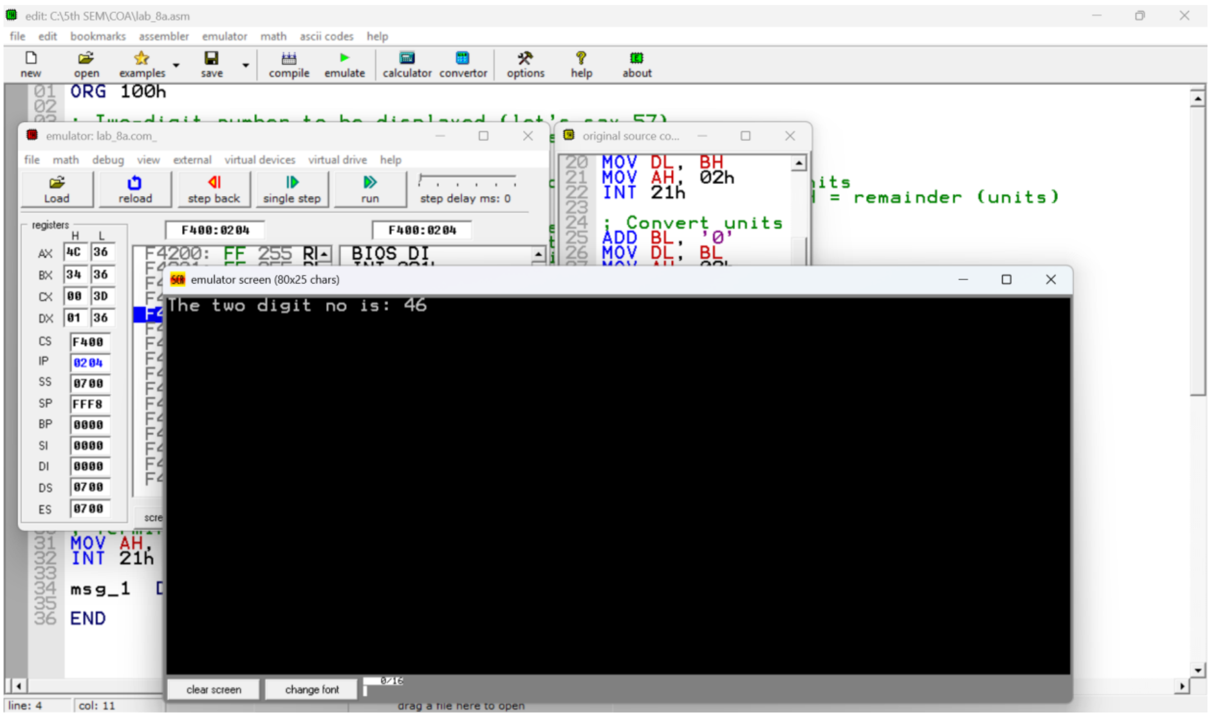
1. Write a program in assembly language to display a two-digit number on the screen. The two-digits number is required to be taken in the program itself.

```
ORG 100h
; Two-digit number to be displayed (let's say 57)
MOV AL, 64    ; Load the two-digit number into AL
; Split the number into tens and units
MOV BL, 10        ; Set divisor to 10 to separate tens and units
DIV BL            ; Divide AL by 10, AL = quotient (tens), AH = remainder (units)
; Store the quotient (tens) and remainder (units)
MOV BH, AL        ; Store the tens digit in BH
MOV BL, AH        ; Store the units digit in BL
MOV DX, OFFSET msg_1
MOV AH, 09h
INT 21h
; Convert tens digit to ASCII
ADD BH, '0'       ; Convert the tens digit to ASCII
MOV DL, BH        ; Move the ASCII tens digit to DL for printing
MOV AH, 02h       ; DOS interrupt to print a character
INT 21h           ; Print the tens digit
; Convert units digit to ASCII
ADD BL, '0'       ; Convert the units digit to ASCII
MOV DL, BL        ; Move the ASCII units digit to DL for printing
MOV AH, 02h       ; DOS interrupt to print a character
INT 21h           ; Print the units digit

; Terminate the program
MOV AH, 4Ch       ; DOS interrupt to exit the program
INT 21h
msg_1  DB 'The two digit no is : $'
END
```

OUTPUT:

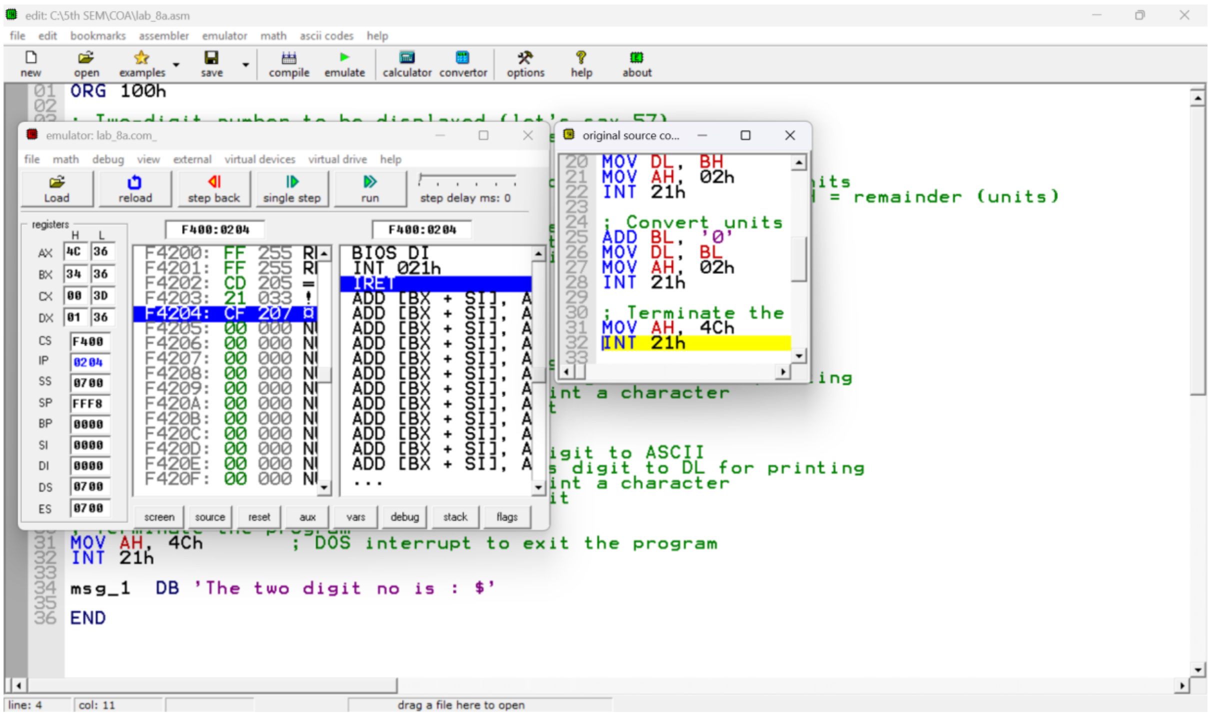


```
ORG 100h
; Two-digit number to be displayed (let's say 57)
```

```
MOV DL, BH
MOV AH, 02h
INT 21h

; Convert units
ADD BL, '0'
MOV DL, BL
```

```
The two digit no is: 46
```

```
MOV AH,
INT 21h

msg_1

END
```



```
AX 4C 36
BX 34 36
CX 00 3D
DX 01 36
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
...
```

```
MOV DL, BH
MOV AH, 02h
INT 21h

; Convert units
ADD BL, '0'
MOV DL, BL
MOV AH, 02h
INT 21h

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

```
ORG 100h
; Two-digit number to be displayed (let's say 57)
...
; Terminate the program
MOV AH, 4Ch                ; DOS interrupt to exit the program
INT 21h

msg_1  DB 'The two digit no is : $'

END
```

Practice Set:

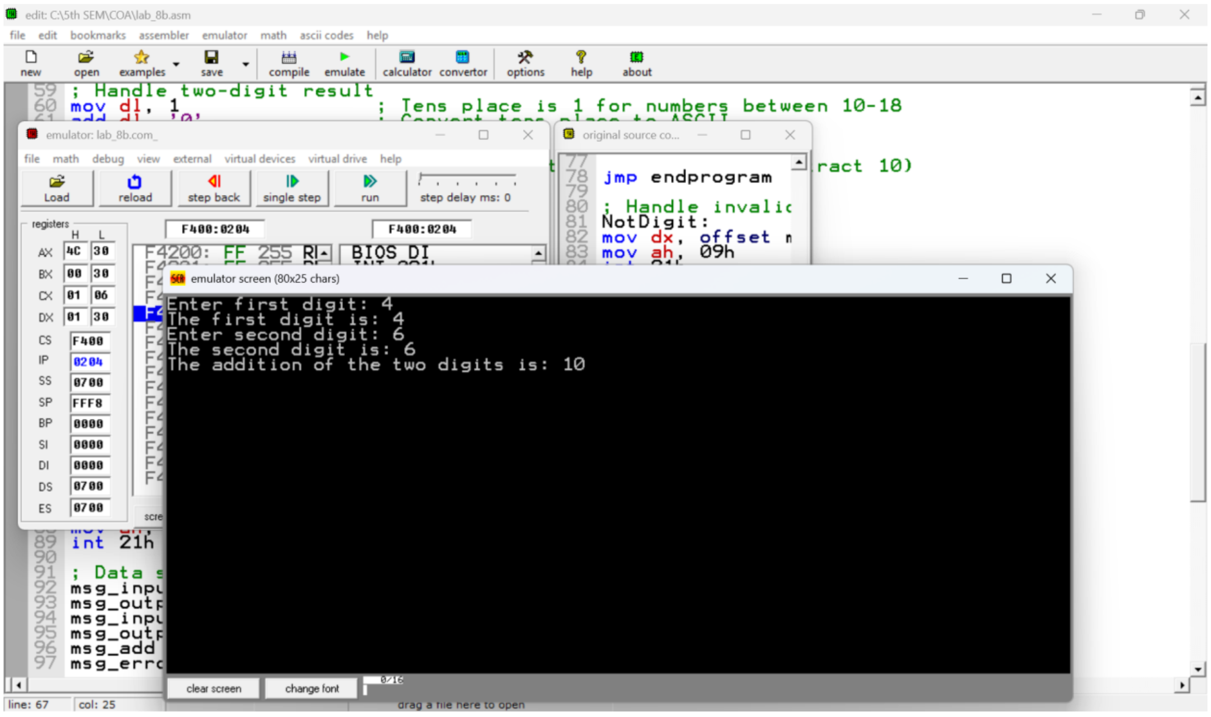
2. Write an assembly language program to take two single-digit integers from the user and print the result of addition on the screen.

```
ORG 100h

; Prompt for the first single-digit number

mov dx, offset msg_input1

mov ah, 09h

int 21h

; Get first digit

mov ah, 01h

int 21h

mov bl, al              ; Store first digit in BL

cmp al, '0'             ; Check if it's a valid digit

jl NotDigit

cmp al, '9'

jg NotDigit

; Display the first digit

mov dx, offset msg_output1

mov ah, 09h

int 21h

mov dl, bl

mov ah, 02h

int 21h

; Prompt for the second single-digit number

mov dx, offset msg_input2

mov ah, 09h

int 21h

; Get second digit

mov ah, 01h

int 21h

mov cl, al              ; Store second digit in CL

cmp al, '0'             ; Check if it's a valid digit
```

```
jl NotDigit
cmp al, '9'
jg NotDigit
; Display the second digit
mov dx, offset msg_output2
mov ah, 09h
int 21h
mov dl, cl
mov ah, 02h
int 21h
; Perform addition of the two digits
mov dx, offset msg_add
mov ah, 09h
int 21h
sub bl, '0'          ; Convert first digit from ASCII to numeric value
sub cl, '0'          ; Convert second digit from ASCII to numeric value
add bl, cl           ; Add the two digits
; Check if the result is a two-digit number (>= 10)
cmp bl, 10
jl SingleDigit        ; If less than 10, it's a single-digit result
; Handle two-digit result
mov dl, 1             ; Tens place is 1 for numbers between 10-18
add dl, '0'          ; Convert tens place to ASCII
mov ah, 02h
int 21h
sub bl, 10            ; Adjust result for ones place (subtract 10)
add bl, '0'          ; Convert ones place to ASCII
mov dl, bl
mov ah, 02h
int 21h
jmp endprogram
SingleDigit:
```

```
; Handle single-digit result
add bl, '0'             ; Convert the result to ASCII
mov dl, bl
mov ah, 02h
int 21h
jmp endprogram
; Handle invalid input
NotDigit:
mov dx, offset msg_error
mov ah, 09h
int 21h
; End the program
endprogram:
mov ah, 4Ch
int 21h
; Data section
msg_input1 DB "Enter first digit: $"
msg_output1 DB 0Dh, 0Ah, "The first digit is: $"
msg_input2 DB 0Dh, 0Ah, "Enter second digit: $"
msg_output2 DB 0Dh, 0Ah, "The second digit is: $"
msg_add DB 0Dh, 0Ah, "The addition of the two digits is: $"
msg_error DB 0Dh, 0Ah, "Error: Not a digit!$"
```

OUTPUT:



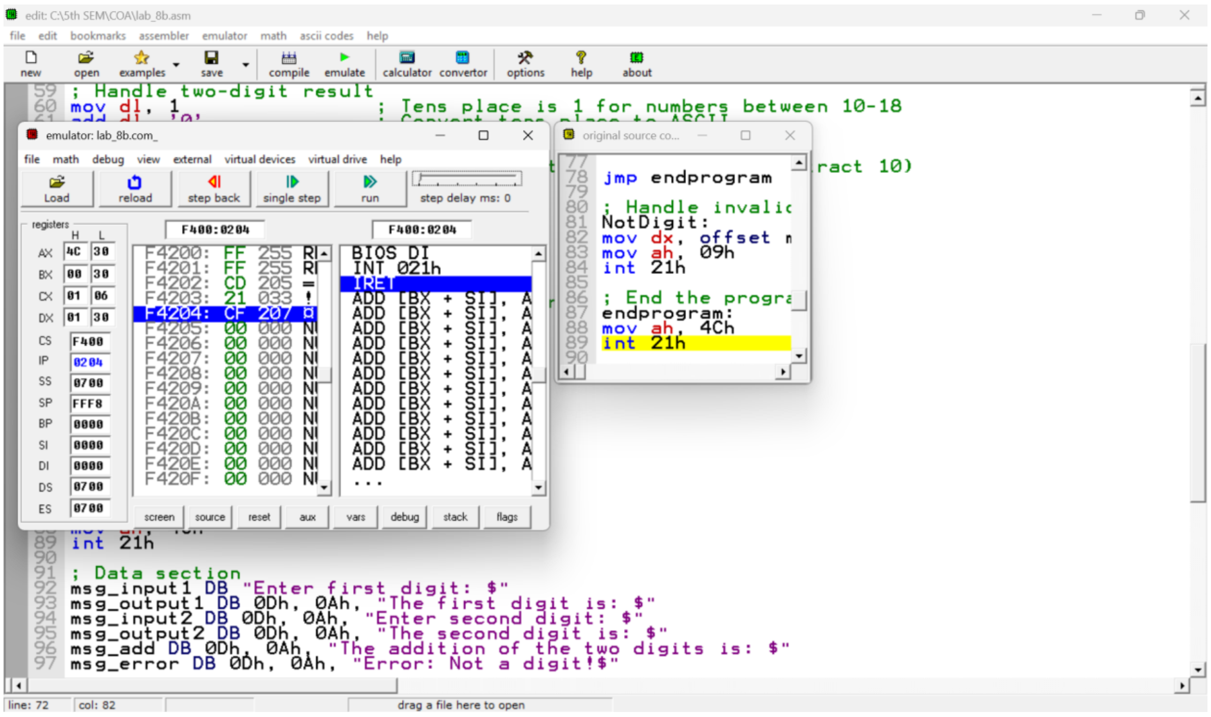
```
; Handle two-digit result
mov dl, 1                ; Tens place is 1 for numbers between 10-18
```

```
jmp endprogram

; Handle invalic
NotDigit:
mov dx, offset m
mov ah, 09h
```

Registers:

	H	L
AX	4C	30
BX	00	30
CX	01	06
DX	01	30
CS	F400	
IP	0204	
SS	0700	
SP	FFF8	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

```
Enter first digit: 4
The first digit is: 4
Enter second digit: 6
The second digit is: 6
The addition of the two digits is: 10
```



```
; Handle two-digit result
mov dl, 1                ; Tens place is 1 for numbers between 10-18
```

```
F400:0204                 F400:0204
F4200: FF 255 RI          BIOS DI
F4201: FF 255 RI          INT 021h
F4202: CD 205 =           IRET
F4203: 21 033 !           ADD [BX + SI], A
F4204: CF 207 ¤           ADD [BX + SI], A
F4205: 00 000 NI          ADD [BX + SI], A
F4206: 00 000 NI          ADD [BX + SI], A
F4207: 00 000 NI          ADD [BX + SI], A
F4208: 00 000 NI          ADD [BX + SI], A
F4209: 00 000 NI          ADD [BX + SI], A
F420A: 00 000 NI          ADD [BX + SI], A
F420B: 00 000 NI          ADD [BX + SI], A
F420C: 00 000 NI          ADD [BX + SI], A
F420D: 00 000 NI          ADD [BX + SI], A
F420E: 00 000 NI          ADD [BX + SI], A
F420F: 00 000 NI          ...
```

```
jmp endprogram

; Handle invalic
NotDigit:
mov dx, offset m
mov ah, 09h
int 21h

; End the progra
endprogram:
mov ah, 4Ch
int 21h
```

```
int 21h

; Data section
msg_input1 DB "Enter first digit: $"
msg_output1 DB 0Dh, 0Ah, "The first digit is: $"
msg_input2 DB 0Dh, 0Ah, "Enter second digit: $"
msg_output2 DB 0Dh, 0Ah, "The second digit is: $"
msg_add DB 0Dh, 0Ah, "The addition of the two digits is: $"
msg_error DB 0Dh, 0Ah, "Error: Not a digit!$"
```