

Task-03:

```
.MODEL small
.STACK 100h
.DATA
    msg1 db "Enter a number: $"
    msg2 db 0Dh, 0Ah, "Positive $"
    msg3 db 0Dh, 0Ah, "Negative $"
.CODE
MAIN PROC
    mov ax, @DATA
    mov ds, ax

    ; Display "Enter a number: "
    mov ah, 09h
    lea dx, msg1
    int 21h

    ; Read a number from user
    mov ah, 01h
    int 21h
    mov bl, al ; Store the number in BL

    ; Convert ASCII to integer
    sub bl, 30h

    ; Check if the number is positive or negative
    cmp bl, 0
    jl negative
    jg positive

    jmp endProgram ; If the input is 0, exit

positive:
    mov ah, 09h
    lea dx, msg2
    int 21h
    jmp endProgram

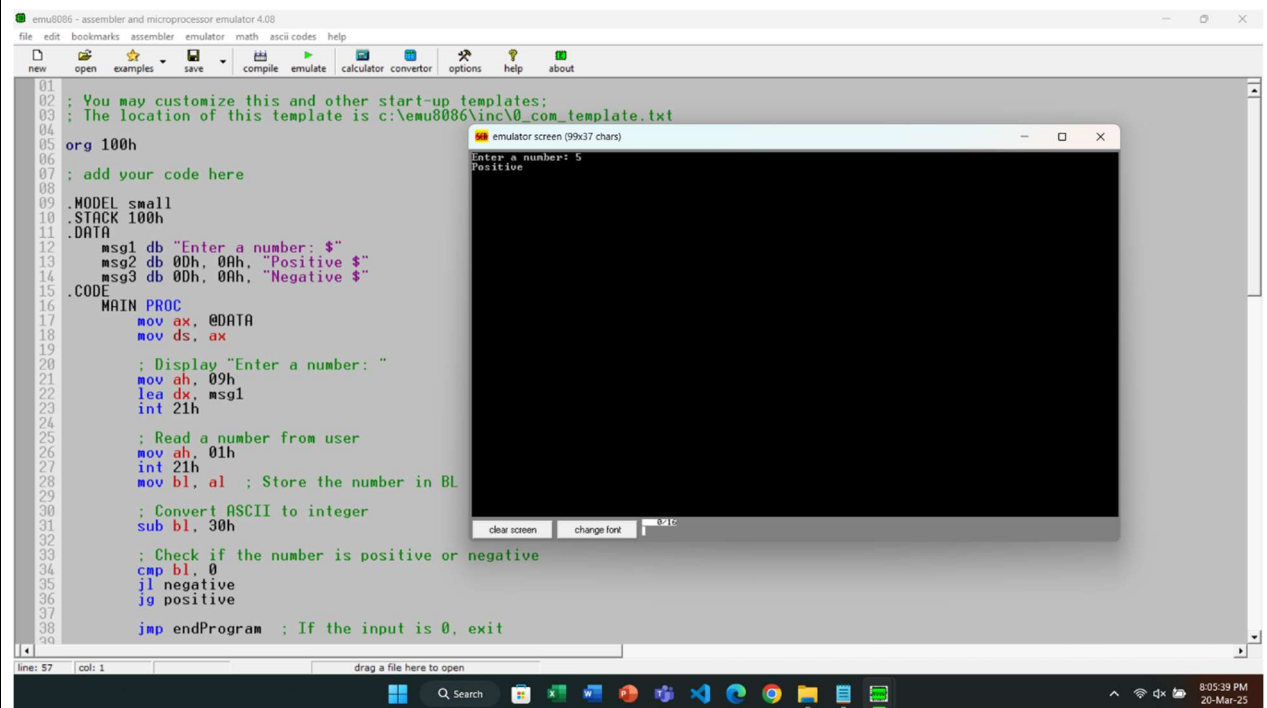
negative:
    mov ah, 09h
    lea dx, msg3
    int 21h

endProgram:
```

```
mov ah, 4Ch
int 21h
```

```
MAIN ENDP
END MAIN
```

Output:



```
01
02 ; You may customize this and other start-up templates;
03 ; The location of this template is c:\emu8086\inc\0_com_template.txt
04
05 org 100h
06
07 ; add your code here
08
09 .MODEL small
10 .STACK 100h
11 .DATA
12     msg1 db "Enter a number: $"
13     msg2 db 0Dh, 0Ah, "Positive $"
14     msg3 db 0Dh, 0Ah, "Negative $"
15 .CODE
16 MAIN PROC
17     mov ax, @DATA
18     mov ds, ax
19
20     ; Display "Enter a number: "
21     mov ah, 09h
22     lea dx, msg1
23     int 21h
24
25     ; Read a number from user
26     mov ah, 01h
27     int 21h
28     mov bl, al ; Store the number in BL
29
30     ; Convert ASCII to integer
31     sub bl, 30h
32
33     ; Check if the number is positive or negative
34     cmp bl, 0
35     jl negative
36     jg positive
37
38     jmp endProgram ; If the input is 0, exit
39
```

emulator screen (99x37 chars)

```
Enter a number: 5
Positive
```

clear screen change font 8.710

line: 57 col: 1 drag a file here to open

8:05:39 PM 20-Mar-25

Task-04:

```
.MODEL small
.STACK 100h
.DATA
    msg1 db "Enter a number: $"
    msg2 db 0Dh, 0Ah, "Less than 5 $"
    msg3 db 0Dh, 0Ah, "Greater than 5 $"
    msg4 db 0Dh, 0Ah, "Equal to 5 $"
```

```
.CODE
MAIN PROC
    mov ax, @DATA
    mov ds, ax

    ; Store 5 in CL
    mov cl, 5

    ; Display "Enter a number: "
    mov ah, 09h
    lea dx, msg1
    int 21h

    ; Read a number from user
    mov ah, 01h
    int 21h
    mov bl, al ; Store input in BL

    ; Convert ASCII to integer
    sub bl, 30h

    ; Compare user input with CL
    cmp bl, cl
    jl lessThan
    jg greaterThan
    je equalTo

lessThan:
    mov ah, 09h
    lea dx, msg2
    int 21h
    jmp endProgram

greaterThan:
```

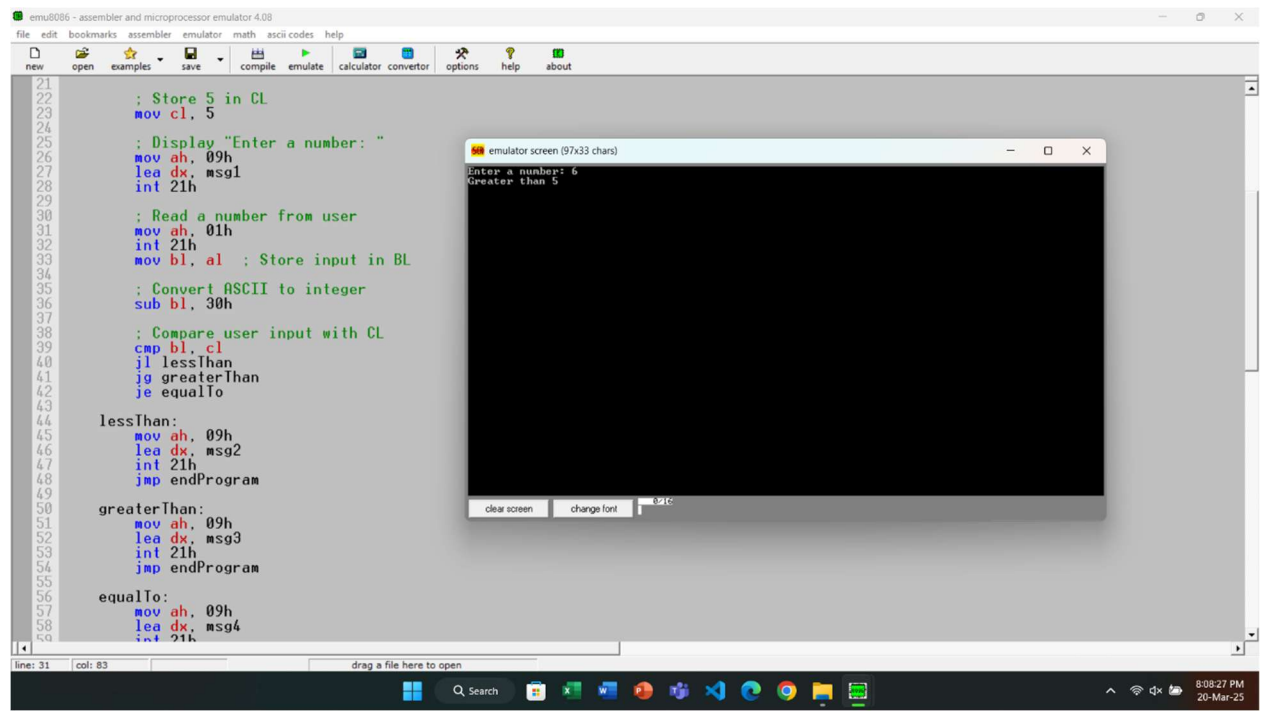
```
mov ah, 09h
lea dx, msg3
int 21h
jmp endProgram
```

```
equalTo:
mov ah, 09h
lea dx, msg4
int 21h
```

```
endProgram:
mov ah, 4Ch
int 21h
```

```
MAIN ENDP
END MAIN
```

Output:



Task-06:

```
.MODEL small
.STACK 100h
.DATA
    msg1 db "Enter a character: $"
    msg2 db 0Dh, 0Ah, "$" ; New line before printing the character
    msg3 db 0Dh, 0Ah, "Thank you.$" ; Message at the end

.CODE
MAIN PROC
    mov ax, @DATA
    mov ds, ax

    ; Display "Enter a character: "
    mov ah, 09h
    lea dx, msg1
    int 21h

    ; Read a character from user
    mov ah, 01h
    int 21h
    mov bl, al ; Store input character in BL

    ; Print a new line
    mov ah, 09h
    lea dx, msg2
    int 21h

    ; Initialize counter to 50
    mov cx, 50

printLoop:
    ; Print the character stored in BL
    mov ah, 02h
    mov dl, bl
    int 21h

    ; Decrement counter and loop until zero
    dec cx
    jnz printLoop

    ; Print "Thank you."
    mov ah, 09h
    lea dx, msg3
```

int 21h

; Exit program

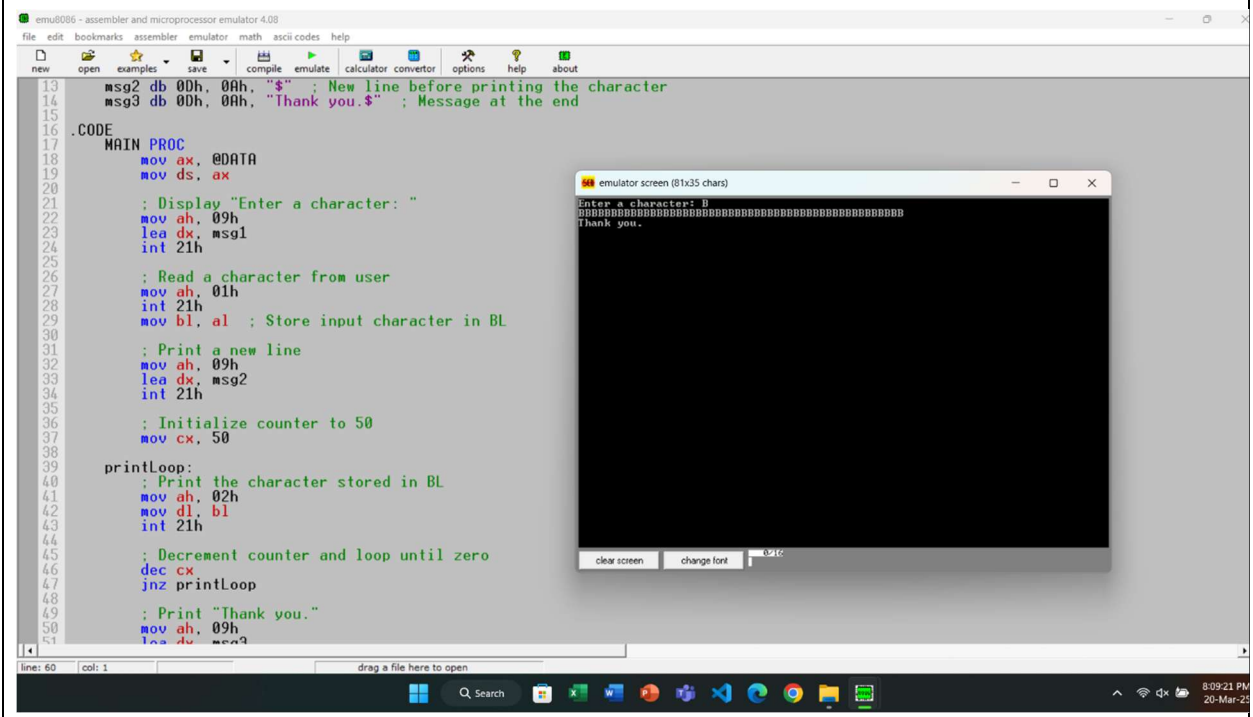
mov ah, 4Ch

int 21h

MAIN ENDP

END MAIN

Output:



```
13 msg2 db 0Dh, 0Ah, "$" ; New line before printing the character
14 msg3 db 0Dh, 0Ah, "Thank you.$" ; Message at the end
15
16
17 .CODE
18 MAIN PROC
19     mov ax, @DATA
20     mov ds, ax
21
22     ; Display "Enter a character: "
23     mov ah, 09h
24     lea dx, msg1
25     int 21h
26
27     ; Read a character from user
28     mov ah, 01h
29     int 21h
30     mov bl, al ; Store input character in BL
31
32     ; Print a new line
33     mov ah, 09h
34     lea dx, msg2
35     int 21h
36
37     ; Initialize counter to 50
38     mov cx, 50
39
40     printLoop:
41     ; Print the character stored in BL
42     mov ah, 02h
43     mov dl, bl
44     int 21h
45
46     ; Decrement counter and loop until zero
47     dec cx
48     jnz printLoop
49
50     ; Print "Thank you."
51     mov ah, 09h
52     lea dx, msg3
53     int 21h
54
55     ; Exit program
56     mov ah, 4Ch
57     int 21h
58
59
60
61
```

emulator screen (81x35 chars)

```
Enter a character: B
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
Thank you.
```

clear screen change font 8x16

line: 60 col: 1 drag a file here to open

8:09:21 PM 20-Mar-25

Task-07:

```
.MODEL small
.STACK 100h
.DATA
    msg1 db "Enter two characters: $"
    msg2 db 0Dh, 0Ah, "$" ; New line

.CODE
MAIN PROC
    mov ax, @DATA
    mov ds, ax

    ; Display "Enter two characters: "
    mov ah, 09h
    lea dx, msg1
    int 21h

    ; Read first character
    mov ah, 01h
    int 21h
    mov bl, al ; Store first character in BL

    ; Read second character
    mov ah, 01h
    int 21h
    mov bh, al ; Store second character in BH

    ; Print a new line
    mov ah, 09h
    lea dx, msg2
    int 21h

    ; Print first character
    mov ah, 02h
    mov dl, bl
    int 21h

    ; Print second character
    mov ah, 02h
    mov dl, bh
    int 21h

    ; Print a new line
    mov ah, 09h
```

```

    lea dx, msg2
    int 21h

; Print first character again
    mov ah, 02h
    mov dl, bl
    int 21h

; Print second character again
    mov ah, 02h
    mov dl, bh
    int 21h

; Exit program
    mov ah, 4Ch
    int 21h

MAIN ENDP
END MAIN

```

Output:

The screenshot displays the emu8086 software interface. The main editor window contains the following assembly code:

```

16  MAIN PROC
17      mov ax, @DATA
18      mov ds, ax
19
20      ; Display "Enter two characters: "
21      mov ah, 09h
22      lea dx, msg1
23      int 21h
24
25      ; Read first character
26      mov ah, 01h
27      int 21h
28      mov bl, al ; Store first character in BL
29
30      ; Read second character
31      mov ah, 01h
32      int 21h
33      mov bh, al ; Store second character in BH
34
35      ; Print a new line
36      mov ah, 09h
37      lea dx, msg2
38      int 21h
39
40      ; Print first character
41      mov ah, 02h
42      mov dl, bl
43      int 21h
44
45      ; Print second character
46      mov ah, 02h
47      mov dl, bh
48      int 21h
49
50      ; Print a new line
51      mov ah, 09h
52      lea dx, msg2
53      int 21h

```

The 'emulator screen (84x33 chars)' window shows the program's execution output:

```

Enter two characters: HI
HI
HI

```

The status bar at the bottom indicates 'line: 71 col: 1' and 'drag a file here to open'. The Windows taskbar at the very bottom shows the date and time as '8:10:03 PM 20-Mar-25'.