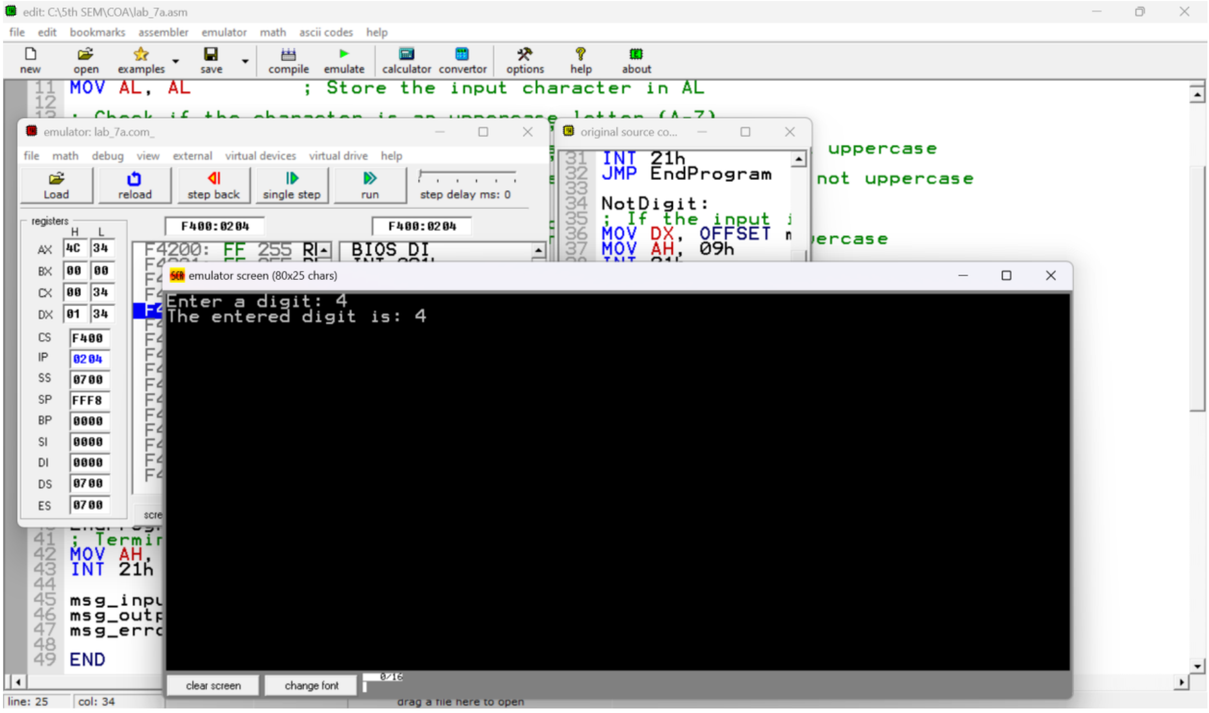
LAB ASSIGNMENT - 7

Write a program in assembly language to take a single-digit integer from the user and print it on the screen.

```
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 AL, AL        ; Store the input character in AL
; Check if the character is an uppercase letter (A-Z)
CMP AL, '0'       ; Compare AL with 'A'
JL NotDigit       ; If the input is less than 'A', it is not uppercase
CMP AL, '9'       ; Compare AL with 'Z'
JG NotDigit       ; If the input is greater than 'Z', it is not uppercase
mov cl,al
; Convert the uppercase letter to lowercase
;ADD AL, 20h      ; Add 32 (20h) to convert uppercase to lowercase
; 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
; Print the converted lowercase letter
MOV DL, CL        ; 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
NotDigit:
; 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 a digit: $'
msg_output DB 0Dh, 0Ah, 'The entered digit is: $'  ; Output message
msg_error  DB 0Dh, 0Ah, 'Error: Not a digit! $'  ; Error message

END                ; End of program
```

OUTPUT:



edit: C:\5th SEM\COA\lab_7a.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator: lab_7a.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
original source co...
emulator screen (80x25 chars)
clear screen change font
drag a file here to open

```
MOV AL, AL            ; Store the input character in AL
```

```
INT 21h
JMP EndProgram

NotDigit:
; If the input i
MOV DX, OFFSET m
MOV AH, 09h
```

registers

	H	L
AX	4C	34
BX	00	00
CX	00	34
DX	01	34
CS	F400	
IP	0204	
SS	0700	
SP	FFF8	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

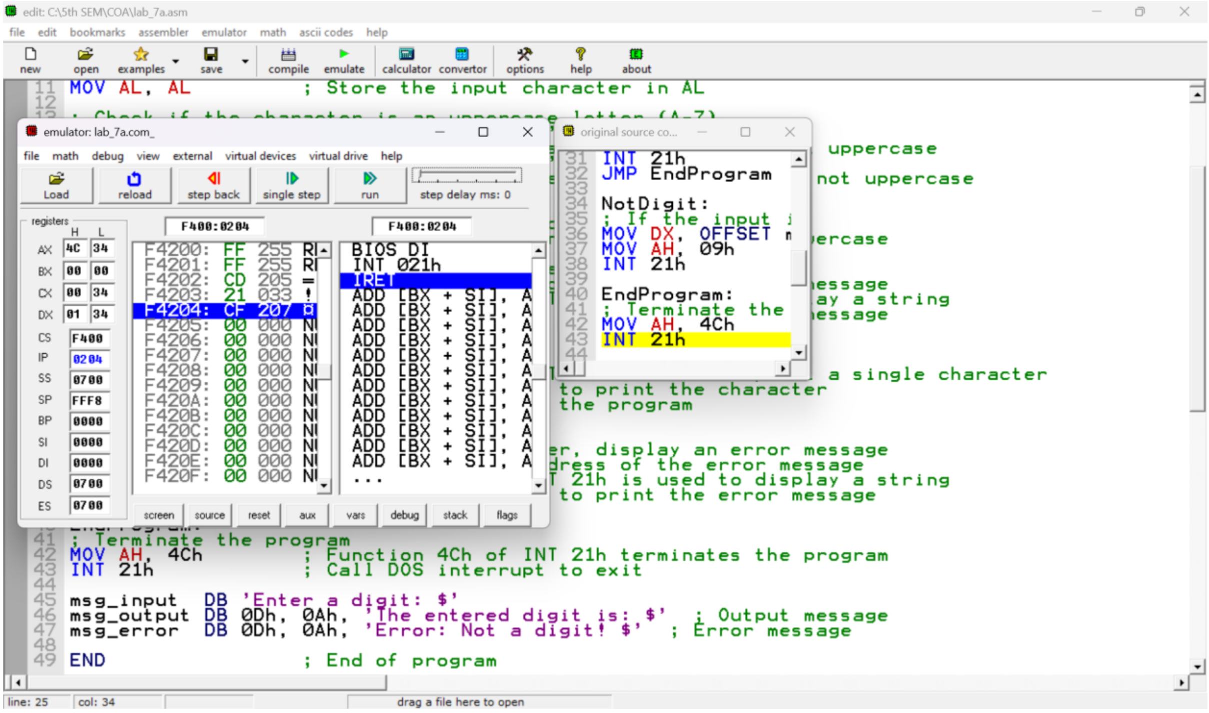
F400:0204 F400:0204

```
F4200: FF 255 RI
```

```
Enter a digit: 4
The entered digit is: 4
```

```
; Termir
MOV AH,
INT 21h

msg_inpu
msg_outp
msg_errc

END
```

line: 25 col: 34


edit: C:\5th SEM\COA\lab_7a.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator: lab_7a.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
original source co...
screen source reset aux vars debug stack flags
line: 25 col: 34 drag a file here to open

```
MOV AL, AL              ; Store the input character in AL

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

```
registers
     H    L
AX   4C   34
BX   00   00
CX   00   34
DX   01   34
CS   F400
IP   0204
SS   0700
SP   FFF8
BP   0000
SI   0000
DI   0000
DS   0700
ES   0700
```

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

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

```
INT 21h
JMP EndProgram

NotDigit:
; If the input i
MOV DX, OFFSET m
MOV AH, 09h
INT 21h

EndProgram:
; Terminate the
MOV AH, 4Ch
INT 21h
```

```
uppercase
not uppercase
ercase
essage
ay a string
essage
a single character
to print the character
the program

er, display an error message
dress of the error message
T 21h is used to display a string
to print the error message
```

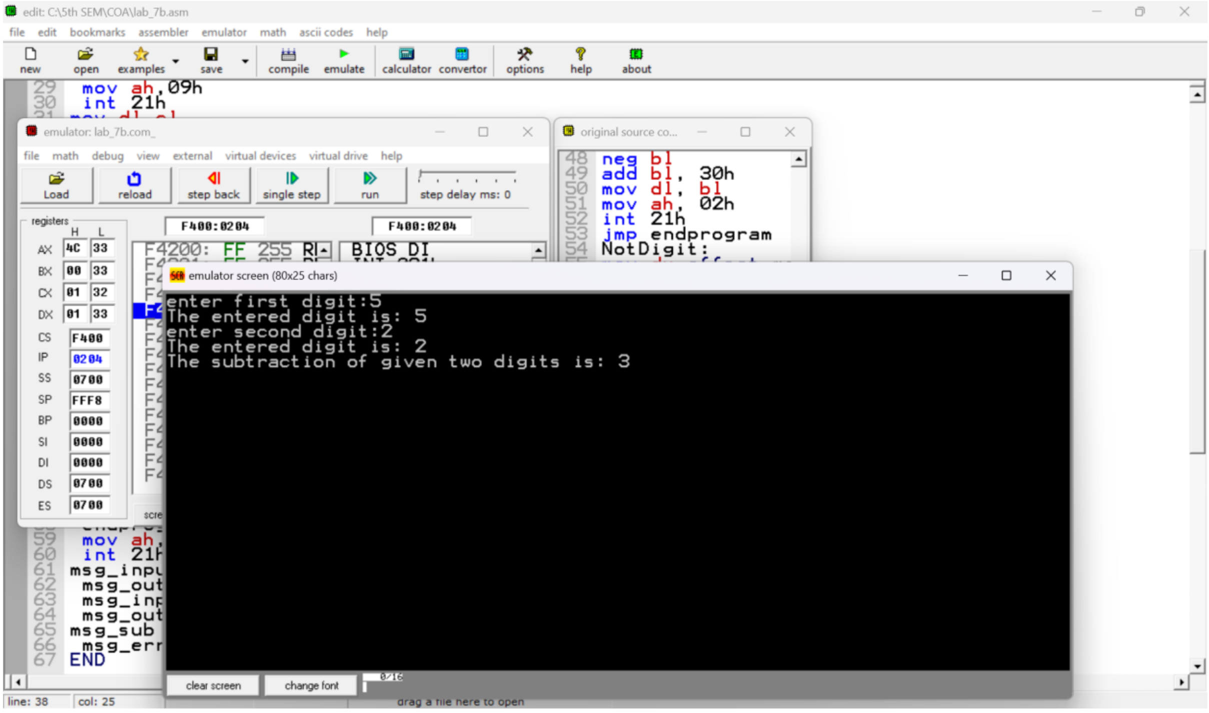
```
; Terminate the program
MOV AH, 4Ch             ; Function 4Ch of INT 21h terminates the program
INT 21h                 ; Call DOS interrupt to exit

msg_input  DB 'Enter a digit: $'
msg_output DB 0Dh, 0Ah, 'The entered digit is: $'  ; Output message
msg_error  DB 0Dh, 0Ah, 'Error: Not a digit! $'   ; Error message

END                     ; End of program
```

Write a program in assembly language to take two single-digit integers from the user and print the result of subtraction on the screen.

```
org 100h
 mov dx,offset msg_input1
 mov ah,09h
 int 21h
mov ah,01h
 int 21h
mov bl,al
 cmp al,'0'
 jl NotDigit
 cmp al,'9'
 jg NotDigit
mov dx,offset msg_output1
 mov ah,09h
 int 21h
mov dl,bl
 mov ah,02h
 int 21h
mov dx,offset msg_input2
 mov ah,09h
 int 21h
 mov ah,01h
 int 21h
mov cl,al
 cmp al,'0'
 jl NotDigit
 cmp al,'9'
 jg NotDigit
mov dx,offset msg_output2
 mov ah,09h
 int 21h
mov dl,cl
 mov ah,02h
 int 21h
 mov dx,offset msg_sub
 mov ah,09h
int 21h
sub bl,cl
 js NegativeResult
 add bl,30h
 mov dl,bl
 mov ah,02h
 int 21h
jmp endprogram
NegativeResult:
 mov dl, '-'
 mov ah, 02h
 int 21h
neg bl
add bl, 30h
mov dl, bl
 mov ah, 02h
 int 21h
 jmp endprogram
 NotDigit:
 mov dx,offset msg_error
 mov ah,09h
 int 21h
 endprogram:
 mov ah,4Ch
 int 21h
msg_input1 DB "enter first digit:$"
```

```
 msg_output1 Db 0dh,0ah,"The entered digit is: $"
 msg_input2 DB 0dh,0ah,"enter second digit:$"
 msg_output2 Db 0dh,0ah,"The entered digit is: $"
msg_sub db 0dh,0ah,"The subtraction of given two digits is: $"
 msg_error db 0dh,0ah,"Error: Not a digit!$ "
END
```

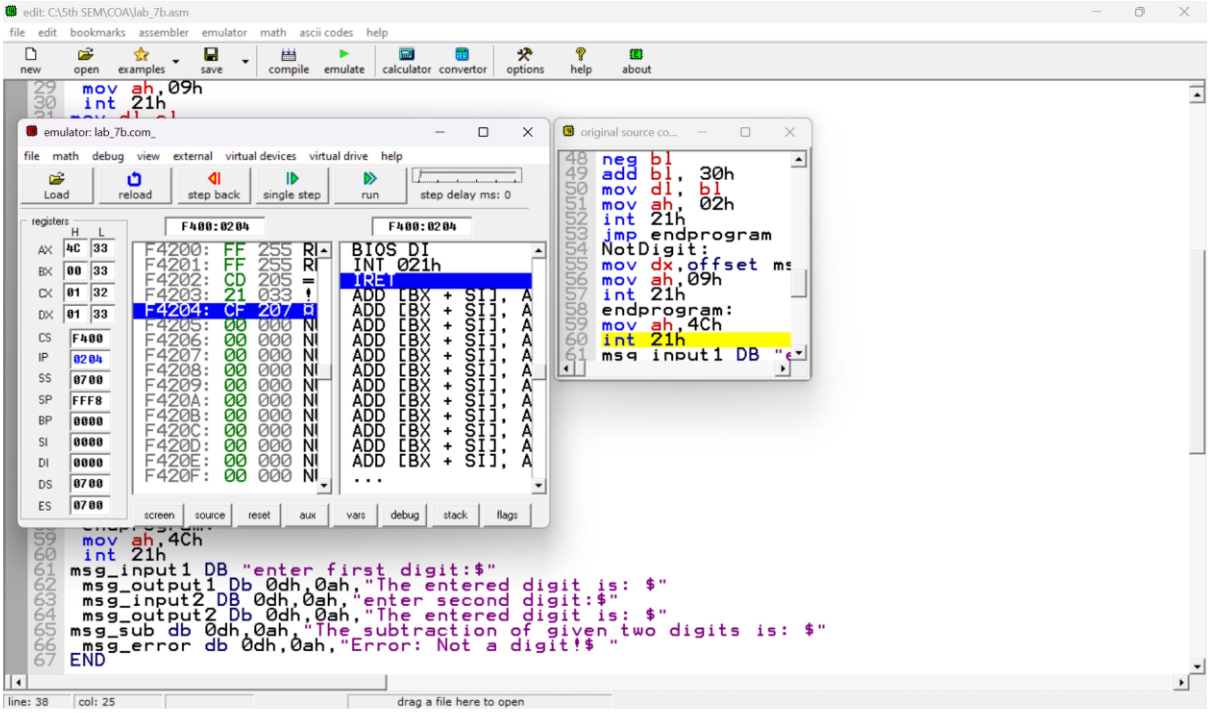
OUTPUT:


edit: C:\5th SEM\COA\lab_7b.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator: lab_7b.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
original source co...
emulator screen (80x25 chars)
clear screen change font
line: 38 col: 25
drag a file here to open

```
enter first digit:5
The entered digit is: 5
enter second digit:2
The entered digit is: 2
The subtraction of given two digits is: 3
```

```
neg bl
add bl, 30h
mov dl, bl
mov ah, 02h
int 21h
jmp endprogram
NotDigit:
```

registers	H	L
AX	4C	33
BX	00	33
CX	01	32
DX	01	33
CS	F400	
IP	0204	
SS	0700	
SP	FFF8	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	



```
 mov ah,09h
 int 21h
```

registers

```
    H  L
AX  4C 33
BX  00 33
CX  01 32
DX  01 33
CS  F400
IP  0204
SS  0700
SP  FFF8
BP  0000
SI  0000
DI  0000
DS  0700
ES  0700
```

F400:0204

```
F4200: FF 255 RI
F4201: FF 255 RI
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ¤
F4205: 00 000 NI
F4206: 00 000 NI
F4207: 00 000 NI
F4208: 00 000 NI
F4209: 00 000 NI
F420A: 00 000 NI
F420B: 00 000 NI
F420C: 00 000 NI
F420D: 00 000 NI
F420E: 00 000 NI
F420F: 00 000 NI
```

F400:0204

```
BIOS DI
INT 021h
IRET
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
...
```

```
neg bl
add bl, 30h
mov dl, bl
mov ah, 02h
int 21h
jmp endprogram
NotDigit:
mov dx,offset ms
mov ah,09h
int 21h
endprogram:
mov ah,4Ch
int 21h
msg input1 DB "e
```

```
 mov ah,4Ch
 int 21h
msg_input1 DB "enter first digit:$"
 msg_output1 Db 0dh,0ah,"The entered digit is: $"
 msg_input2 DB 0dh,0ah,"enter second digit:$"
 msg_output2 Db 0dh,0ah,"The entered digit is: $"
msg_sub db 0dh,0ah,"The subtraction of given two digits is: $"
 msg_error db 0dh,0ah,"Error: Not a digit!$ "
END
```

line: 38 col: 25