COA-LAB-9(AP22110011637)

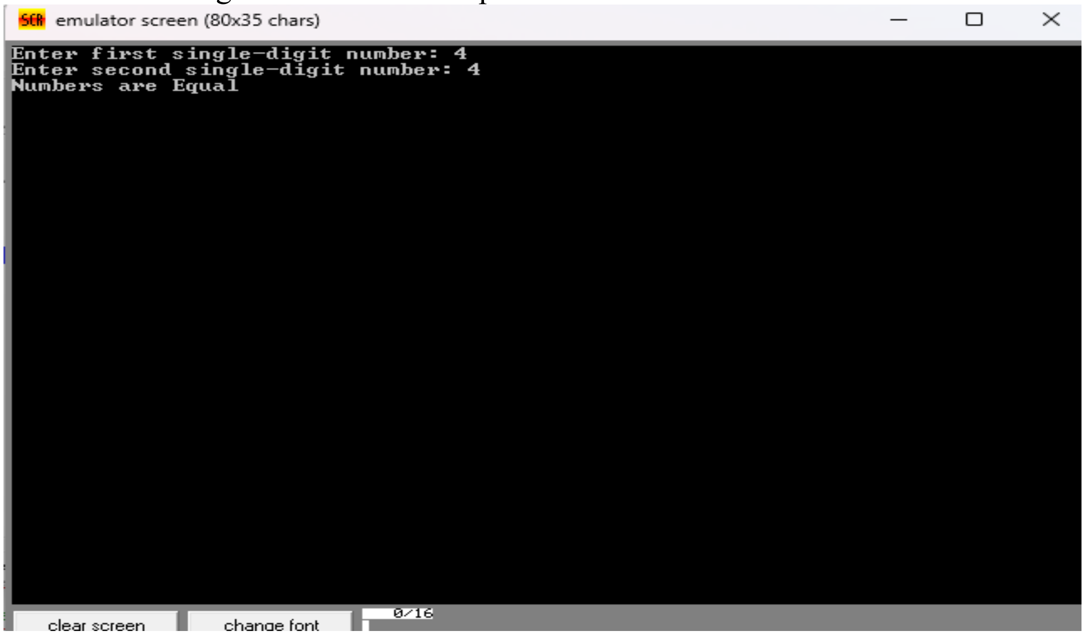
1. Write a program in assembly language to take two single-digit numbers as input and display whether they are equal or not.

Code:

```
.model small
.stack 100h
.data
   MsgEq db 'Numbers are Equal$'
   MsgUneq db 'Numbers are Unequal$'
   Prompt1 db 'Enter first single-digit number: $'
   Prompt2 db 'Enter second single-digit number: $'
   NewLine db 13, 10, '$'

.code
main proc
   mov ax, @data
   mov ds, ax

   ; Prompt for the first number
   mov dx, offset Prompt1
   mov ah, 9
   int 21h

   ; Read the first number
   mov ah, 1
   int 21h
   sub al, '0'
   mov bl, al

   ; Print a newline before the second prompt
   mov dx, offset NewLine
   mov ah, 9
   int 21h

   ; Prompt for the second number
   mov dx, offset Prompt2
   mov ah, 9
   int 21h

   ; Read the second number
   mov ah, 1
   int 21h
   sub al, '0'
   mov cl, al

   ; Compare the numbers
   cmp bl, cl
   je EQUAL

   ; If not equal
```

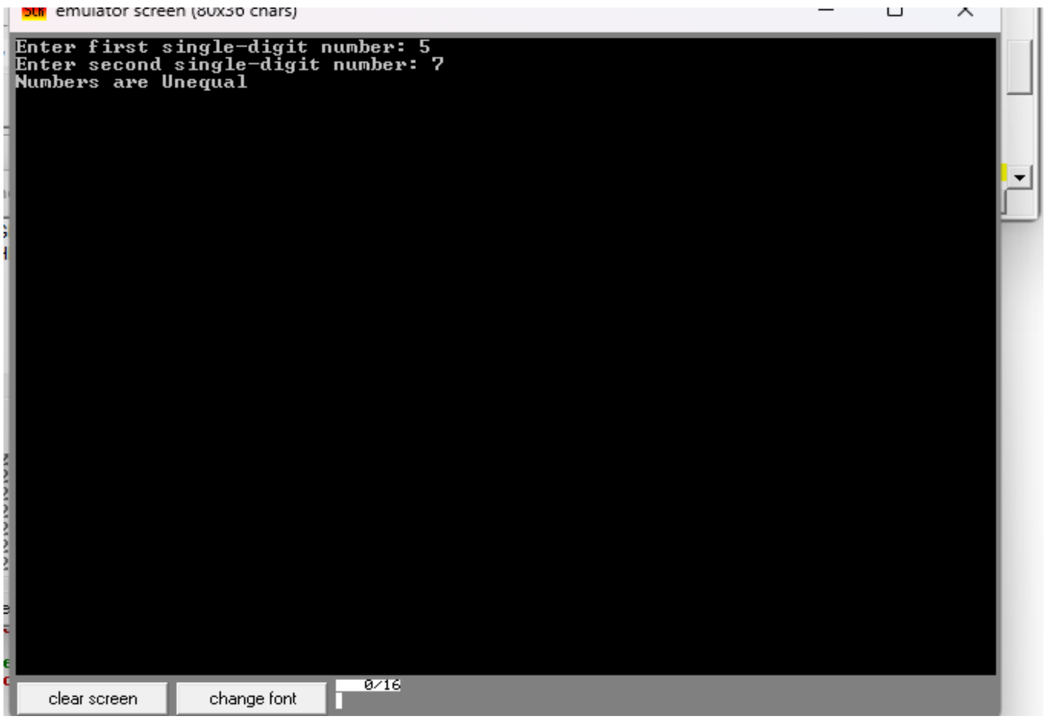
```
    mov dx, offset NewLine
    mov ah, 9
    int 21h

    mov dx, offset MsgUneq
    mov ah, 9
    int 21h
    jmp EXIT

EQUAL:
    mov dx, offset NewLine
    mov ah, 9
    int 21h

    mov dx, offset MsgEq
    mov ah, 9
    int 21h

EXIT:
    mov ah, 4Ch
    int 21h
main endp
end main
```

When the two digits are same the output like that:



```
Enter first single-digit number: 4
Enter second single-digit number: 4
Numbers are Equal
```

When the two digits are different then the output is like that:



```
Enter first single-digit number: 5
Enter second single-digit number: 7
Numbers are Unequal
```

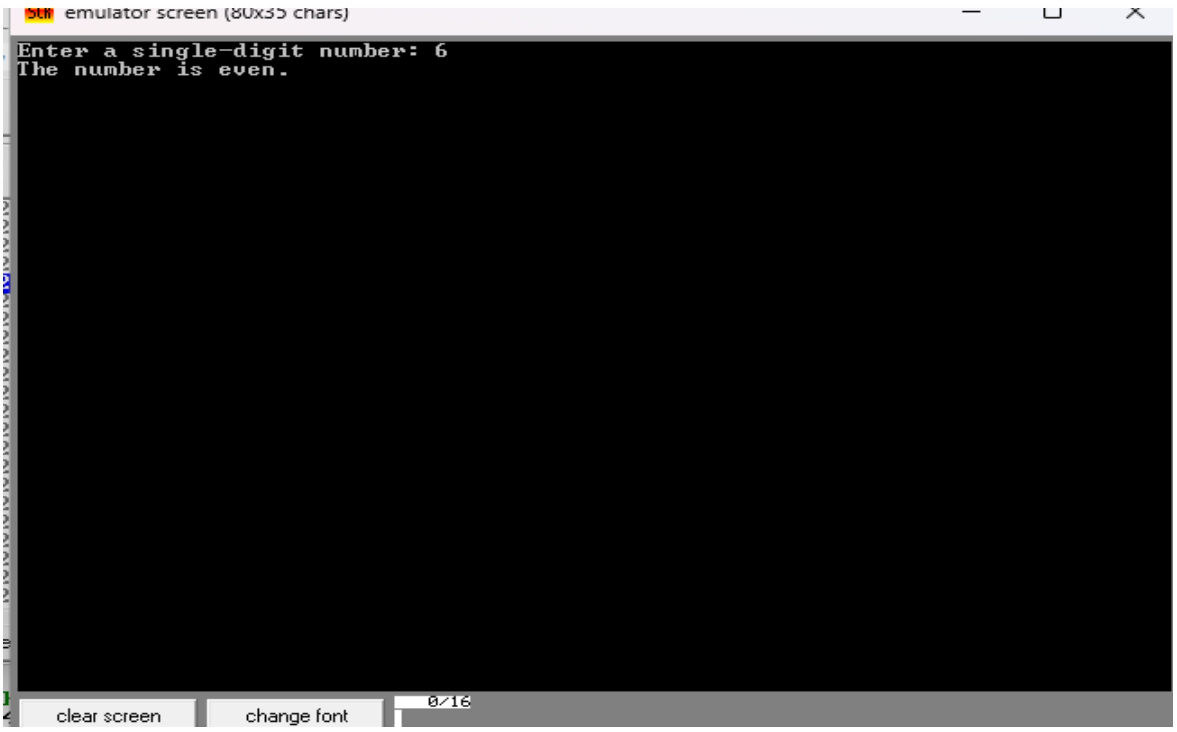
2. Write a program in assembly language to check whether a single-digit number is odd or even.

Code:

```
.stack 100h
.data
   prompt db 'Enter a single-digit number: $'
   even_msg db 'The number is even.$'
   odd_msg  db 'The number is odd.$'
   newline db 13, 10, '$'   ; Newline for output

.code
main proc
   ; Initialize the data segment
   mov ax, @data
   mov ds, ax

   ; Display prompt for user input
   mov dx, offset prompt
   mov ah, 9
   int 21h

   ; Get the user input (single character)
   mov ah, 1        ; Function 1: Input single character
   int 21h
   sub al, '0'      ; Convert ASCII input to numeric value (0-9)

   ; Validate that the input is a single digit (0-9)
   cmp al, 9
   ja InvalidInput  ; If input is greater than 9, it's invalid
```

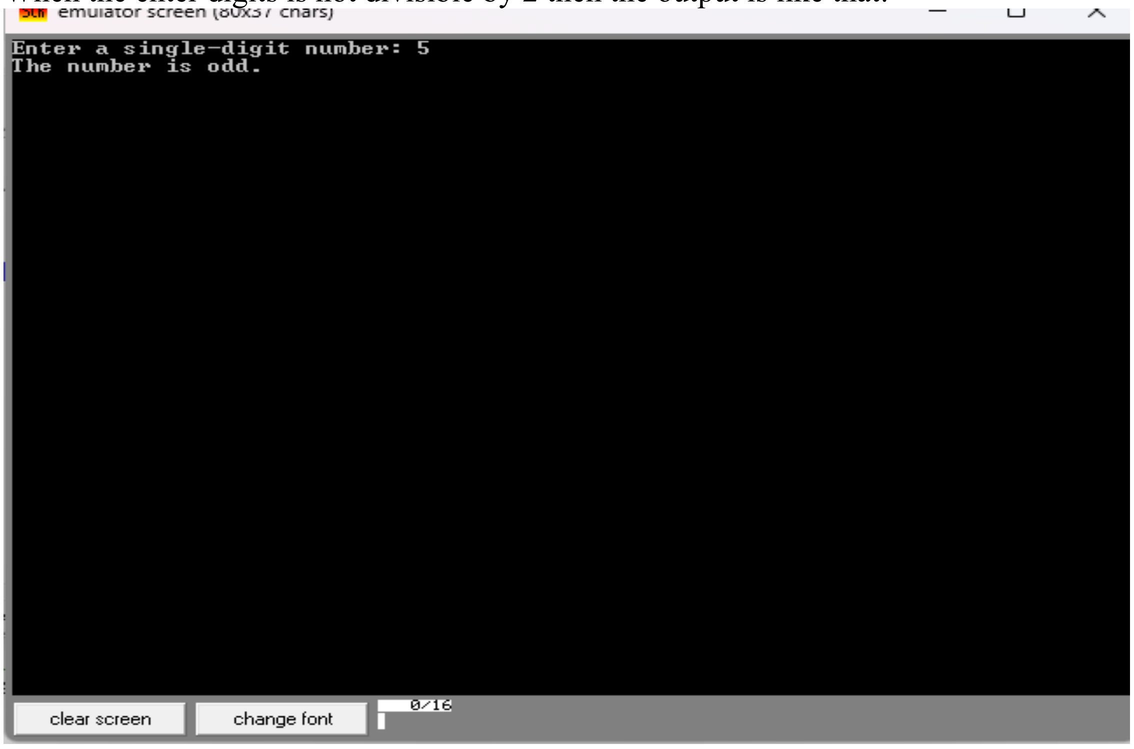
```
    ; Check if the number is even or odd
    mov bl, 2        ; Set divisor as 2 (for even/odd check)
    div bl           ; Divide the input number by 2, remainder in AH

    cmp ah, 0        ; Check remainder (AH)
    je IsEven        ; If remainder is 0, jump to IsEven

    ; If the number is odd
    mov dx, offset newline
    mov ah, 9
    int 21h
    mov dx, offset odd_msg
    mov ah, 9
    int 21h
    jmp EndProgram

IsEven:
    ; If the number is even
    mov dx, offset newline
    mov ah, 9
    int 21h
    mov dx, offset even_msg
    mov ah, 9
    int 21h
    jmp EndProgram

InvalidInput:
    ; Handle invalid input
    mov dx, offset newline
    mov ah, 9
    int 21h
    mov dx, offset prompt
    mov ah, 9
    int 21h

EndProgram:
    ; Exit the program
    mov ah, 4Ch      ; Function 4Ch: Exit to DOS
    int 21h

main endp
end main

      ; Message for sum result, with newline
```

Output
When the enter number is divisible by 2 then the output is like that



```
Enter a single-digit number: 6
The number is even.
```

When the enter digits is not divisible by 2 then the output is like that:



```
Enter a single-digit number: 5
The number is odd.
```