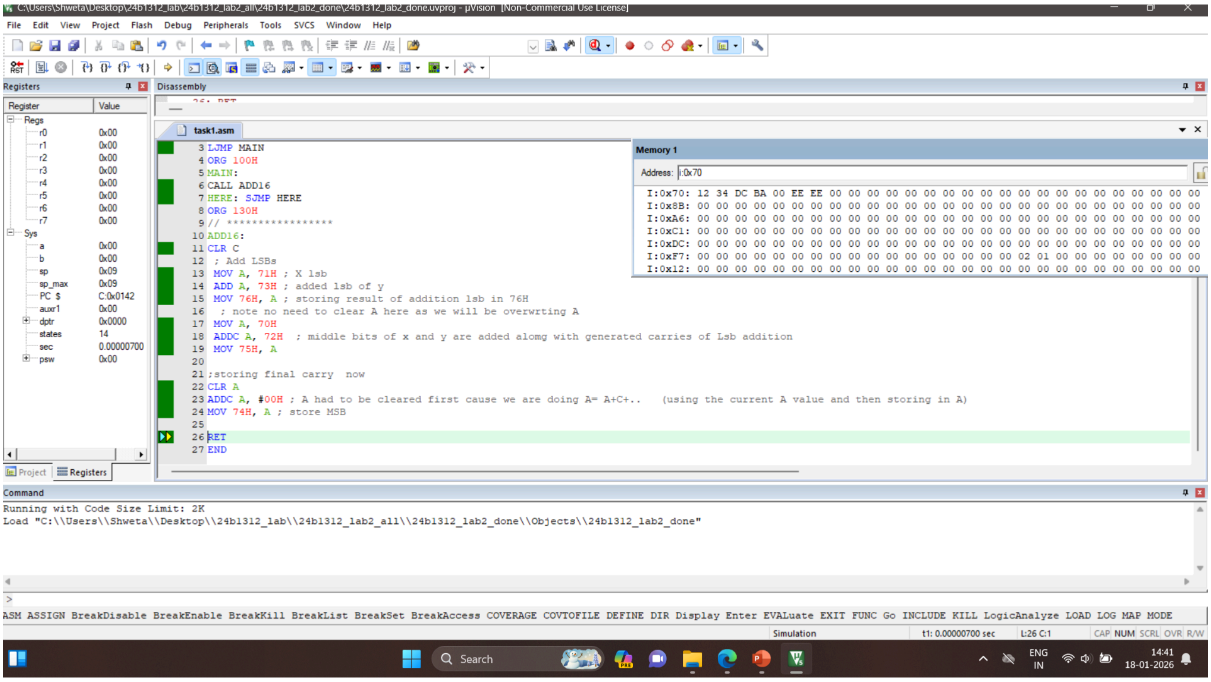
24B1312 , Supriya Anand Mishra
Microprocessors Lab 2
Electrical Eng, IITB

Q1)



C:\Users\Shweta\Desktop\24b1312_lab\24b1312_lab2_all\24b1312_lab2_done\24b1312_lab2_done.uvproj - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
task1.asm
Memory 1

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x09
sp_max	0x09
PC $	C:0x0142
auxr1	0x00
dptr	0x0000
states	14
sec	0.00000700
psw	0x00

```
LJMP MAIN
ORG 100H
MAIN:
CALL ADD16
HERE: SJMP HERE
ORG 130H
// *****************
ADD16:
CLR C
 ; Add LSBs
 MOV A, 71H ; X lsb
 ADD A, 73H ; added lsb of y
 MOV 76H, A ; storing result of addition lsb in 76H
  ; note no need to clear A here as we will be overwrting A
 MOV A, 70H
 ADDC A, 72H  ; middle bits of x and y are added alomg with generated carries of Lsb addition
 MOV 75H, A

;storing final carry  now
CLR A
ADDC A, #00H ; A had to be cleared first cause we are doing A= A+C+..   (using the current A value and then storing in A)
MOV 74H, A ; store MSB

RET
END
```

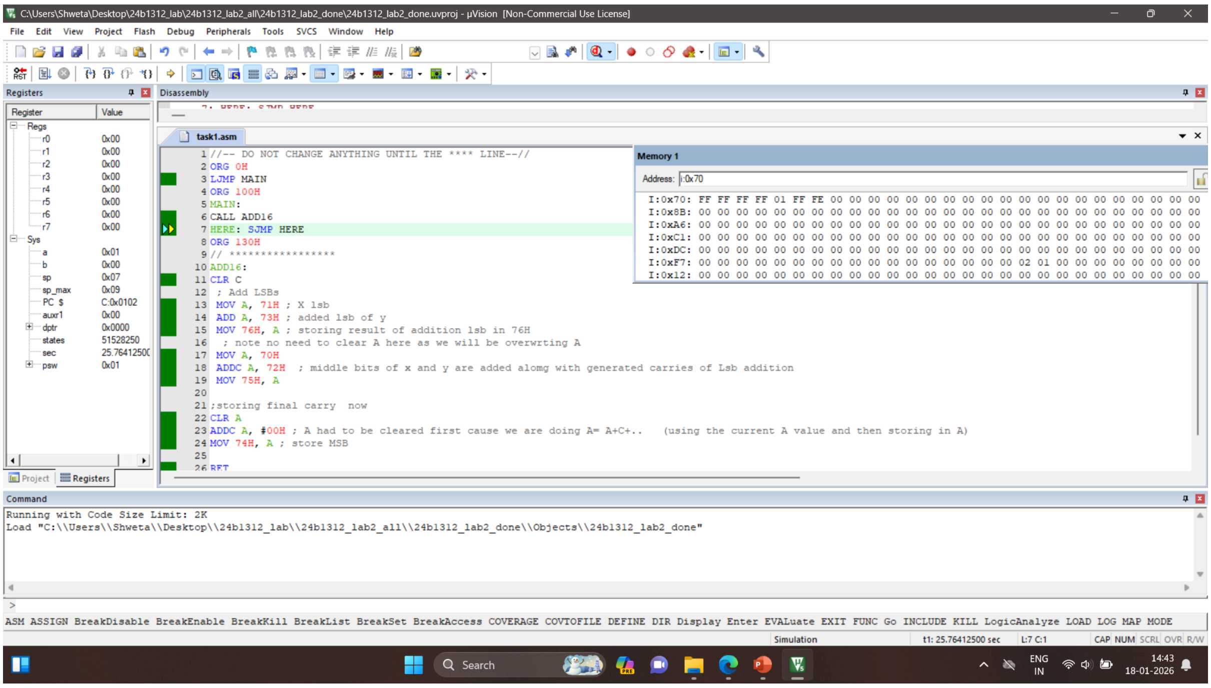
Address: i:0x70

```
I:0x70: 12 34 DC BA 00 EE EE 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x8B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xA6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xC1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xDC: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xF7: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 01 00 00 00 00 00 00 00 00
I:0x12: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Command

```
Running with Code Size Limit: 2K
Load "C:\\Users\\Shweta\\Desktop\\24b1312_lab\\24b1312_lab2_all\\24b1312_lab2_done\\Objects\\24b1312_lab2_done"
```

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE KILL LogicAnalyze LOAD LOG MAP MODE
Simulation
t1: 0.00000700 sec
L:26 C:1
CAP NUM SCRL OVR R/W
Search
ENG IN
14:41
18-01-2026



C:\Users\Shweta\Desktop\24b1312_lab\24b1312_lab2_all\24b1312_lab2_done\24b1312_lab2_done.uvproj - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
task1.asm
Memory 1

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x01
b	0x00
sp	0x07
sp_max	0x09
PC $	C:0x0102
auxr1	0x00
dptr	0x0000
states	51528250
sec	25.76412500
psw	0x01

```
//-- DO NOT CHANGE ANYTHING UNTIL THE **** LINE--//
ORG 0H
LJMP MAIN
ORG 100H
MAIN:
CALL ADD16
HERE: SJMP HERE
ORG 130H
// *****************
ADD16:
CLR C
 ; Add LSBs
 MOV A, 71H ; X lsb
 ADD A, 73H ; added lsb of y
 MOV 76H, A ; storing result of addition lsb in 76H
  ; note no need to clear A here as we will be overwrting A
 MOV A, 70H
 ADDC A, 72H  ; middle bits of x and y are added alomg with generated carries of Lsb addition
 MOV 75H, A

;storing final carry  now
CLR A
ADDC A, #00H ; A had to be cleared first cause we are doing A= A+C+..   (using the current A value and then storing in A)
MOV 74H, A ; store MSB

RET
```

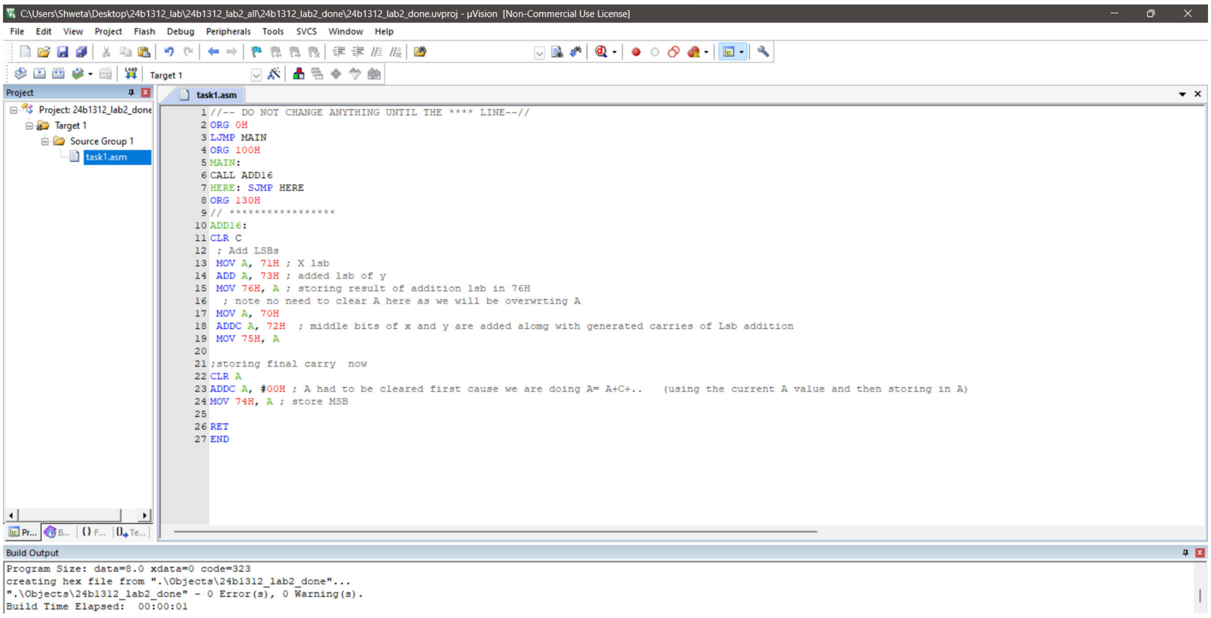
Address: i:0x70

```
I:0x70: FF FF FF FF 01 FF FE 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x8B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xA6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xC1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xDC: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xF7: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 01 00 00 00 00 00 00 00 00
I:0x12: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Command

```
Running with Code Size Limit: 2K
Load "C:\\Users\\Shweta\\Desktop\\24b1312_lab\\24b1312_lab2_all\\24b1312_lab2_done\\Objects\\24b1312_lab2_done"
```

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE KILL LogicAnalyze LOAD LOG MAP MODE
Simulation
t1: 25.76412500 sec
L:7 C:1
CAP NUM SCRL OVR R/W
Search
ENG IN
14:43
18-01-2026



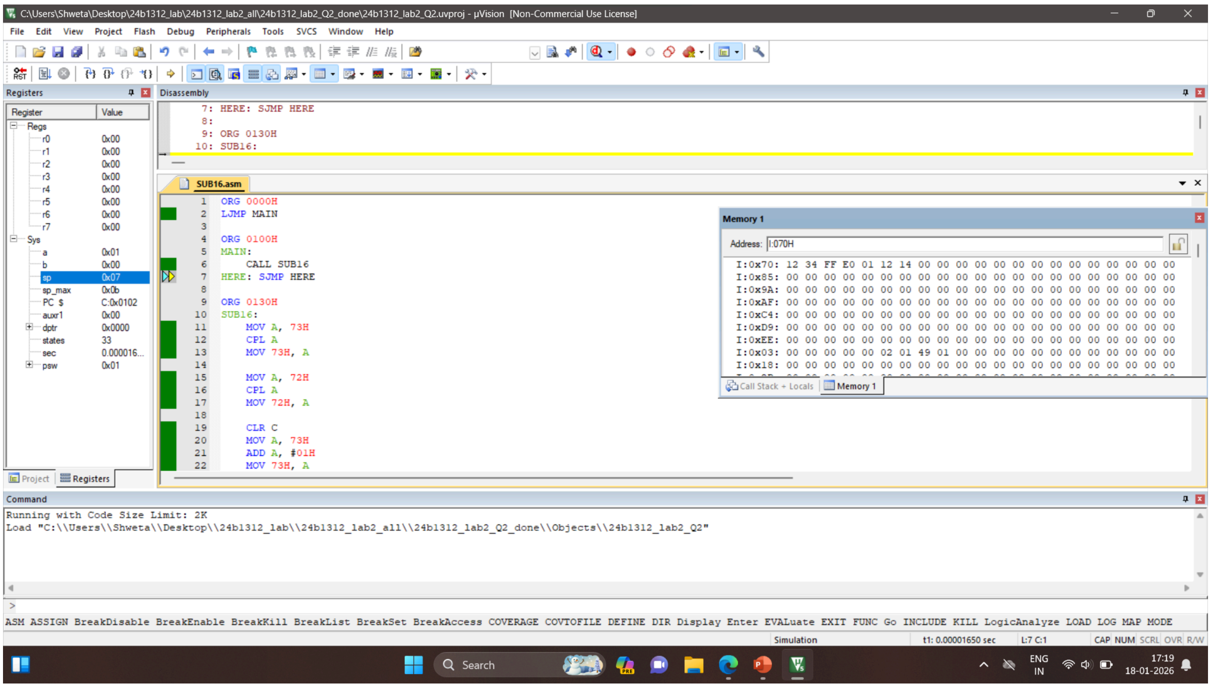
C:\Users\Shweta\Desktop\24b1312_lab\24b1312_lab2_all\24b1312_lab2_done\24b1312_lab2_done.uvproj - µVision [Non-Commercial Use License]
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: 24b1312_lab2_done
Target 1
Source Group 1
task1.asm

```
//-- DO NOT CHANGE ANYTHING UNTIL THE **** LINE--//
ORG 0H
LJMP MAIN
ORG 100H
MAIN:
CALL ADD16
HERE: SJMP HERE
ORG 130H
// *****************
ADD16:
CLR C
 ; Add LSBs
 MOV A, 71H ; X lsb
 ADD A, 73H ; added lsb of y
 MOV 76H, A ; storing result of addition lsb in 76H
  ; note no need to clear A here as we will be overwrting A
 MOV A, 70H
 ADDC A, 72H  ; middle bits of x and y are added alomg with generated carries of Lsb addition
 MOV 75H, A

;storing final carry  now
CLR A
ADDC A, #00H ; A had to be cleared first cause we are doing A= A+C+..   (using the current A value and then storing in A)
MOV 74H, A ; store MSB

RET
END
```

Build Output

```
Program Size: data=8.0 xdata=0 code=323
creating hex file from ".\Objects\24b1312_lab2_done"...
".\Objects\24b1312_lab2_done" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
```

Q2)



C:\Users\Shweta\Desktop\24b1312_lab\24b1312_lab2_all\24b1312_lab2_Q2_done\24b1312_lab2_Q2.uvproj - µVision [Non-Commercial Use License]
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
SUB16.asm
Memory 1
Call Stack + Locals
Project
Registers
Command

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x01
b	0x00
sp	0x07
sp_max	0x0b
PC $	C:0x0102
auxr1	0x00
dptr	0x0000
states	33
sec	0.000016...
psw	0x01

```
     7: HERE: SJMP HERE
     8:
     9: ORG 0130H
    10: SUB16:
```

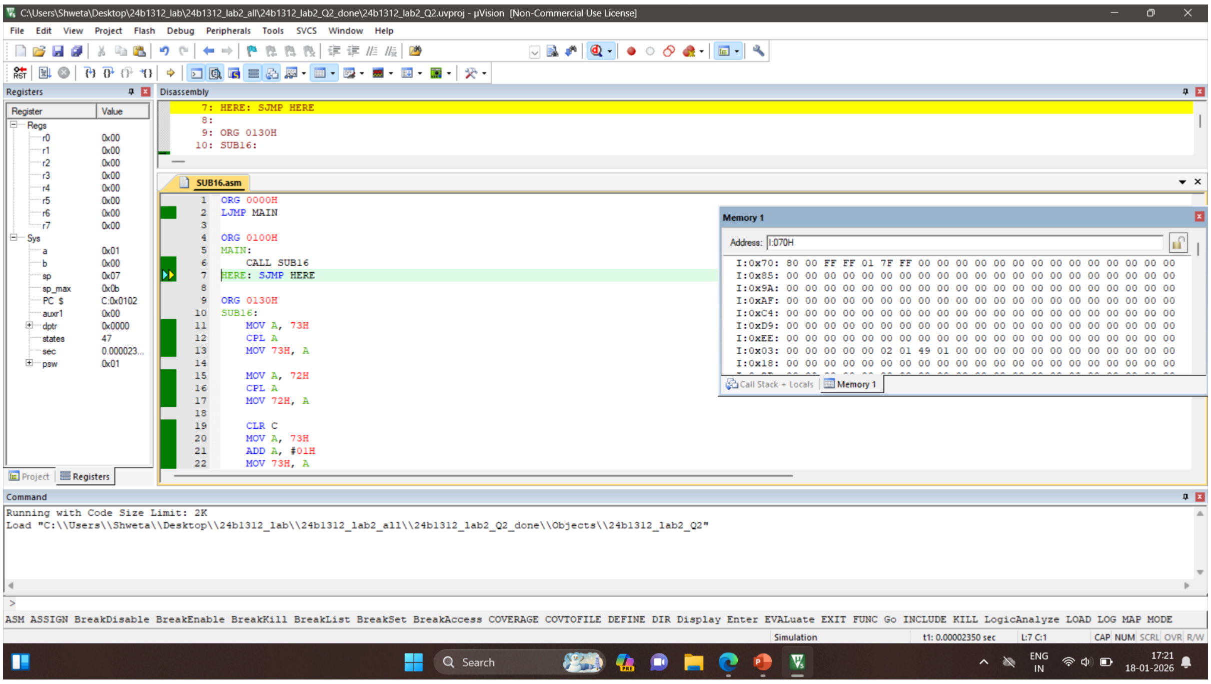
```
ORG 0000H
LJMP MAIN

ORG 0100H
MAIN:
    CALL SUB16
HERE: SJMP HERE

ORG 0130H
SUB16:
    MOV A, 73H
    CPL A
    MOV 73H, A

    MOV A, 72H
    CPL A
    MOV 72H, A

    CLR C
    MOV A, 73H
    ADD A, #01H
    MOV 73H, A
```

Address: I:070H

```
I:0x70: 12 34 FF E0 01 12 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x85: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x9A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xAF: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xC4: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xD9: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xEE: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x03: 00 00 00 00 00 02 01 49 01 00 00 00 00 00 00 00 00 00 00 00 00
I:0x18: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
Running with Code Size Limit: 2K
Load "C:\\Users\\Shweta\\Desktop\\24b1312_lab\\24b1312_lab2_all\\24b1312_lab2_Q2_done\\Objects\\24b1312_lab2_Q2"
```

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE KILL LogicAnalyze LOAD LOG MAP MODE
Simulation
t1: 0.00001650 sec
L:7 C:1
CAP NUM SCRL OVR R/W
Search
ENG IN
17:19
18-01-2026



C:\Users\Shweta\Desktop\24b1312_lab\24b1312_lab2_all\24b1312_lab2_Q2_done\24b1312_lab2_Q2.uvproj - µVision [Non-Commercial Use License]

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x01
b	0x00
sp	0x07
sp_max	0x0b
PC $	C:0x0102
auxr1	0x00
dptr	0x0000
states	47
sec	0.000023...
psw	0x01

```
     7: HERE: SJMP HERE
     8:
     9: ORG 0130H
    10: SUB16:
```

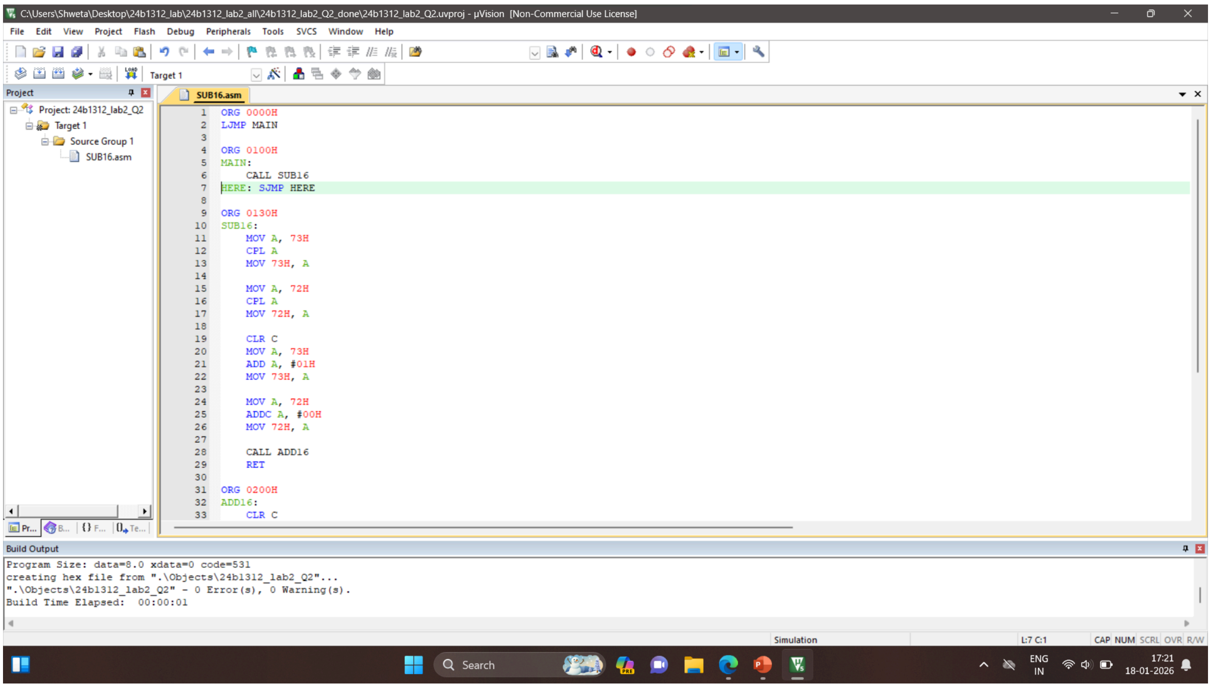
```
ORG 0000H
LJMP MAIN

ORG 0100H
MAIN:
    CALL SUB16
HERE: SJMP HERE

ORG 0130H
SUB16:
    MOV A, 73H
    CPL A
    MOV 73H, A

    MOV A, 72H
    CPL A
    MOV 72H, A

    CLR C
    MOV A, 73H
    ADD A, #01H
    MOV 73H, A
```

Address: I:070H

```
I:0x70: 80 00 FF FF 01 7F FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x85: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x9A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xAF: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xC4: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xD9: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xEE: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x03: 00 00 00 00 00 02 01 49 01 00 00 00 00 00 00 00 00 00 00 00 00
I:0x18: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
Running with Code Size Limit: 2K
Load "C:\\Users\\Shweta\\Desktop\\24b1312_lab\\24b1312_lab2_all\\24b1312_lab2_Q2_done\\Objects\\24b1312_lab2_Q2"
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE KILL LogicAnalyze LOAD LOG MAP MODE
```

Simulation t1: 0.00002350 sec L:7 C:1 CAP NUM SCRL OVR R/W



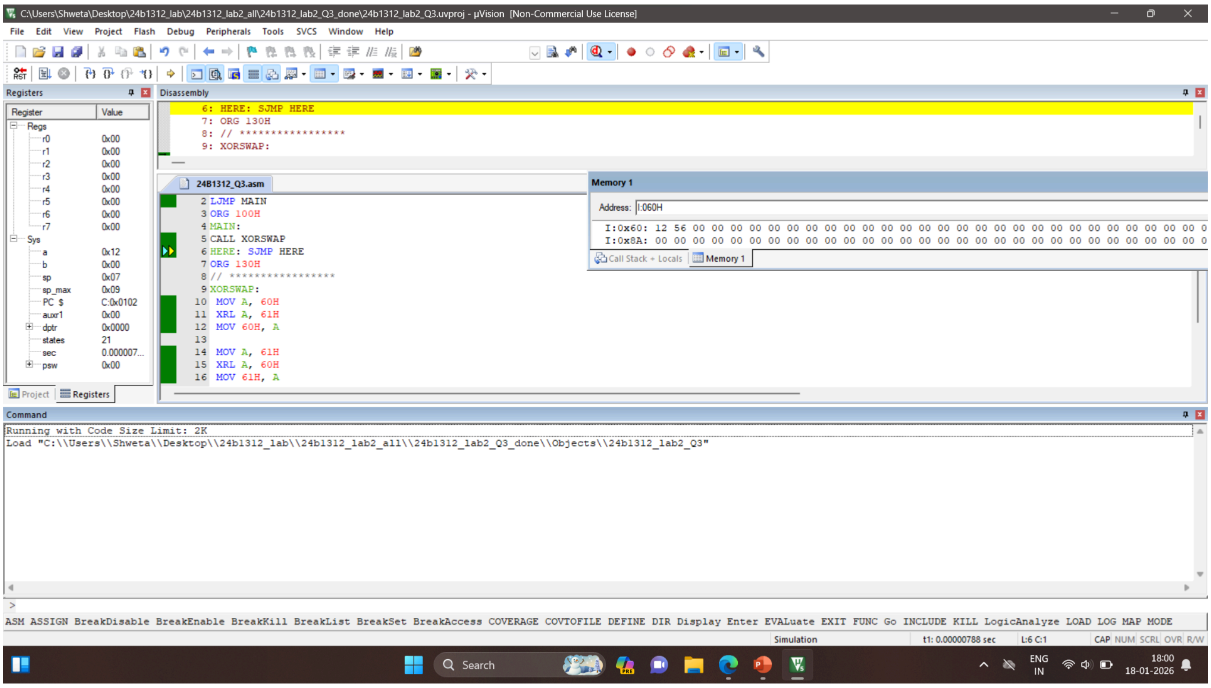
C:\Users\Shweta\Desktop\24b1312_lab\24b1312_lab2_all\24b1312_lab2_Q2_done\24b1312_lab2_Q2.uvproj - µVision [Non-Commercial Use License]

```
ORG 0000H
LJMP MAIN

ORG 0100H
MAIN:
    CALL SUB16
HERE: SJMP HERE

ORG 0130H
SUB16:
    MOV A, 73H
    CPL A
    MOV 73H, A

    MOV A, 72H
    CPL A
    MOV 72H, A

    CLR C
    MOV A, 73H
    ADD A, #01H
    MOV 73H, A

    MOV A, 72H
    ADDC A, #00H
    MOV 72H, A

    CALL ADD16
    RET

ORG 0200H
ADD16:
    CLR C
```

```
Program Size: data=8.0 xdata=0 code=531
creating hex file from ".\Objects\24b1312_lab2_Q2"...
".\Objects\24b1312_lab2_Q2" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
```

Simulation L:7 C:1 CAP NUM SCRL OVR R/W

Q3)



C:\Users\Shweta\Desktop\24b1312_lab\24b1312_lab2_all\24b1312_lab2_Q3_done\24b1312_lab2_Q3.uvproj - µVision [Non-Commercial Use License]

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x12
b	0x00
sp	0x07
sp_max	0x09
PC $	C:0x0102
auxr1	0x00
dptr	0x0000
states	21
sec	0.000007...
psw	0x00

```
    6: HERE: SJMP HERE
    7: ORG 130H
    8: // *****************
    9: XORSWAP:
```

24B1312_Q3.asm

```
LJMP MAIN
ORG 100H
MAIN:
CALL XORSWAP
HERE: SJMP HERE
ORG 130H
// *****************
XORSWAP:
 MOV A, 60H
 XRL A, 61H
 MOV 60H, A

 MOV A, 61H
 XRL A, 60H
 MOV 61H, A
```

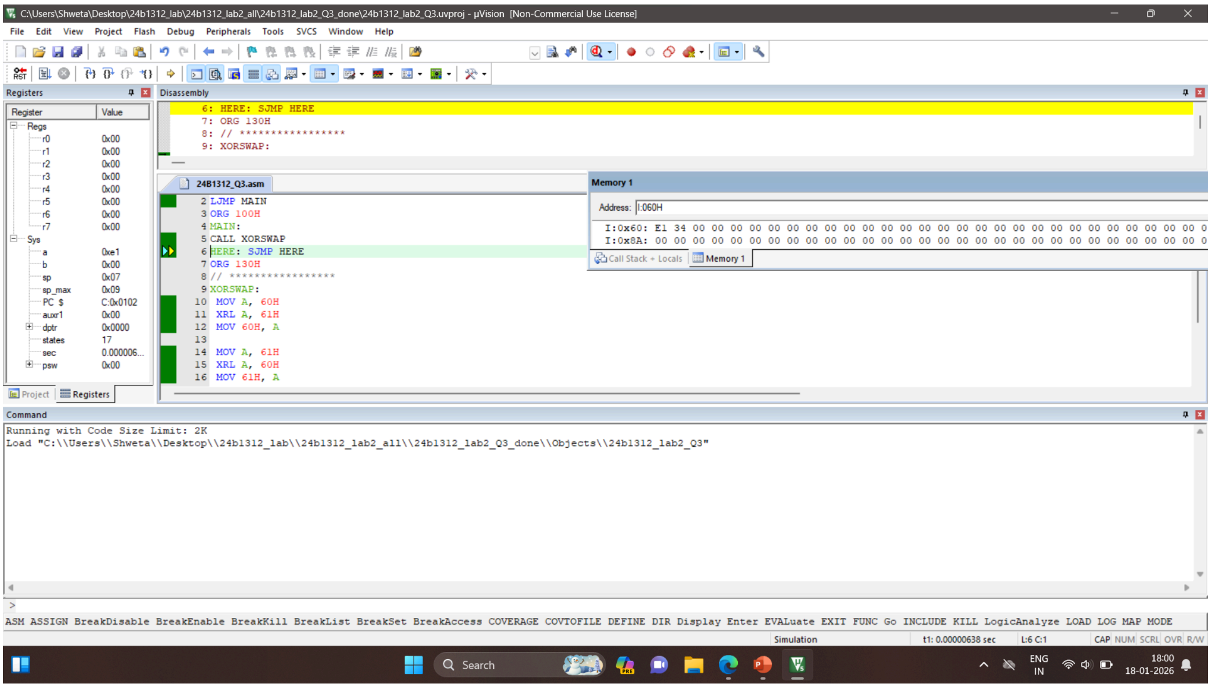
Address: I:060H

```
I:0x60: 12 56 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
I:0x8A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
```

```
Running with Code Size Limit: 2K
Load "C:\\Users\\Shweta\\Desktop\\24b1312_lab\\24b1312_lab2_all\\24b1312_lab2_Q3_done\\Objects\\24b1312_lab2_Q3"
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE KILL LogicAnalyze LOAD LOG MAP MODE
```

Simulation   t1: 0.00000788 sec   L:6 C:1   CAP NUM SCRL OVR R/W

ENG IN   18:00 18-01-2026



C:\Users\Shweta\Desktop\24b1312_lab\24b1312_lab2_all\24b1312_lab2_Q3_done\24b1312_lab2_Q3.uvproj - µVision [Non-Commercial Use License]

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0xe1
b	0x00
sp	0x07
sp_max	0x09
PC $	C:0x0102
auxr1	0x00
dptr	0x0000
states	17
sec	0.000006...
psw	0x00

```
    6: HERE: SJMP HERE
    7: ORG 130H
    8: // *****************
    9: XORSWAP:
```

24B1312_Q3.asm

```
LJMP MAIN
ORG 100H
MAIN:
CALL XORSWAP
HERE: SJMP HERE
ORG 130H
// *****************
XORSWAP:
 MOV A, 60H
 XRL A, 61H
 MOV 60H, A

 MOV A, 61H
 XRL A, 60H
 MOV 61H, A
```

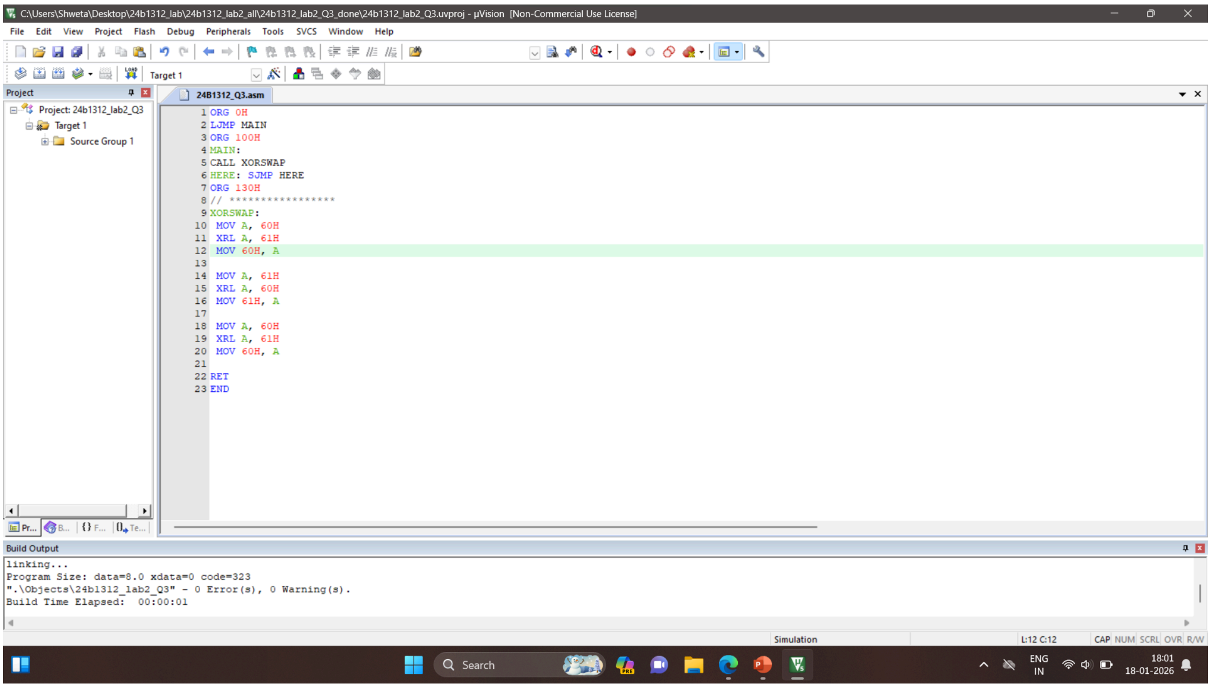
Address: I:060H

```
I:0x60: E1 34 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
I:0x8A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
```

```
Running with Code Size Limit: 2K
Load "C:\\Users\\Shweta\\Desktop\\24b1312_lab\\24b1312_lab2_all\\24b1312_lab2_Q3_done\\Objects\\24b1312_lab2_Q3"
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE KILL LogicAnalyze LOAD LOG MAP MODE
```

Simulation   t1: 0.00000638 sec   L:6 C:1   CAP NUM SCRL OVR R/W

ENG IN   18:00 18-01-2026



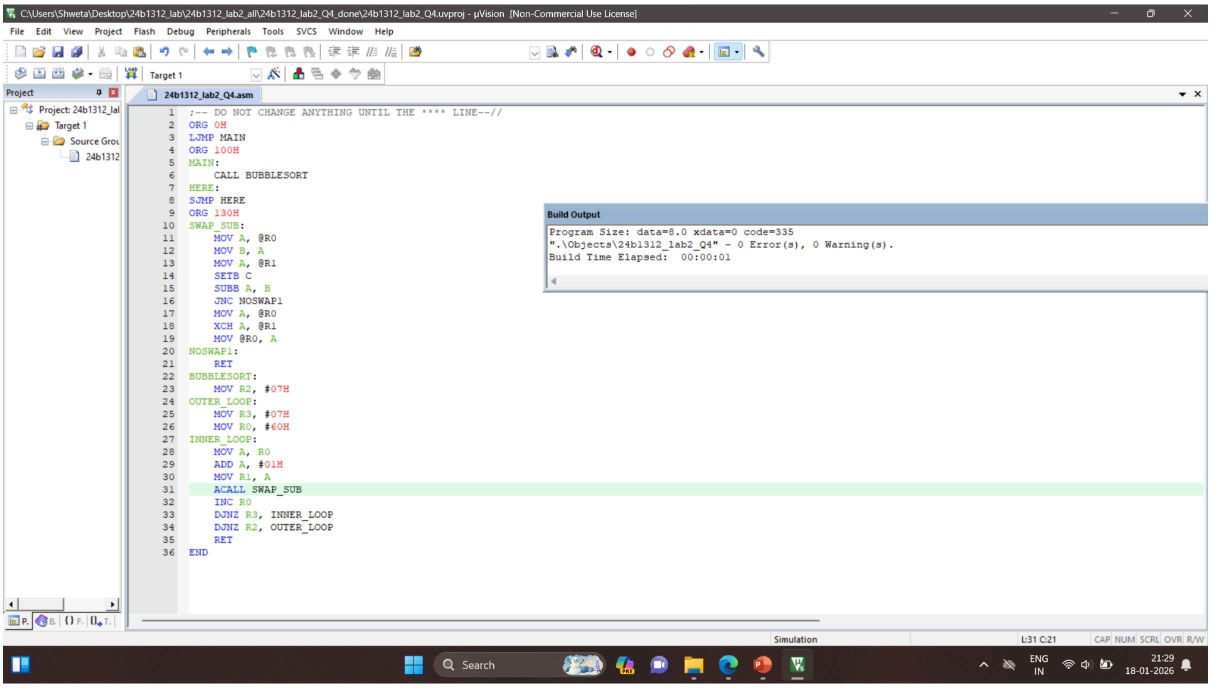
C:\Users\Shweta\Desktop\24b1312_lab\24b1312_lab2_all\24b1312_lab2_Q3_done\24b1312_lab2_Q3.uvproj - µVision [Non-Commercial Use License]

```
ORG 0H
LJMP MAIN
ORG 100H
MAIN:
CALL XORSWAP
HERE: SJMP HERE
ORG 130H
// *****************
XORSWAP:
 MOV A, 60H
 XRL A, 61H
 MOV 60H, A

 MOV A, 61H
 XRL A, 60H
 MOV 61H, A

 MOV A, 60H
 XRL A, 61H
 MOV 60H, A

RET
END
```

Build Output

```
linking...
Program Size: data=8.0 xdata=0 code=323
".\Objects\24b1312_lab2_Q3" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
```

Q4)

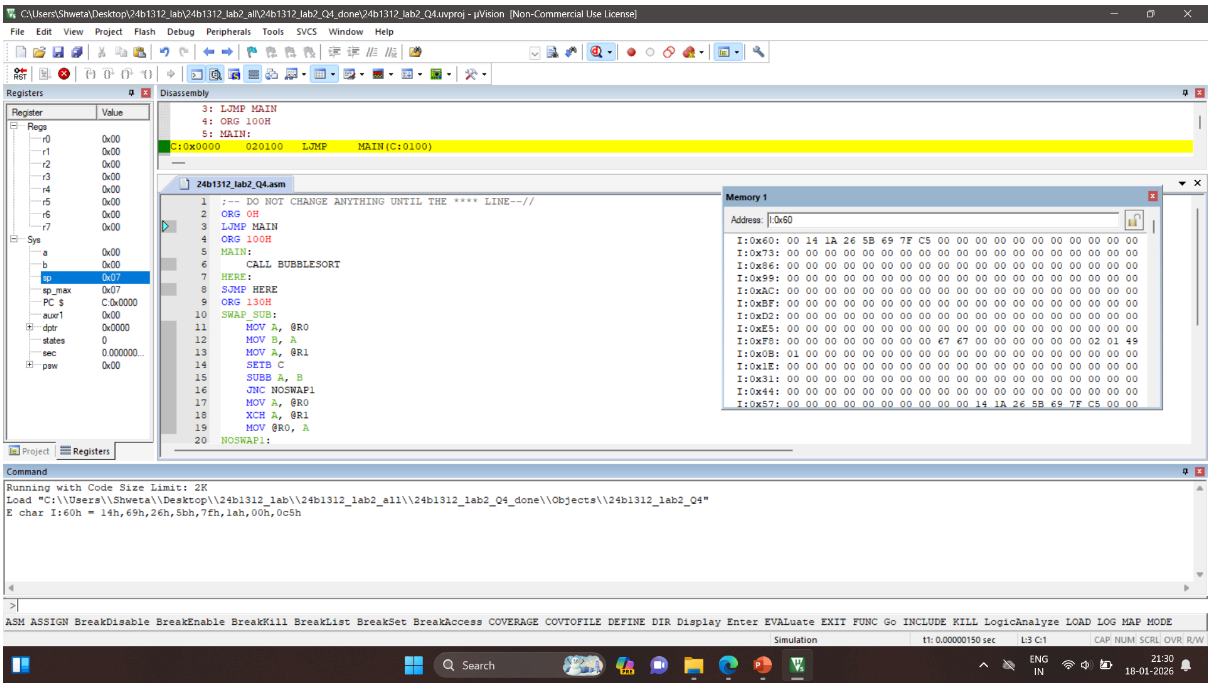


C:\Users\Shweta\Desktop\24b1312_lab\24b1312_lab2_all\24b1312_lab2_Q4_done\24b1312_lab2_Q4.uvproj - µVision [Non-Commercial Use License]

```
;-- DO NOT CHANGE ANYTHING UNTIL THE **** LINE--//
ORG 0H
LJMP MAIN
ORG 100H
MAIN:
    CALL BUBBLESORT
HERE:
SJMP HERE
ORG 130H
SWAP_SUB:
    MOV A, @R0
    MOV B, A
    MOV A, @R1
    SETB C
    SUBB A, B
    JNC NOSWAP1
    MOV A, @R0
    XCH A, @R1
    MOV @R0, A
NOSWAP1:
    RET
BUBBLESORT:
    MOV R2, #07H
OUTER_LOOP:
    MOV R3, #07H
    MOV R0, #60H
INNER_LOOP:
    MOV A, R0
    ADD A, #01H
    MOV R1, A
    ACALL SWAP_SUB
    INC R0
    DJNZ R3, INNER_LOOP
    DJNZ R2, OUTER_LOOP
    RET
END
```

Build Output

```
Program Size: data=8.0 xdata=0 code=335
".\Objects\24b1312_lab2_Q4" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
```



C:\Users\Shweta\Desktop\24b1312_lab\24b1312_lab2_all\24b1312_lab2_Q4_done\24b1312_lab2_Q4.uvproj - µVision [Non-Commercial Use License]

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x07
sp_max	0x07
PC $	C:0x0000
auxr1	0x00
dptr	0x0000
states	0
sec	0.000000...
psw	0x00

Disassembly

```
     3: LJMP MAIN
     4: ORG 100H
     5: MAIN:
C:0x0000    020100   LJMP     MAIN(C:0100)
```

24b1312_lab2_Q4.asm

```
;-- DO NOT CHANGE ANYTHING UNTIL THE **** LINE--//
ORG 0H
LJMP MAIN
ORG 100H
MAIN:
    CALL BUBBLESORT
HERE:
SJMP HERE
ORG 130H
SWAP_SUB:
    MOV A, @R0
    MOV B, A
    MOV A, @R1
    SETB C
    SUBB A, B
    JNC NOSWAP1
    MOV A, @R0
    XCH A, @R1
    MOV @R0, A
NOSWAP1:
```

Memory 1

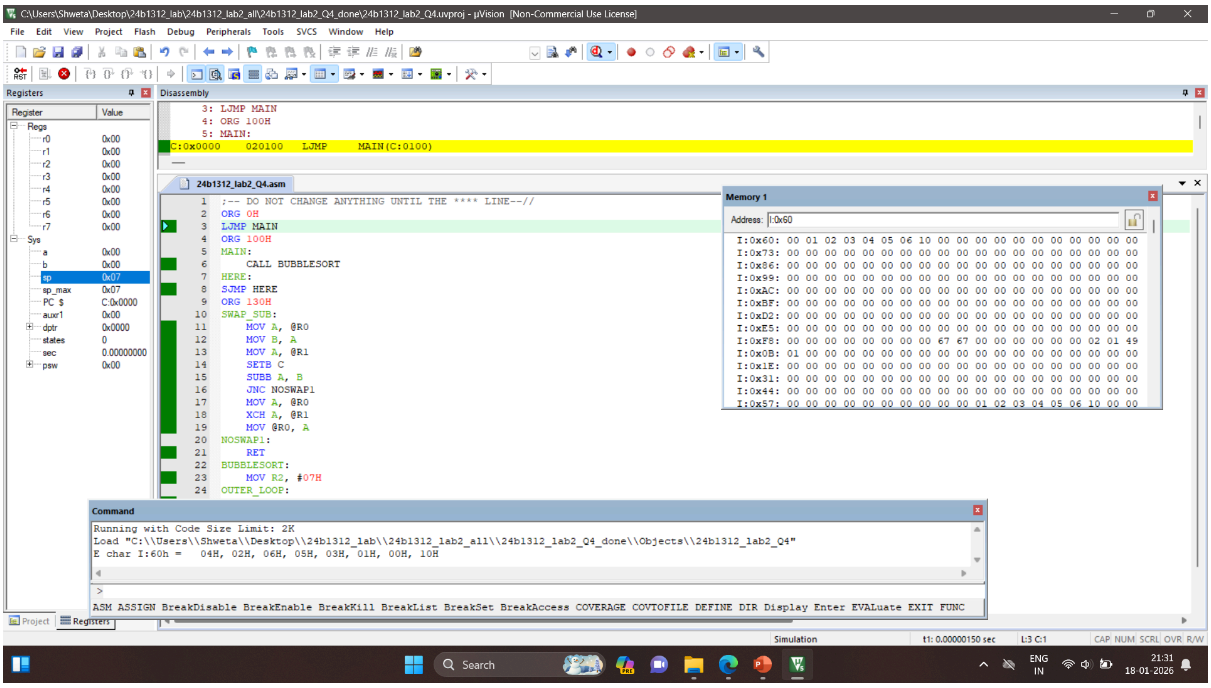
Address: I:0x60

```
I:0x60: 00 14 1A 26 5B 69 7F C5 00 00 00 00 00 00 00 00 00 00 00
I:0x73: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x86: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x99: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xAC: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xBF: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xD2: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xE5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xF8: 00 00 00 00 00 00 00 00 67 67 00 00 00 00 00 00 02 01 49
I:0x0B: 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x1E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x31: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x44: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x57: 00 00 00 00 00 00 00 00 00 00 14 1A 26 5B 69 7F C5 00 00
```

Command

```
Running with Code Size Limit: 2K
Load "C:\\Users\\Shweta\\Desktop\\24b1312_lab\\24b1312_lab2_all\\24b1312_lab2_Q4_done\\Objects\\24b1312_lab2_Q4"
E char I:60h = 14h,69h,26h,5bh,7fh,1ah,00h,0c5h
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE KILL LogicAnalyze LOAD LOG MAP MODE
```

Simulation   t1: 0.00000150 sec   L:3 C:1



C:\Users\Shweta\Desktop\24b1312_lab\24b1312_lab2_all\24b1312_lab2_Q4_done\24b1312_lab2_Q4.uvproj - µVision [Non-Commercial Use License]

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x07
sp_max	0x07
PC $	C:0x0000
auxr1	0x00
dptr	0x0000
states	0
sec	0.00000000
psw	0x00

Disassembly

```
     3: LJMP MAIN
     4: ORG 100H
     5: MAIN:
C:0x0000    020100   LJMP     MAIN(C:0100)
```

24b1312_lab2_Q4.asm

```
;-- DO NOT CHANGE ANYTHING UNTIL THE **** LINE--//
ORG 0H
LJMP MAIN
ORG 100H
MAIN:
    CALL BUBBLESORT
HERE:
SJMP HERE
ORG 130H
SWAP_SUB:
    MOV A, @R0
    MOV B, A
    MOV A, @R1
    SETB C
    SUBB A, B
    JNC NOSWAP1
    MOV A, @R0
    XCH A, @R1
    MOV @R0, A
NOSWAP1:
    RET
BUBBLESORT:
    MOV R2, #07H
OUTER_LOOP:
```

Memory 1

Address: I:0x60

```
I:0x60: 00 01 02 03 04 05 06 10 00 00 00 00 00 00 00 00 00 00 00
I:0x73: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x86: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x99: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xAC: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xBF: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xD2: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xE5: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0xF8: 00 00 00 00 00 00 00 00 67 67 00 00 00 00 00 00 02 01 49
I:0x0B: 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x1E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x31: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x44: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x57: 00 00 00 00 00 00 00 00 00 00 01 02 03 04 05 06 10 00 00
```

Command

```
Running with Code Size Limit: 2K
Load "C:\\Users\\Shweta\\Desktop\\24b1312_lab\\24b1312_lab2_all\\24b1312_lab2_Q4_done\\Objects\\24b1312_lab2_Q4"
E char I:60h =   04H, 02H, 06H, 05H, 03H, 01H, 00H, 10H
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter EVALuate EXIT FUNC
```

Simulation   t1: 0.00000150 sec   L:3 C:1

Aim

To write and execute 8051 assembly language programs for:

1. Addition and subtraction of 16-bit numbers

2. Swapping two memory locations using XOR

3. Sorting an array using the Bubble Sort algorithm

Observation

- Carry propagation is essential in multi-byte arithmetic.
- Two's complement subtraction works correctly when reuse of addition logic is done.
- XOR swap exchanges values without using extra memory.
- Bubble sort correctly arranges unsigned 8-bit numbers using repeated comparisons.

Conclusion

All programs executed successfully and produced correct results for the given test cases. The ADD16 subroutine proved reusable for subtraction, improving modularity. Bubble sort effectively sorted the array in ascending order.

What I Learnt

- Handling carry and borrow in 16-bit arithmetic on an 8-bit microcontroller
- Implementation of 2's complement arithmetic in assembly
- Efficient data swapping using XOR logic
- Use of indirect addressing and nested loops for array processing
- Writing modular and reusable assembly subroutines