Output**

