.model small

Display micro 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 2nd 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 string 1
call readstr
MOV BL, CL
display msg2
mov si, offset string 2
call readstr
Push BX
Push CX
display msg3
mov al, BL
call len_dis
display msg4
mov al, CL
call len_dis
pop CX
Pop BX
CMP CL, BL
JNE FAIL
MOV SI, OFFSET string 1
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

Hex_dis  Proc Near
         XOR AH,AH
         ADD AL, 00H
         AAM
         ADD AX, 3030H
         MOV BH, AL
         MOV DL, AH
         MOV AH, 02H
         INT 21h
         MOV DL,BH
         MOV AH, 02H
         INT 21H

RET
Hex_DISP ENDP
Readstr  Proc near
         XOR CL,CL
Back:    MOV AH, 01H
         int 21h
         CMP AL, 0DH
```

AAD

AL

BH = 5 | DL = 0

DL = 5

8 bit.

-128 to 128

ASCII of enter.

// Read from Key board and store ASCII in AL register.

// 0DH → to know end of string.

```
JE FINISH
mov [si], AL
inc si
inc dl
jmp back

finish: mov [si], byte ptr '$'   // Append $ symbol eog
        ret

Readstr endp

Fail:   display msg6

Final:  mov ah, 4ch
        int 21h

        END start.
```