```
.model small
display macro msg
       lea dx, msg
       mov ah, 09h
       int 21h

endm


.data
msg1   db   0dh, 0ah, "Enter First String : $"
msg2   db   0dh, 0ah, "Enter Second String : $"
msg3   db   0dh, 0ah, "Length of First String : $"
msg4   db   0dh, 0ah, "Length of Second String: $"
msg5   db   0dh, 0ah, "Strings are EQUAL $"
msg6   db   0dh, 0ah, "Strings are NOT EQUAL $"
string1 db  80h dup(?)
string2 db  80h dup(?)


.code
start : mov ax, @data
        mov ds, ax
        display msg1
        mov si, offset string1
        call readstr
        mov bl, cl    ; store length of first string
        display msg2
        mov si, offset string2
        call readstr
        push bx
        push cx
```

```asm
        display  msg 3
        mov  al, bl
        call  len-dis
        display  msg 4
        mov  al, cl
        call  len-dis
        pop  cx
        pop  bx
        cmp  cl, bl   ;  compare the lengths
        jne  fail     ;  if lengths are equal,
                            process next statement
        mov  si, offset string1
        mov  di, offset string 2
        cld

chk:    mov  al, [si]
        cmp  al, [di]
        jne  fail
        inc  si
        inc  di
        dec  cl
        jnz  chk
        display  msg 5
        jmp  final

len-dis  proc near
        xor  ah, ah
        add  al, 00h
        aam
        add  ax, 3030h
        mov  bh, al
```

```asm
        mov   dl, ah
        mov   ah, 02 h
        int   21h
        mov   dl, bh
        mov   ah, 02 h
        int   21h
        ret
len_dis endp
readstr proc near
        xor   cl, cl

Back :  mov   ah, 01h
        int   21h
        cmp   al, dh
        je    finish
        mov   [si], al
        inc   si
        inc   cl
        jmp   back

finish : mov  [si], byte ptr '$'
         ret
readstr endp

fail : display msg 6
final : mov ah, 4ch
        int 21 h
endstart
```