

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.

```
//code
.model small
.stack 100h

.data
num1 db 5
num2 db 4
msg db 13,10,'the number is:$'

.code
start:
mov ax,@data
mov ds,ax

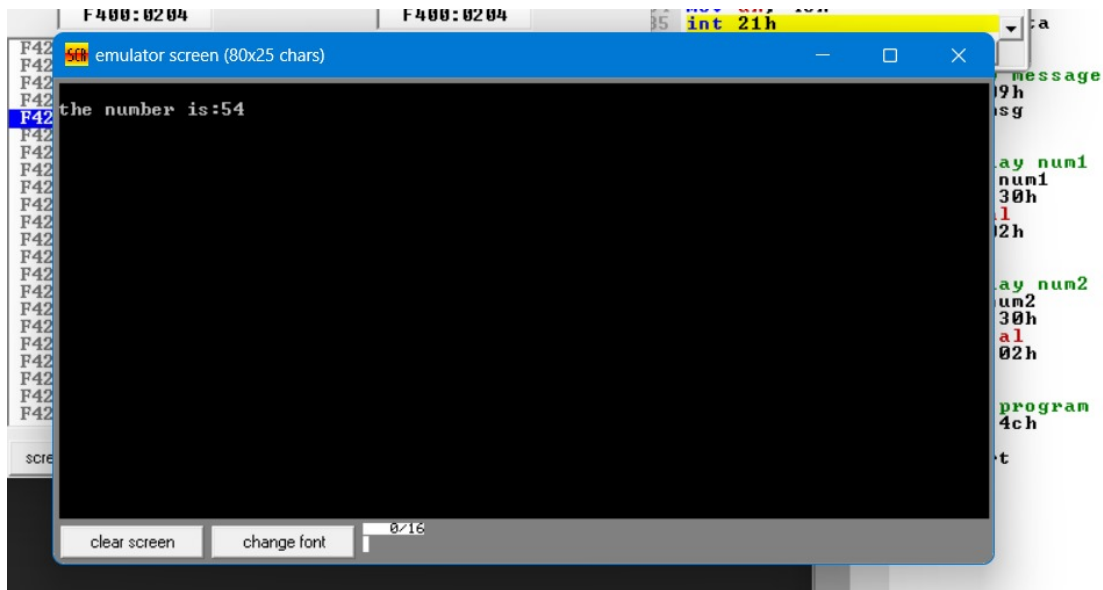
;display message
mov ah,09h
lea dx,msg
int 21h

;display num1
mov al,num1
add al,30h
mov dl,al
mov ah,02h
int 21h

;display num2
mov al,num2
add al,30h
mov dl,al
mov ah,02h
int 21h

;exit program
mov ah,4ch
int 21h
end start
```

output



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

//code

ORG 100h

_start:

```
; Display message "Enter first digit: "
MOV DX, OFFSET msg_input1
MOV AH, 09h
INT 21h
```

```
; Get the first single-digit integer from the user
MOV AH, 01h
INT 21h
CMP AL, '0'
JL InvalidInput
CMP AL, '9'
JG InvalidInput
SUB AL, '0'
MOV BL, AL
```

```
; Display message "Enter second digit: "
MOV DX, OFFSET msg_input2
MOV AH, 09h
INT 21h
```

; Get the second single-digit integer from the user

MOV AH, 01h

INT 21h

CMP AL, '0'

JL InvalidInput

CMP AL, '9'

JG InvalidInput

SUB AL, '0'

MOV BH, AL

; Perform the subtraction (BL - BH)

ADD BL, BH

; Convert the result back to ASCII

ADD BL, '0'

; Display the result message

MOV DX, OFFSET msg_output

MOV AH, 09h

INT 21h

; Display the result of the Addition

MOV DL, BL

MOV AH, 02h

INT 21h

JMP EndProgram ; End program execution

InvalidInput:

; If input is not a valid digit, display an error message

MOV DX, OFFSET msg_error

MOV AH, 09h

INT 21h

EndProgram:

; Terminate the program

MOV AH, 4Ch

INT 21h

msg_input1 DB 'Enter first digit: \$'

msg_input2 DB 0Dh, 0Ah, 'Enter second digit: \$'

msg_output DB 0Dh, 0Ah, 'The result is: \$'

msg_error DB 0Dh, 0Ah, 'Error: Invalid input! \$'

END _start

Output

The image shows a screenshot of an x86 emulator. The main window is titled "emulator screen (80x25 chars)" and displays a DOS text-mode interface. The text on the screen is:

```
Enter first digit: 3
Enter second digit: 5
The result is: 8
```

Below the text, there are two buttons: "clear screen" and "change font". To the right of the "change font" button is a small text box containing "0/16".

The background of the emulator shows assembly code. The code is organized into columns, with addresses on the left and instructions on the right. The instructions are color-coded: blue for instructions, red for registers, and green for constants or labels. The code includes instructions like `INT 21h`, `MOU AH, 4Ch`, and `INT 21h`.