AIM 1: Given block of 100h to 120h. Find out how many bytes from this block are greater than the number in r2 and less than the number in r3. Store the count in r4

- Taking each byte one by one from given block
- Here two limits are given higher limit in r3 and lower limit in r2
- We check for higher limit and then lower limit
- If the byte lies within these limits then the count will be incremented

ASSEMBLY LANGUAGE CODE:

Ex:

Address	100h	101h	102h	… 120h
Number	12h	44h	7fh	… 99h

R3 is the upper limit – 8bh
R2 is the lower limit – 32h
R4 is the counter

1. Taking value first in DPTR for external ram since it can carry 16-bit address location
2. Counter is used as index
3. Getting lower limit in r2
4. Getting upper limit in r3
5. Taking the first data i.e., 12 moved to accumulator
6. Compare the value stored in accumulator with r3
7. Subtracting 8b from 12 and then result will be checked by instruction if it's zero or not, if equal is zero then numbers will be zero and the case when it's not zero then go to lower
8. Carry will sit when one number is greater than other
9. If no carry, move out of the condition
10. CJNE a 20h Higher – to check if number is greater than lower limit or not
11. Similar approach is followed like previously, numbers not equal then go to higher
12. Numbers are equal then jump out
13. Changing of address location to 44h when we come to the condition of OUT for DPTR
14. Program will continue until r7 reaches 0.

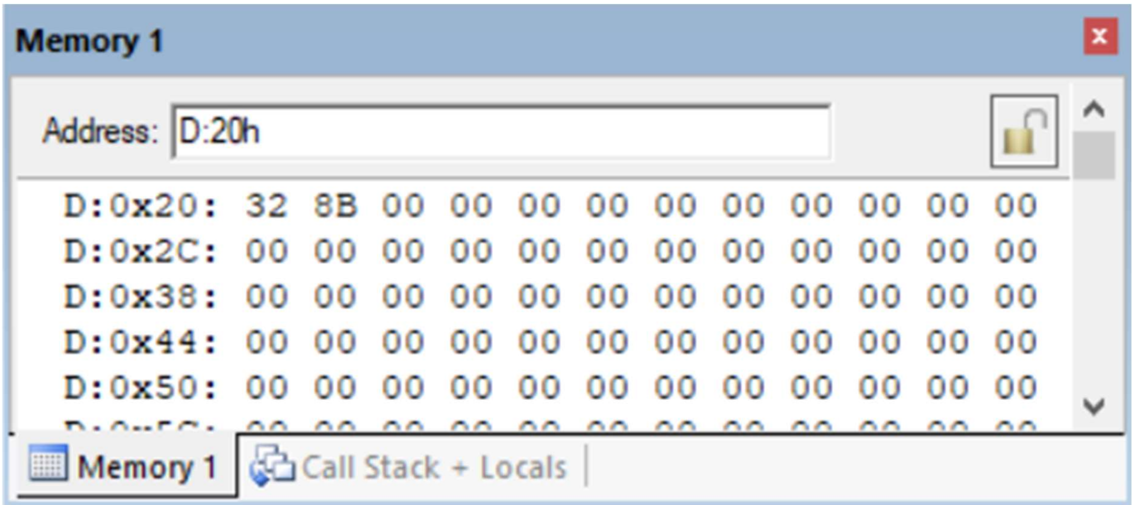
100 101 102 103 104 105 106 107 108 109
10a 10b 10c 10d 10e 10f 110 111 112 113
114 115 116 117 118 119 11a 11b 11c 11d
11e 11f 120
total 33 numbers
0-ff is the limit in hexadecimal

CJNE

- To check if a is lower or not
- If not equal then go to next line

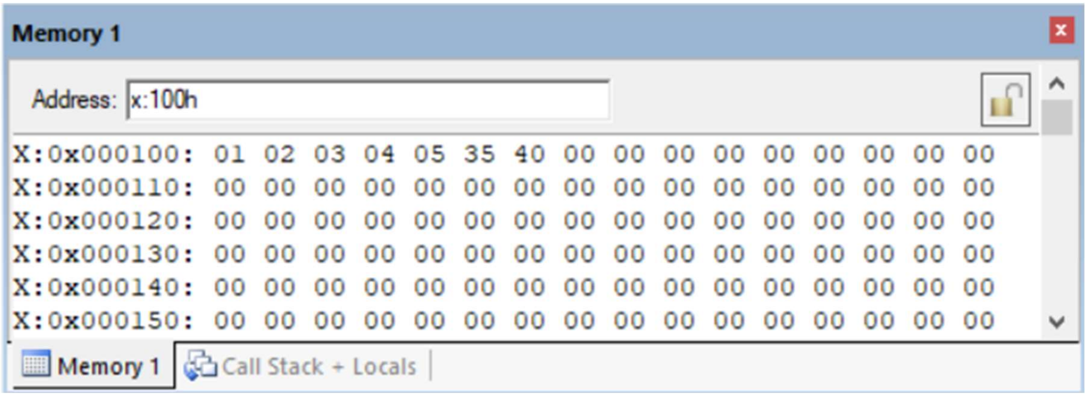
```
            mov dptr, #0100h ; get initial location
            mov r7, #20h ; counter used as index
            mov r4, #00h ; number counter
            mov r3,21h ; get the upper limit in 21h
            mov r2,20h ; get the lower limt in 20h

Nxt:    movx a, @dptr ; get the content in acc
            cjne a,21h,Lower ; check the upper limit first
            sjmp Out ; if number is equal

Lower:      mov b,21h
            subb a,b
            jnc Out ; jump out
            movx a,@dptr
            cjne a,20h, Higher ; check lower limi
            sjmp Out ; if number is lower

Higher: mov b,20h
            movx a,@dptr
            subb a,b
            jc Out ; jump out
            inc r4 ; if number within limit increment count

Out:    inc dptr ; get next location
            djnz r7, Nxt ; repeat until block completes
```

INPUT GIVEN:



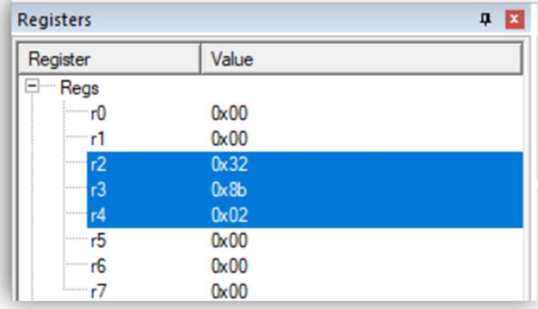
Address: D:20h

```
D:0x20:  32 8B 00 00 00 00 00 00 00 00 00 00
D:0x2C:  00 00 00 00 00 00 00 00 00 00 00 00
D:0x38:  00 00 00 00 00 00 00 00 00 00 00 00
D:0x44:  00 00 00 00 00 00 00 00 00 00 00 00
D:0x50:  00 00 00 00 00 00 00 00 00 00 00 00
```



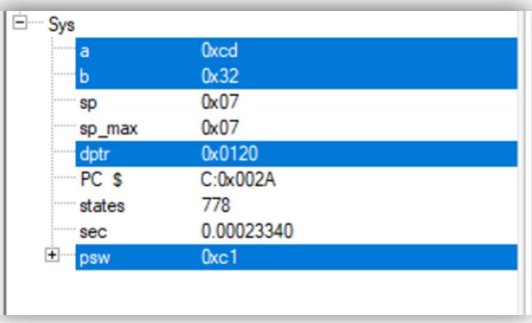
```
Memory 1
Address: x:100h
X:0x000100: 01 02 03 04 05 35 40 00 00 00 00 00 00 00 00 00
X:0x000110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x000120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x000130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x000140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x000150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

OBSERVED OUTPUT:

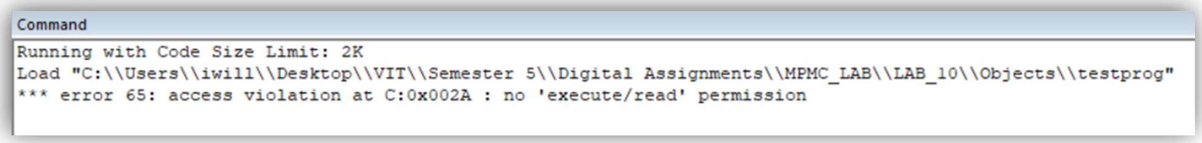


Registers

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x32
r3	0x8b
r4	0x02
r5	0x00
r6	0x00
r7	0x00

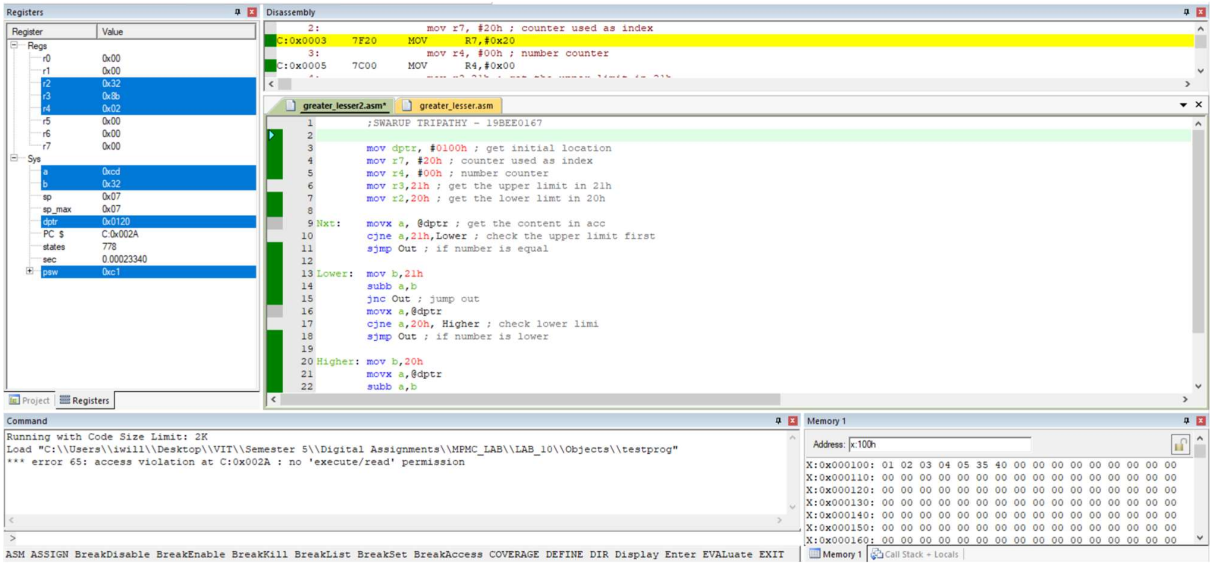


Sys	
a	0xcd
b	0x32
sp	0x07
sp_max	0x07
dptr	0x0120
PC $	C:0x002A
states	778
sec	0.00023340
psw	0xc1



```
Command
Running with Code Size Limit: 2K
Load "C:\\Users\\iwill\\Desktop\\VIT\\Semester 5\\Digital Assignments\\MPMC_LAB\\LAB_10\\Objects\\testprog"
*** error 65: access violation at C:0x002A : no 'execute/read' permission
```

For some reason I was getting this error again and again



Registers

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x32
r3	0x8b
r4	0x02
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0xcd
b	0x32
sp	0x07
sp_max	0x07
dptr	0x0120
PC $	C:0x002A
states	778
sec	0.00023340
psw	0xc1

Disassembly

```
      2:                mov r7, #20h ; counter used as index
C:0x0003    7F20     MOV      R7,#0x20
      3:                mov r4, #00h ; number counter
C:0x0005    7C00     MOV      R4,#0x00
```

```
        ;SWARUP TRIPATHY - 19BEE0167

        mov dptr, #0100h ; get initial location
        mov r7, #20h ; counter used as index
        mov r4, #00h ; number counter
        mov r3,21h ; get the upper limit in 21h
        mov r2,20h ; get the lower limt in 20h

Nxt:    movx a, @dptr ; get the content in acc
        cjne a,21h,Lower ; check the upper limit first
        sjmp Out ; if number is equal

Lower:  mov b,21h
        subb a,b
        jnc Out ; jump out
        movx a,@dptr
        cjne a,20h, Higher ; check lower limi
        sjmp Out ; if number is lower

Higher: mov b,20h
        movx a,@dptr
        subb a,b
```

```
Command
Running with Code Size Limit: 2K
Load "C:\\Users\\iwill\\Desktop\\VIT\\Semester 5\\Digital Assignments\\MPMC_LAB\\LAB_10\\Objects\\testprog"
*** error 65: access violation at C:0x002A : no 'execute/read' permission
```

```
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate EXIT
```

```
Memory 1
Address: x:100h
X:0x000100: 01 02 03 04 05 35 40 00 00 00 00 00 00 00 00 00
X:0x000110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x000120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x000130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x000140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x000150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
X:0x000160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

AIM 2: Given block of 100h to 120h. Find out how many bytes from this block are smaller than the number in r2

ASSEMBLY LANGUAGE CODE:

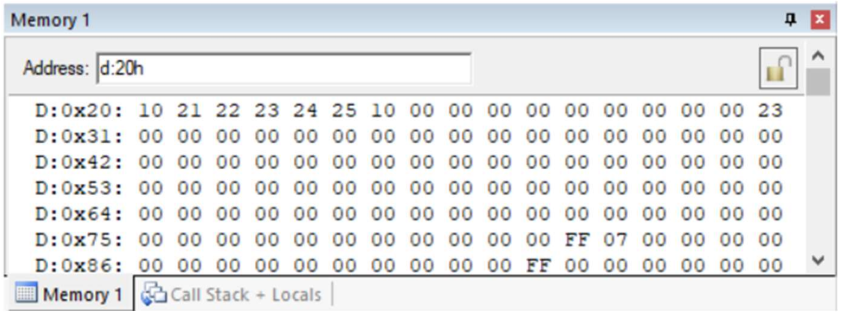
Ex:

Address	100h	101h	102h	… 120h
Number	12h	44h	7fh	… 99h

R3 is the upper limit – 8bh
R2 is the lower limit – 32h
R4 is the counter

```
MOV dtpr,#0100h                          ;get initial location
MOV r7,#20h                              ;counter
MOV r4,#00h                              ;number counter
MOV 20h,r2                               ;get the lower limit here in 20th location
MOV 21h,r3                               ;get the Upper Limit here in 21st location

Nxt:    CJNE a,21h,Lower                 ;check the upper limit first
        SJMP out                         ; If number is larger

Lower: JNC out                           ; jump out
        CJNE a,20h, Limit                ; check lower limit
        SJMP out                         ; if number is equal

Higher:JNC out                           ; jump out
        INC r4                           ; if number within limit increment

out:    INC dptr                         ; get next location
        DJNZ r7,Nxt                      ;repeat until block completes
        END
```

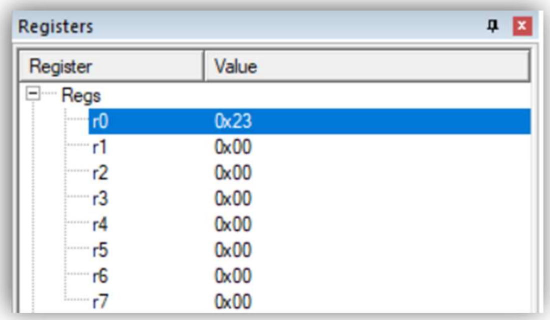
INPUT GIVEN:



Address: d:20h

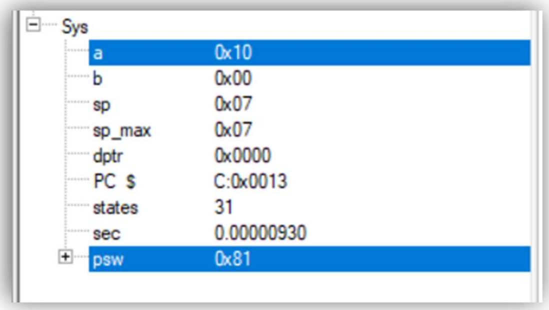
```
D:0x20: 10 21 22 23 24 25 10 00 00 00 00 00 00 00 00 00 23
D:0x31: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x42: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x53: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x64: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x75: 00 00 00 00 00 00 00 00 00 00 00 FF 07 00 00 00 00
D:0x86: 00 00 00 00 00 00 00 00 00 00 FF 00 00 00 00 00 00
```

OBSERVED OUTPUT:

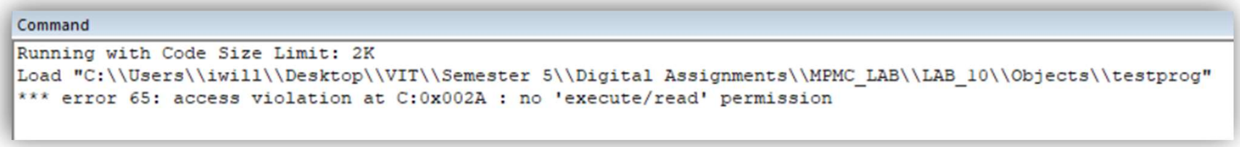


Registers

Register	Value
Regs	
r0	0x23
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00



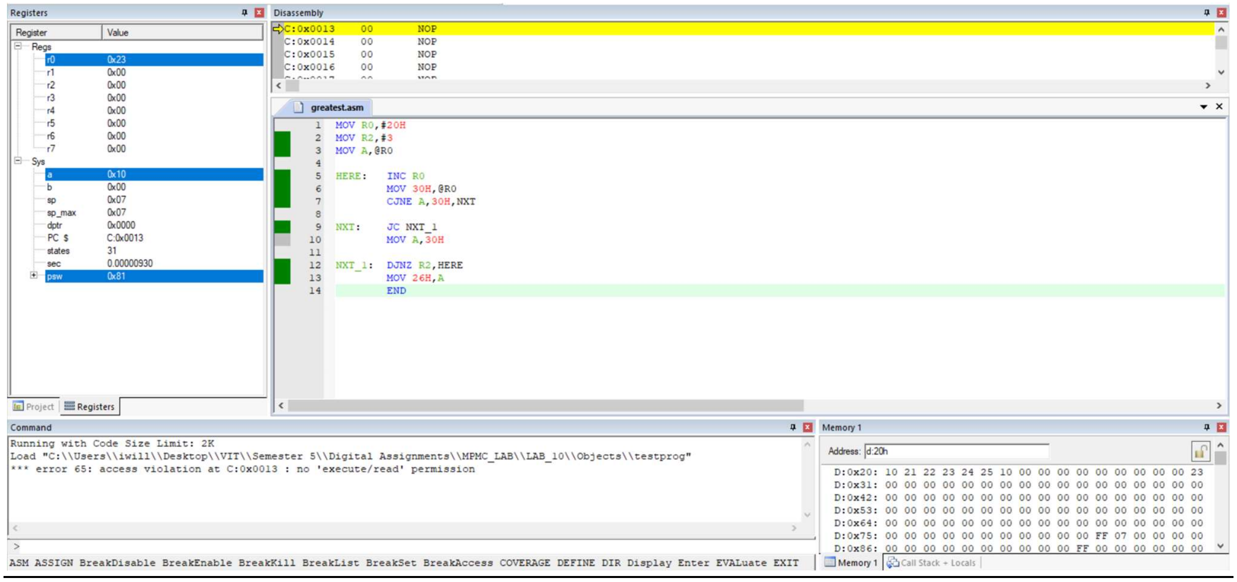
Sys	
a	0x10
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC $	C:0x0013
states	31
sec	0.00000930
psw	0x81



Command

```
Running with Code Size Limit: 2K
Load "C:\\Users\\iwill\\Desktop\\VIT\\Semester 5\\Digital Assignments\\MPMC_LAB\\LAB_10\\Objects\\testprog"
*** error 65: access violation at C:0x002A : no 'execute/read' permission
```

For some reason I was getting this error again and again



Registers

Register	Value
Regs	
r0	0x23
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x10
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC $	C:0x0013
states	31
sec	0.00000930
psw	0x81

Disassembly

```
C:0x0013    00        NOP
C:0x0014    00        NOP
C:0x0015    00        NOP
C:0x0016    00        NOP
```

greatest.asm

```
MOV R0,#20H
MOV R2,#3
MOV A,@R0

HERE:   INC R0
        MOV 30H,@R0
        CJNE A,30H,NXT

NXT:    JC NXT_1
        MOV A,30H

NXT_1:  DJNZ R2,HERE
        MOV 26H,A
        END
```

Command

```
Running with Code Size Limit: 2K
Load "C:\\Users\\iwill\\Desktop\\VIT\\Semester 5\\Digital Assignments\\MPMC_LAB\\LAB_10\\Objects\\testprog"
*** error 65: access violation at C:0x0013 : no 'execute/read' permission
```

Memory 1

Address: d:20h

```
D:0x20: 10 21 22 23 24 25 10 00 00 00 00 00 00 00 00 00 23
D:0x31: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x42: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x53: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x64: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x75: 00 00 00 00 00 00 00 00 00 00 00 FF 07 00 00 00 00
D:0x86: 00 00 00 00 00 00 00 00 00 00 00 FF 00 00 00 00 00
```

Project
Registers
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate EXIT
Memory 1
Call Stack + Locals

INFERENCE:

Using Keil software we were able to observe an increment in r4 in exp 1 determining the total numbers which are within the specified limits. As for aim 2 we see that we were able to find the smallest byte in the array. The only problem that I was facing was the error regarding the *execution permission displayed* in the command window.