DEPARTMENT OF COMPUTER &
SOFTWARE ENGINEERING

COLLEGE OF E&ME, NUST, RAWALPINDI

Microprocessor and Microcontroller-Based Design

Lab 02

SUBMITTED TO:
Dr Taimoor Zahid

SUBMITTED BY:
AMINA QADEER
Reg # 00000359607
DE-42 (C&SE)-A

Submission Date: 10/7/2022

Objectives:

Writing basic assembly in emu8086 and running programs on the console.

Related Topic/Chapter in theory class:

Hardware/Software required:

Hardware: PC
Software Tool: emu8086 v2.57

Tasks 1 :

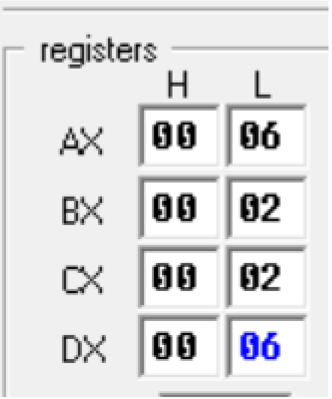
Define three variables of the size of registers. Add all three of them and store their sum in a register. Verify your results from the register contents. Also, confirm by doing the binary calculations on paper.

Solution:

```
    org 100h

    . model small
    .data

        register1 DW 2
     ; declaration of three 8 bits variables named register 1,2 and 3
        register2 DW 2
        register3 DW 2




    .code

        MOV AX,  register1
    ;moving the value of variables to 8-bit half of general purpose registers
        MOV BX,  register2
        MOV CX,  register3
```

```
main proc


    add AX,BX
    add AX,CX

    mov DX,AX



mov ah,4ch
int 21h




main endp
end main
```

Output:



```
registers
      H   L
AX   00  06
BX   00  02
CX   00  02
DX   00  06
```

Tasks 2:

Define three variables half the size of registers. Add all three of them and store their sum in a register. Verify your results from the register contents. Also, confirm by doing the binary calculations on paper. Remember to clear all the registers before using them

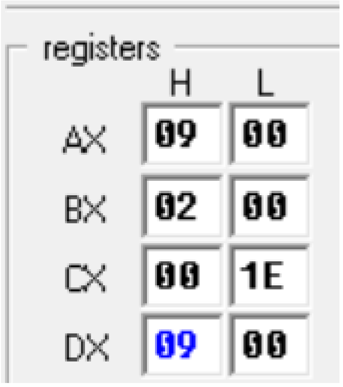
Solution:

```
org 100h

. model small
.data

      register1 DB 2
  ; declaration of three 8 bits variables named register 1,2 and 3
      register2 DB 2
      register3 DB 2




.code
```

```
            MOV AH,  register1    ;moving value of variables to 8 bit half og general purpose registers
            MOV BL,  register2
            MOV BH,  register3




      main proc


         add AH,BL
         add AH,BH

         mov DH,AH



      mov ah,4ch
      int 21h


        main endp
      end main

      ret
```

Output:

```
8+8+8 = 24 = 18 hexa value
2+2+2=6= 9
```



registers

	H	L
AX	09	00
BX	02	00
CX	00	1E
DX	09	00

Tasks:

Prompt the user for entering an upper-case letter. Upon receiving the input, your code should convert it into a lower-case letter and display it on the screen. What happens if the user inputs a lower-case letter?

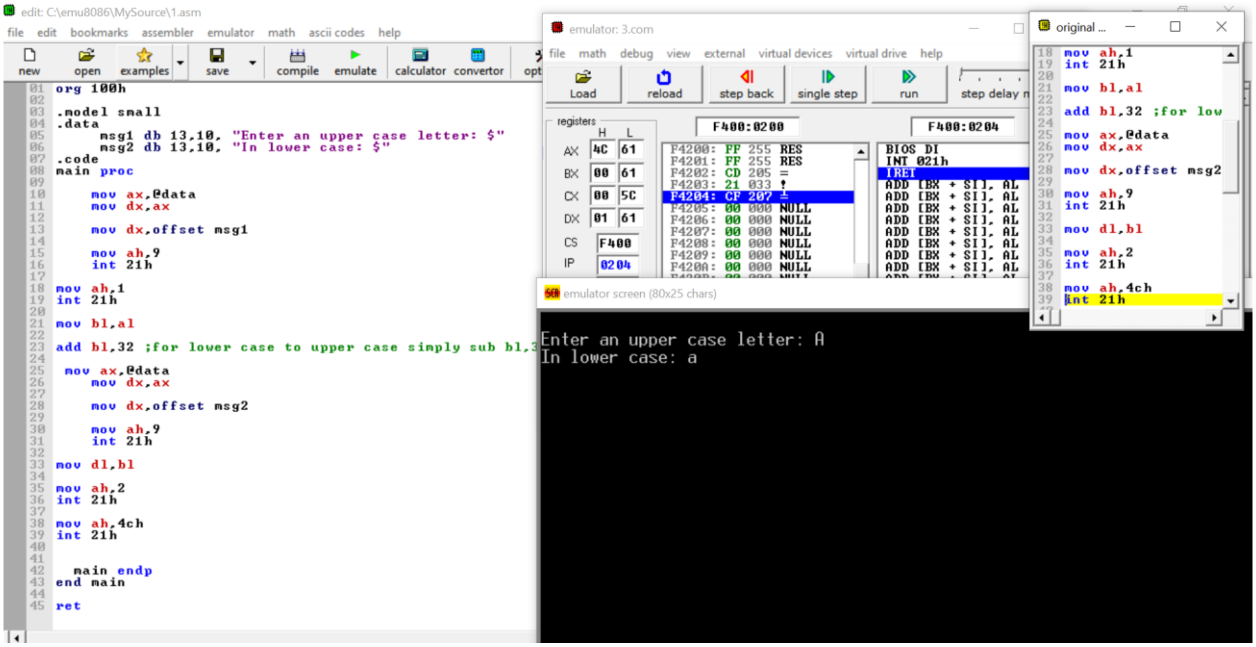
Solution:

```
org 100h

.model small
.data
    msg1 db 13,10, "Enter an upper case letter: $"
    msg2 db 13,10, "In lower case: $"
.code
main proc

   mov ax,@data
   mov dx,ax

   mov dx,offset msg1

   mov ah,9
   int 21h
```

```
        mov ah,1
        int 21h

        mov bl,al

        add bl,32 ;for lower case to upper case simply sub bl,32

         mov ax,@data
            mov dx,ax

            mov dx,offset msg2

            mov ah,9
            int 21h

  mov dl,bl
; Display the content of register DL on screen in ASCII form.

        mov ah,2
        int 21h

        mov ah,4ch
        int 21h


          main endp
        end main

        ret
```

Output:

UPPER TO LOWER

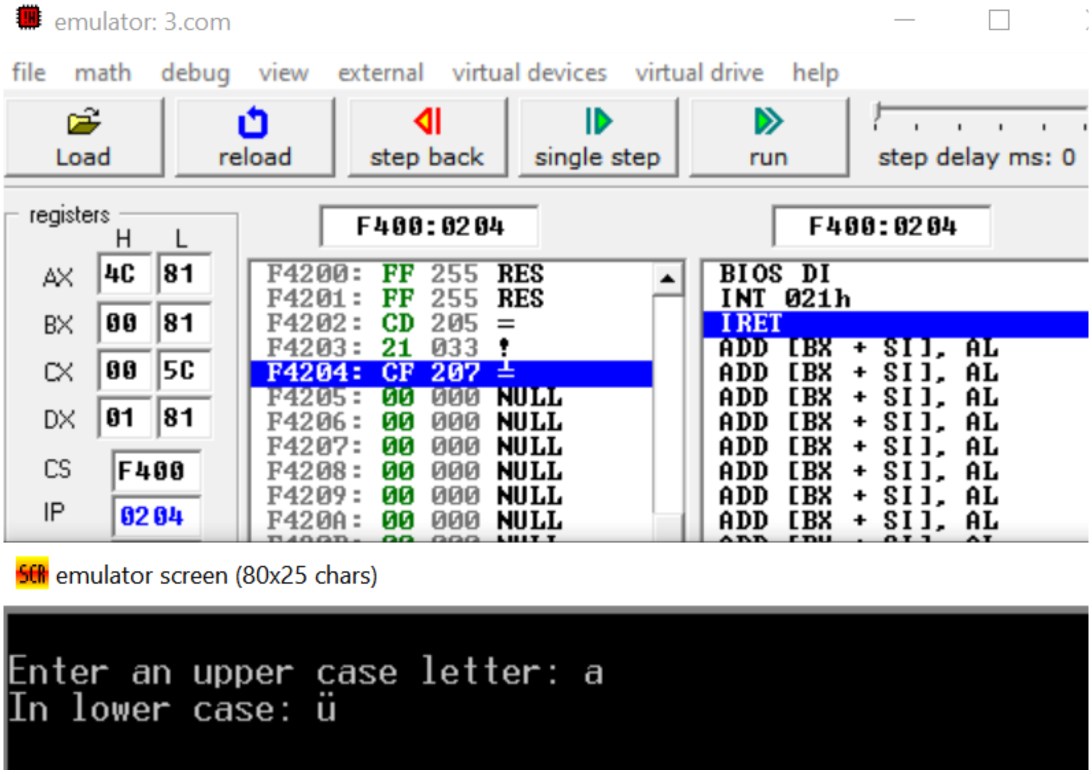

edit: C:\emu8086\MySource\1.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor opt
emulator: 3.com
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay m
original ...
emulator screen (80x25 chars)

```
org 100h

.model small
.data
     msg1 db 13,10, "Enter an upper case letter: $"
     msg2 db 13,10, "In lower case: $"
.code
main proc

        mov ax,@data
        mov dx,ax

        mov dx,offset msg1

        mov ah,9
        int 21h

mov ah,1
int 21h

mov bl,al

add bl,32 ;for lower case to upper case simply sub bl,3

 mov ax,@data
        mov dx,ax

        mov dx,offset msg2

        mov ah,9
        int 21h

mov dl,bl

mov ah,2
int 21h

mov ah,4ch
int 21h


   main endp
end main

ret
```

```
registers
     H   L
AX   4C  61
BX   00  61
CX   00  5C
DX   01  61
CS   F400
IP   0204

F400:0200                 F400:0204
F4200: FF 255 RES         BIOS DI
F4201: FF 255 RES         INT 021h
F4202: CD 205 =           IRET
F4203: 21 033 !           ADD [BX + SI], AL
F4204: CF 207 ±           ADD [BX + SI], AL
F4205: 00 000 NULL        ADD [BX + SI], AL
F4206: 00 000 NULL        ADD [BX + SI], AL
F4207: 00 000 NULL        ADD [BX + SI], AL
F4208: 00 000 NULL        ADD [BX + SI], AL
F4209: 00 000 NULL        ADD [BX + SI], AL
F420A: 00 000 NULL        ADD [BX + SI], AL
```

```
Enter an upper case letter: A
In lower case: a
```

WHEN A LOWER CASE PROMPT IS GIVEN:


emulator: 3.com
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
emulator screen (80x25 chars)

```
registers
     H   L
AX   4C  81
BX   00  81
CX   00  5C
DX   01  81
CS   F400
IP   0204

F400:0204                 F400:0204
F4200: FF 255 RES         BIOS DI
F4201: FF 255 RES         INT 021h
F4202: CD 205 =           IRET
F4203: 21 033 !           ADD [BX + SI], AL
F4204: CF 207 ±           ADD [BX + SI], AL
F4205: 00 000 NULL        ADD [BX + SI], AL
F4206: 00 000 NULL        ADD [BX + SI], AL
F4207: 00 000 NULL        ADD [BX + SI], AL
F4208: 00 000 NULL        ADD [BX + SI], AL
F4209: 00 000 NULL        ADD [BX + SI], AL
F420A: 00 000 NULL        ADD [BX + SI], AL
```

```
Enter an upper case letter: a
In lower case: ü
```

Tasks:

4. Prompt the user for entering a lower-case letter. Upon receiving the input, your code should convert it into an upper-case letter and display it on the screen. The code should be properly commented.

Solution:

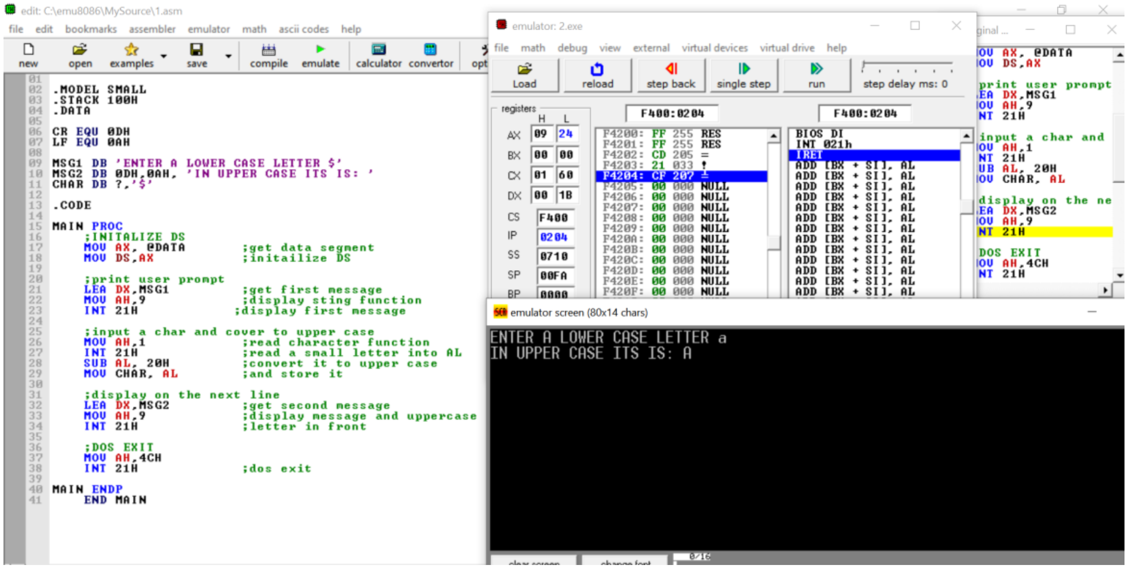
```
.MODEL SMALL
.STACK 100H
.DATA

CR EQU 0DH
LF EQU 0AH

MSG1 DB 'ENTER A LOWER CASE LETTER $'
MSG2 DB 0DH,0AH, 'IN UPPER CASE ITS IS: '
CHAR DB ?,'$'

.CODE

MAIN PROC
   ;INITALIZE DS
   MOV AX, @DATA       ;get data segment
   MOV DS,AX           ;initailize DS

   ;print user prompt
   LEA DX,MSG1         ;get first message
   MOV AH,9            ;display sting function
   INT 21H             ;display first message

   ;input a char and cover to upper case
   MOV AH,1            ;read character function
   INT 21H             ;read a small letter into AL
   SUB AL, 20H         ;convert it to upper case
   MOV CHAR, AL        ;and store it

   ;display on the next line
   LEA DX,MSG2         ;get second message
   MOV AH,9            ;display message and uppercase
```

```
        INT 21H             ;letter in front

        ;DOS EXIT
        MOV AH,4CH
        INT 21H             ;dos exit

    MAIN ENDP
        END MAIN
```

Output:

A = 41 hex
a = 61 hex

LOWER TO UPPER



edit: C:\emu8086\MySource\1.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor opt
emulator: 2.exe
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
emulator screen (80x14 chars)

```
.MODEL SMALL
.STACK 100H
.DATA

CR EQU 0DH
LF EQU 0AH

MSG1 DB 'ENTER A LOWER CASE LETTER $'
MSG2 DB 0DH,0AH, 'IN UPPER CASE ITS IS: '
CHAR DB ?,'$'

.CODE

MAIN PROC
    ;INITALIZE DS
    MOV AX, @DATA           ;get data segment
    MOV DS,AX               ;initailize DS

    ;print user prompt
    LEA DX,MSG1             ;get first message
    MOV AH,9                ;display sting function
    INT 21H                ;display first message

    ;input a char and cover to upper case
    MOV AH,1                ;read character function
    INT 21H                 ;read a small letter into AL
    SUB AL, 20H             ;convert it to upper case
    MOV CHAR, AL            ;and store it

    ;display on the next line
    LEA DX,MSG2             ;get second message
    MOV AH,9                ;display message and uppercase
    INT 21H                 ;letter in front

    ;DOS EXIT
    MOV AH,4CH
    INT 21H                 ;dos exit

MAIN ENDP
    END MAIN
```

	H	L
AX	09	24
BX	00	00
CX	01	60
DX	00	1B
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	

```
F400:0204
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ╧
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
```

```
F400:0204
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
```

```
ENTER A LOWER CASE LETTER a
IN UPPER CASE ITS IS: A
```

Conclusion:

Making logic for different algorithms in assembly and using the emu8086 console.