

1 (a) Write a program in assembly language to print single character on screen.

ORG 100h ; Origin, to specify that the program starts at 100h (COM file format)

; Print "Enter the input: "

MOV AH, 09h ; DOS function 09h: print string

MOV DX, OFFSET msg_enter_input ; Load address of the string

INT 21h ; Call DOS interrupt to print the string

; Read a single character from user

MOV AH, 01h ; DOS function 01h: read single character

INT 21h ; Call DOS interrupt to get the character

MOV BL, AL ; Store the input character in BL register

; Print "The entered input is: "

MOV AH, 09h ; DOS function 09h: print string

MOV DX, OFFSET msg_entered_input ; Load address of the second string

INT 21h ; Call DOS interrupt to print the string

; Print the character stored in BL register

MOV DL, BL ; Move character from BL to DL for printing

MOV AH, 02h ; DOS function 02h: print single character

INT 21h ; Call DOS interrupt to print the character

; Terminate the program

MOV AH, 4Ch ; DOS function 4Ch: terminate program

INT 21h ; Call DOS interrupt to exit

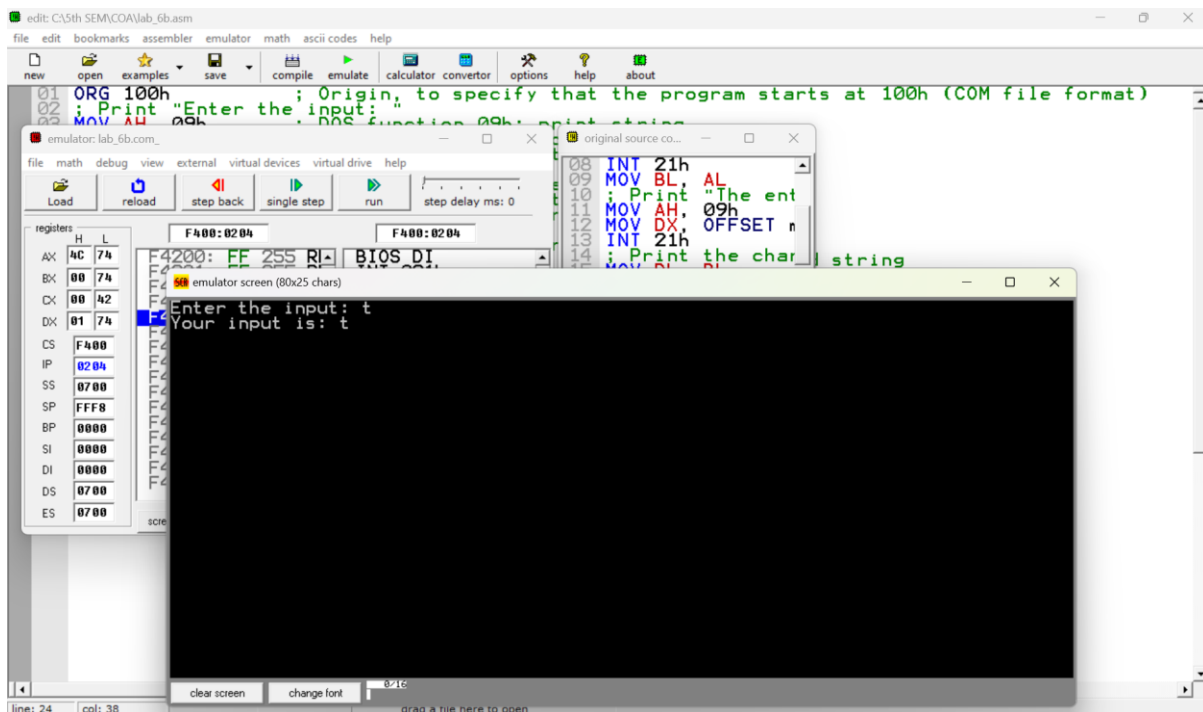
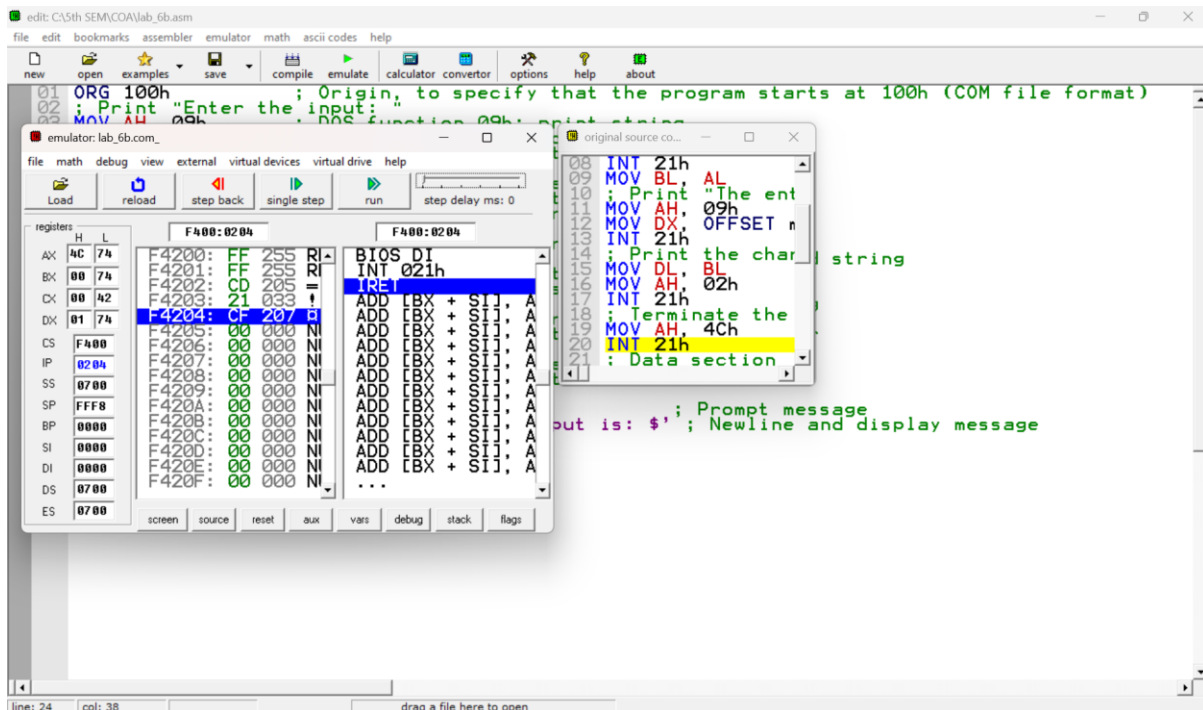
; Data section

msg_enter_input DB 'Enter the input: \$' ; Prompt message

msg_entered_input DB 0Dh, 0Ah, 'Your input is: \$' ; Newline and display message

END ; End of program

OUTPUT:



(b) Write an assembly language program to convert an upper-case letter to the corresponding lower-case letter.

ORG 100h ; Origin, to specify that the program starts at 100h (COM file format)

; Display message "Enter an uppercase letter: "

MOV DX, OFFSET msg_input ; Load the address of the message

MOV AH, 09h ; Function 09h of INT 21h is used to display a string

INT 21h ; Call DOS interrupt to print the message

; Read a single character from the user

MOV AH, 01h ; Function 01h of INT 21h is used to read a character

INT 21h ; Call DOS interrupt to get the character

MOV DL, AL ; Store the input character in AL

; Check if the character is an uppercase letter (A-Z)

CMP AL, 'A' ; Compare AL with 'A'

JL NotUpperCase ; If the input is less than 'A', it is not uppercase

CMP AL, 'Z' ; Compare AL with 'Z'

JG NotUpperCase ; If the input is greater than 'Z', it is not uppercase

; Convert the uppercase letter to lowercase

ADD AL, 20h ; Add 32 (20h) to convert uppercase to lowercase

MOV BL, AL

; Print the message "The lowercase letter is: "

MOV DX, OFFSET msg_output ; Load the address of the output message

MOV AH, 09h ; Function 09h of INT 21h is used to display a string

INT 21h ; Call DOS interrupt to print the output message

MOV AL, BL

; Print the converted lowercase letter

MOV DL, AL ; Move the lowercase letter to DL

MOV AH, 02h ; Function 02h of INT 21h is used to print a single character

INT 21h ; Call DOS interrupt to print the character

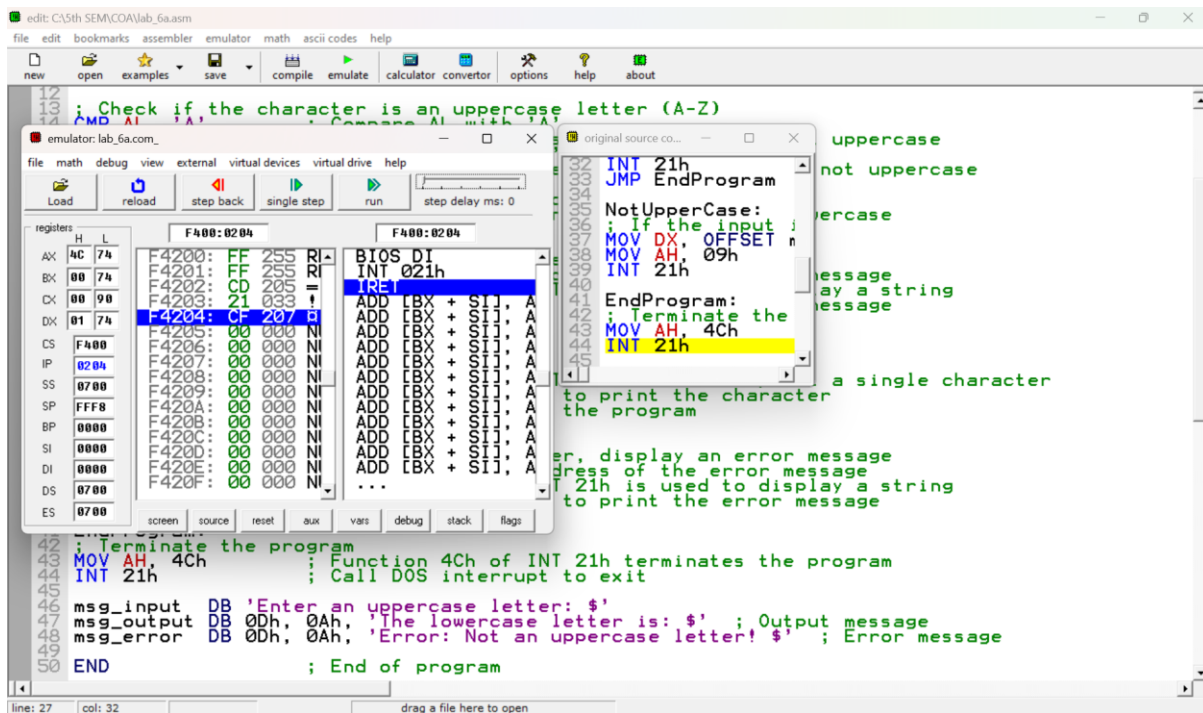
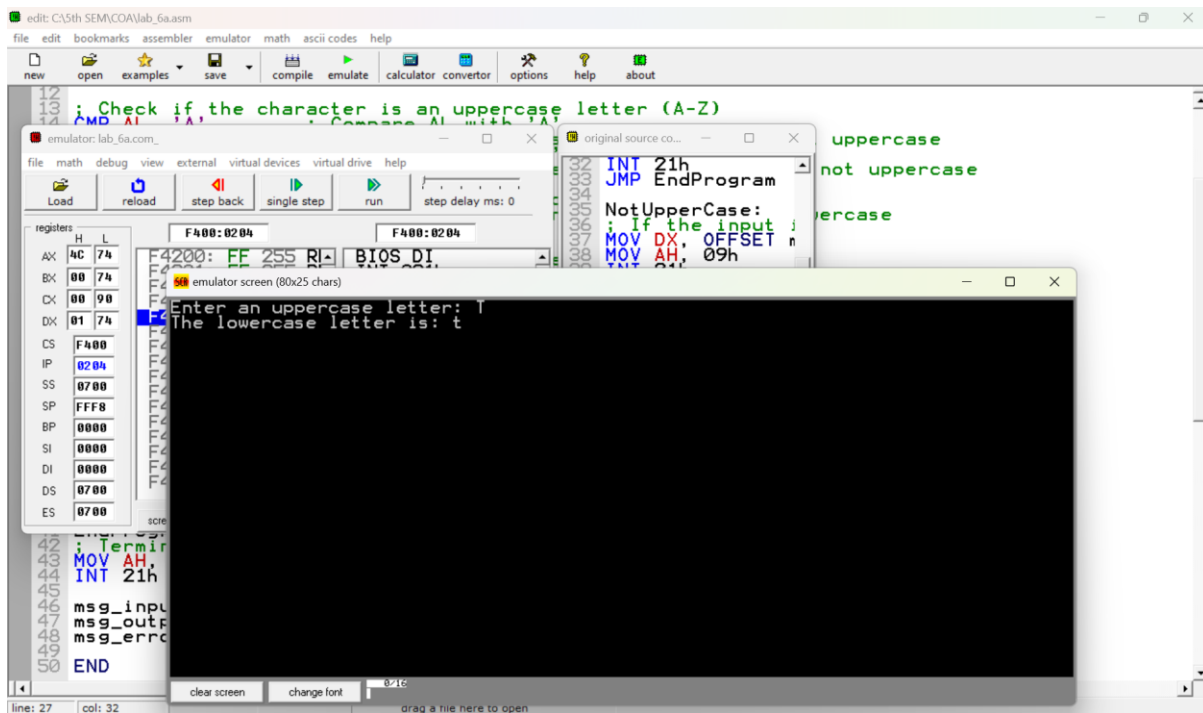
JMP EndProgram ; Jump to the end of the program

NotUpperCase:

; If the input is not an uppercase letter, display an error message

```
MOV DX, OFFSET msg_error ; Load the address of the error message
MOV AH, 09h ; Function 09h of INT 21h is used to display a string
INT 21h ; Call DOS interrupt to print the error message
EndProgram:
; Terminate the program
MOV AH, 4Ch ; Function 4Ch of INT 21h terminates the program
INT 21h ; Call DOS interrupt to exit
msg_input DB 'Enter an uppercase letter: $'
msg_output DB 0Dh, 0Ah, 'The lowercase letter is: $' ; Output message
msg_error DB 0Dh, 0Ah, 'Error: Not an uppercase letter! $' ; Error message
END ; End of program
```

OUTPUT:



Practice set:

2. (a) Write a program in assembly language to print multiple characters on screen.

```
ORG 100h      ; Origin, to specify that the program starts at 100h (COM file format)

; Print "Enter the input: "
MOV AH, 09h   ; DOS function 09h: print string
MOV DX, OFFSET msg_enter_input ; Load address of the string
INT 21h       ; Call DOS interrupt to print the string

; Read multiple characters from user
MOV AH, 0Ah   ; DOS function 0Ah: buffered input
MOV DX, OFFSET input_buffer ; Load address of the input buffer
INT 21h       ; Call DOS interrupt to read the string

; Add a $ at the end of the entered string for printing
MOV AL, '$'   ; Store $ in AL
LEA DI, input_buffer+2 ; DI points to the actual input string
MOV CL, [input_buffer+1] ; Get the count of characters entered
ADD DI, CX    ; Move DI to the end of the entered string
MOV [DI], AL  ; Insert $ at the end of the string

; Print "The entered input is: "
MOV AH, 09h   ; DOS function 09h: print string
MOV DX, OFFSET msg_entered_input ; Load address of the second string
INT 21h       ; Call DOS interrupt to print the string

; Print the entered string
LEA DX, input_buffer+2 ; Load address of the actual input (skip buffer size and count)
MOV AH, 09h   ; DOS function 09h: print string
INT 21h       ; Call DOS interrupt to print the input string

; Terminate the program
MOV AH, 4Ch   ; DOS function 4Ch: terminate program
INT 21h       ; Call DOS interrupt to exit

; Data section
msg_enter_input DB 'Enter the input: $' ; Prompt message
```

msg_entered_input DB 0Dh, 0Ah, 'The entered input is: \$' ; Newline and display message

input_buffer DB 10, 0 ; Buffer to store input: 10 max chars, initial count 0

; The actual characters will be stored starting here

; Followed by a terminator (CR)

END ; End of program

OUTPUT:

The screenshot shows an x86-64 emulator window titled 'emulator: lab_6a(practice).asm'. The main window displays assembly code with comments. A smaller window titled 'original source co...' is overlaid on the main window, showing a different version of the code. The registers window on the left shows the state of the CPU registers. The main window's assembly code includes a data section with a prompt message and a buffer, followed by a program that prints the input and terminates.

Registers window (F400:0204):

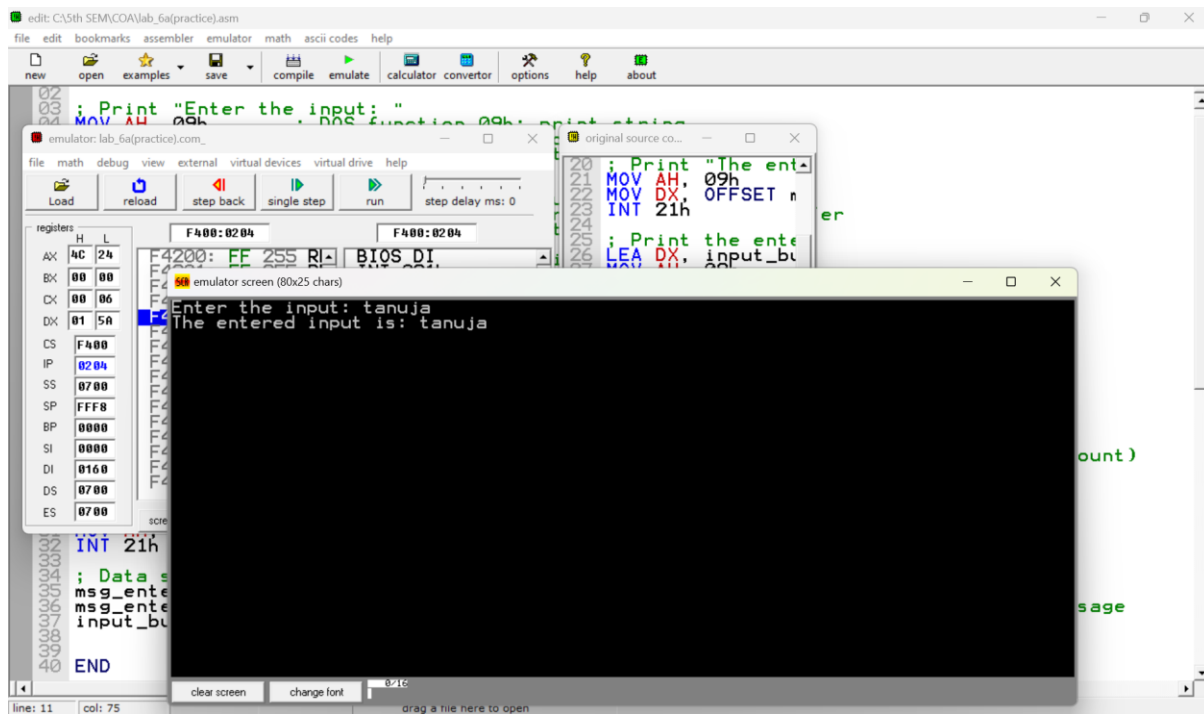
Register	Value
AX	4C 24
BC	00 00
CD	00 06
DX	01 5A
CS	F400
IP	0204
SS	0700
SP	FFF8
BP	0000
SI	0000
DI	0160
DS	0700
ES	0700

Main assembly code (lines 11-40):

```
11 ; Data section
12 msg_enter_input DB 'Enter the input: $' ; Prompt message
13 msg_entered_input DB 0Dh, 0Ah, 'The entered input is: $' ; Newline and display message
14 input_buffer DB 10, 0 ; Buffer to store input: 10 max chars, initial count 0
15 ; The actual characters will be stored starting here
16 ; Followed by a terminator (CR)
17 END ; End of program
```

Overlaid assembly code (lines 20-33):

```
20 ; Print "The entered input is: $"
21 MOV AH, 09h
22 MOV DX, OFFSET msg_entered_input
23 INT 21h
24 ; Print the entered input
25 LEA DX, input_buffer
26 MOV AH, 09h
27 INT 21h
28 ; Terminate the program
29 MOV AH, 4Ch
30 INT 21h
```



(b) Write an assembly language program to convert a lower-case letter to the corresponding upper-case letter.

ORG 100h ; Origin, to specify that the program starts at 100h (COM file format)

; Display message "Enter a lowercase letter: "

MOV DX, OFFSET msg_input ; Load the address of the message

MOV AH, 09h ; Function 09h of INT 21h is used to display a string

INT 21h ; Call DOS interrupt to print the message

; Read a single character from the user

MOV AH, 01h ; Function 01h of INT 21h is used to read a character

INT 21h ; Call DOS interrupt to get the character

MOV DL, AL ; Store the input character in AL

; Check if the character is a lowercase letter (a-z)

CMP AL, 'a' ; Compare AL with 'a'

JL NotLowerCase ; If the input is less than 'a', it is not lowercase

CMP AL, 'z' ; Compare AL with 'z'

JG NotLowerCase ; If the input is greater than 'z', it is not lowercase


```

; Convert the lowercase letter to uppercase

SUB AL, 20h    ; Subtract 32 (20h) to convert lowercase to uppercase

MOV BL,AL

; Print the message "The uppercase letter is: "

MOV DX, OFFSET msg_output ; Load the address of the output message

MOV AH, 09h    ; Function 09h of INT 21h is used to display a string

INT 21h        ; Call DOS interrupt to print the output message

MOV AL, BL

; Print the converted uppercase letter

MOV DL, AL     ; Move the uppercase letter to DL

MOV AH, 02h    ; Function 02h of INT 21h is used to print a single character

INT 21h        ; Call DOS interrupt to print the character

JMP EndProgram ; Jump to the end of the program

NotLowerCase:

; If the input is not a lowercase letter, display an error message

MOV DX, OFFSET msg_error ; Load the address of the error message

MOV AH, 09h    ; Function 09h of INT 21h is used to display a string

INT 21h        ; Call DOS interrupt to print the error message

EndProgram:

; Terminate the program

MOV AH, 4Ch    ; Function 4Ch of INT 21h terminates the program

INT 21h        ; Call DOS interrupt to exit

msg_input DB 'Enter a lowercase letter: $'

msg_output DB 0Dh, 0Ah, 'The uppercase letter is: $' ; Output message

msg_error DB 0Dh, 0Ah, 'Error: Not a lowercase letter! $' ; Error message

END            ; End of program

```

[illegible]

The screenshot displays a Windows environment with several overlapping windows related to x86 assembly development and emulation.

- Main Window (emulator: lab_6b(practice).asm):** Shows assembly code for a program that checks if a character is a lowercase letter. The code includes comments in green and assembly instructions in blue. The status bar at the bottom indicates "line: 31 col: 30".
- Assembly Code Window (original source co...):** Displays the assembly code in a different format, showing instructions like "INT 21h", "JMP EndProgram", "NotLowerCase:", "if the input", "MOV DX, OFFSET", and "MOV AH, 09h".
- Registers and Memory Window:** Shows the state of registers (AX, BX, CX, DX, SI, DI, DS, ES) and memory locations (F400:0204, F400:0204). The registers are listed on the left, and the memory locations are listed on the right.
- Emulator Screen Window:** Displays the output of the emulator, showing the text "Enter a lowercase letter: a" and "The uppercase letter is: A".