* COMPARE STRINGS :]

```
.model small

DISPLAY  MACRO MSG
         LEA DX, MSG
         MOV AH, 09H
         INT 21H

ENDM

.DATA
MSG1  DB  0DH, 0AH, " Enter first sting ..$"
MSG2  DB  0DH, 0AH, " Enter second String $"
MSG3  DB  0DH, 0AH, " Length of string 1 $"
MSG4  DB  0DH, 0AH, " Length of string 2 $"
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
         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
         JNE  FAIL
        MOV  SI,  OFFSET  STRING1
        MOV      DI,  OFFSET  STRING2
         CLD

CHK :      MOV   AL, [SI]
          ~~MOV~~
           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    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
        MOV AH, 4CH
        INT 21H

END START
```

* DISPLAY SYSTEM TIME :-

```
.model small

DISPLAY  MACRO MSG
         Lea DX, MSG
         MOV AH, 09H
         INT 21H

ENDM

.DATA

MSG1 DB 0DH, 0AH, " The Time is $ "

.code
                                        CH -> hour
                                        CL -> min
     MOV AX, @DATA                      DH -> sec
     MOV DS, AX

     MOV AH, 02CH               / reads time from
     INT 21H                      system
```

```
MOV AL, CH
 AAM
MOV BX, AX
 CALL DISP
 MOV DL, ':'          (or) MOV DL, 20H
                                  ascii
 MOV AH, 02H   ;    display
  INT 21H            character one by one.
                           [DL register on

MOV AL, CL
 AAM
 MOV BX, AX
 CALL DISP
  MOV DL, ':'
 MOV AH, 02H
  INT 21H

MOV AL, DH
 AAM
 MOV BX, AX
 CALL DISP
  MOV AH, 02H
   INT 21H

DISP PROC NEAR
 MOV DL, BH
 ADD DL, 30H
 MOV AH, 02H
 INT 21H
                          MOV DL, BL
                          ADD DL, 30H
                           MOV AH, 02H
                            INT 21H
                           RET
                          DISP ENDP
                           END
```