S.R.PUJA
1BM19CS135

COMPARE 2 STRING

```
.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 THE LENGTH OF FIRST STRING
   DISPLAY MSG2
     MOV SI, OFFSET STRING2
     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
```

S.R.POOJA
1BM19CS135

```
CMP CL, BL          ; COMPARE THE LENGTHS
JNE FAIL            ; IF LENGTHS ARE EQUAL, PROCESS NEXT STATEMENT:
MOV SI, OFFSET STRING1
MOV DI, OFFSET STRING2
CLD
CHK : MOV AL, [SI]  ; COMPARE BOTH THE STRING
      CMP AL, [DI]
      JNE FAIL
      INC SI
      INC DI
      DEC CL
      JNZ CHK
      DISPLAY MSG5
      JMP FINAL
LEN_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
LEN_DIS ENDP
READSTR PRO NEAR
      XOR CL, CL
BACK: MOV AH, 01H
      INT 21H
      CMP AL, 0DH
      JE FINISH
```

S.R.POOJA
1BM19CS135

```
        MOV [SI], AL
        INC SI
        INC CL
        JMP BACK
FINISH: MOV [SI], BYTE PTR '$'
        RET
READSTR ENDP
FAIL:   DISPLAY MSG6
FINAL:  MOV AH, 4CH
        INT 21H
END START.
```

File Edit Search View Options Help

C:\MASM\STRING.ASM

```
.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
```

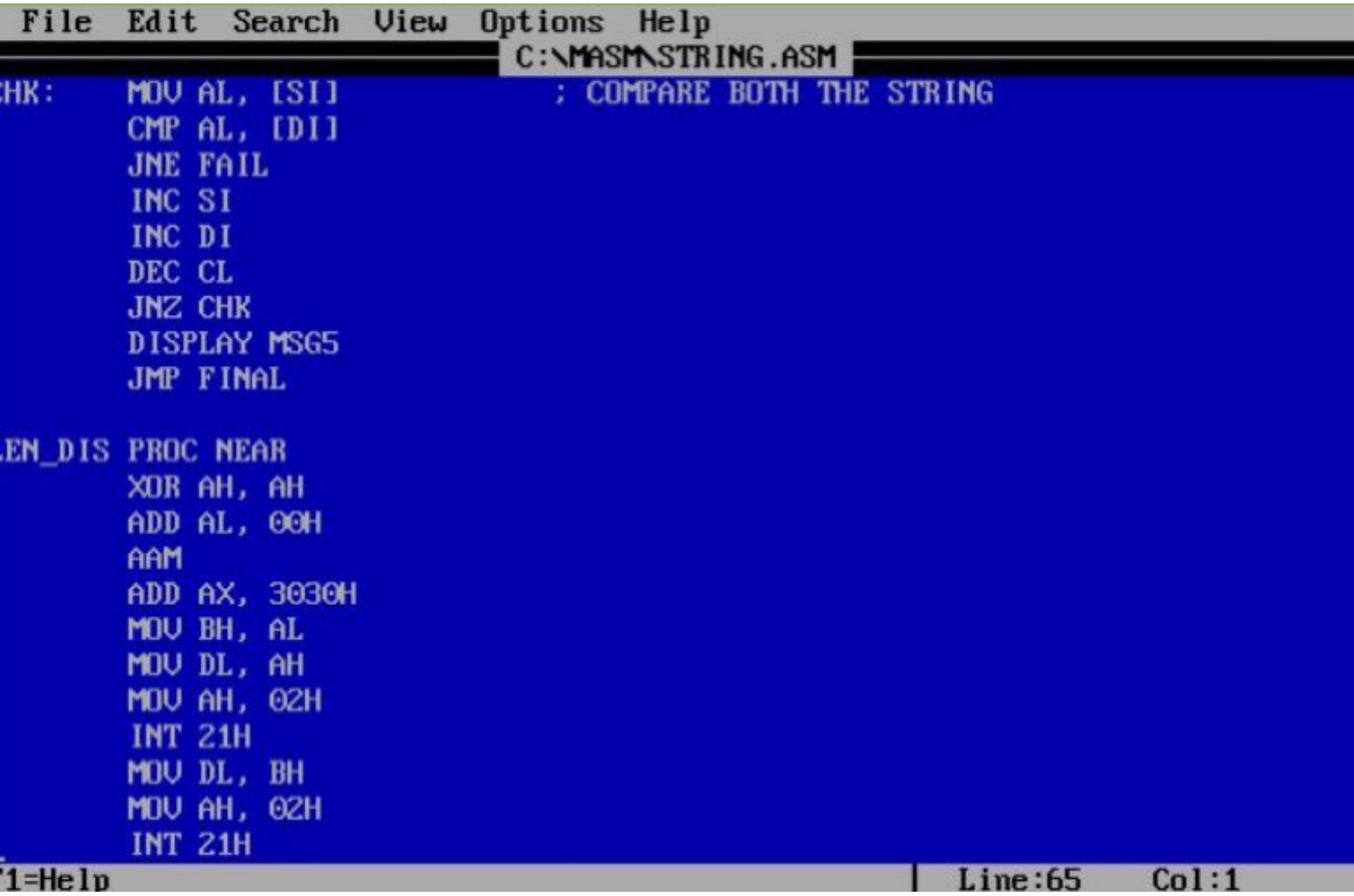
F1=Help | Line:1 Col:1

File Edit Search View Options Help

C:\MASM\STRING.ASM

```
        MOV SI, OFFSET STRING1
        CALL READSTR
        MOV BL, CL              ; STORE THE LENGTH OF FIRST STRING
        DISPLAY MSG2
        MOV SI, OFFSET STRING2
        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              ; COMPARE THE LENGTHS
        JNE FAIL                ; IF LENGTHS ARE EQUAL, PROCESS NEXT STATMENT
        MOV SI, OFFSET STRING1
        MOV DI, OFFSET STRING2
        CLD
CHK:    MOV AL, [SI]            ; COMPARE BOTH THE STRING
```

F1=Help | Line:44 Col:1



```
        INT 21H
RET
LEN_DIS ENDP
READSTR PROC NEAR
        XOR CL, CL
BACK:   MOV AH, 01H
        INT 21H
        CMP AL, 0DH
        JE FINISH
        MOV [SI], AL
        INC SI
        INC CL
        JMP BACK
FINISH: MOV [SI], BYTE PTR '$'
        RET
READSTR ENDP
FAIL:   DISPLAY MSG6
FINAL:  MOV AH, 4CH
        INT 21H
END START
```

F1=Help | Line:85 Col:1