```
.model small
.data
str1 db  10 dup(0)
str2 db  10 dup(0)
len1 db 00
len2 db  00
msg1 db 0dh,0ah,"enter first string $"
msg2 db 0dh,0ah,"enter second string $"
msg3 db 0dh,0ah, "strings are equal $"
msg4 db 0dh,0ah, "strings are not equal
msg5 db 0dh, 0ah,"length of first string is $"
msg6 db 0dh, 0ah,"length of second string is"
msg7 db 0dh,0dh,"length of string$"
.code
mov ax,@data
mov ds,ax
lea dx,msg1
mov ah,09h
int 21h
mov si,00
back1 : mov ah,01h
        int 21h
        cmp al,0dh
        je next1
        mov str1[si],al
```

```
inc si
inc len1
jmp back 1
next1 : lea dx, msg2
        mov ah, 09h
        int 21h
        mov si, 00
back2 : mov ah, 01h
        int 21h
        cmp al, 0dh
        je next2
        mov str2[si], al
next2 : mov al, len1
        cmpal , len2
        jne not equal
        mov si, 00
        mov di, 00
~~back3~~.  mov cl, len1
back3 : mov al, str1 [si]
        cmp al, str2[di]
        jne notequal
        inc si
        inc di
        dec cl
        jnz back3
```

```
lea dx, msg 3
mov ah, 09h
int 21h
lea dx, msg7
mov ah, 09h
int 21h
mov dl, len1
add dl, 30h
mov ah, 02h
int 21h
jmp last
not equal : lea dx, msg4
mov ah, 09h
int 21h
lea dx, msg5
mov ah, 09h
int 21h
mov dl, len1
add dl, 30h
mov ah, 02h
int 21h
lea dx, msg6
mov ah, 09h
int 21h
mov dl, len2
```

```
add dl, 30h
 mov ah, 02h
  int 21h
 last : mov ah, 4ch
             int 21h
 end
```