// PALINDROME

```
        . MODEL   SMALL
        DISPLAY  MACRO MSG
                 LEA  DX, MSG
                 MOV  AH, 09 H
                 INT  21 H
        ENDM
        . Data
        MSG1   DB  ODH, OAH, " Enter String : $"
        MSG2   DB  ODH, OAH, "Reverse String : $"
        MSG3   DB  ODH, OAH, " ilp String is Palindrome. $"
        MSG4   DB  ODH, OAH, "ilp String is not  a  palindrome
                            String. $"
        String DB 80H DUP (?)
        Rstring DB  80 H DUP (?)
        . Code
        START: MOV AX , @ Data
               MOV  DS, AX
               DISPLAY  MSG1
               ; Take  the  String  keyboar  ch by ch
               XOR  CL, CL
        Again : MOV  AH, 01 H
               INT  21 H
               CMP AL, ODH
               JE NEXT
               MOV  [SI], AL
               INC  SI
               INC  CL
               JMP  AGAIN
        NEXT : MOV [SI], BYTE PTR '$'
               ; String  input  over ....
               DEC  SI
               MOV  CX, CL
               ; Reverse  the  String & store in RString
```

```
          MOV  DI, OFFSET  RSTRING
BACK:  MOV  AL,  [SI]          ; BMSLE
          MOV  [DI], AL
          DEC  SI
          INC  DI
          DEC  CH
          JNZ  BAC
          MOV [DI], BYTE  PTR '$'
          DISPLAY  MSG2
          DISPLAY  RSTRING
          MOV  SI, OFFSET  STRING
          MOV  DI, OFFSET  RSTRING
AG:   MOV  AL, [SI]  ; BMSE  madam
          MOV  AL, [DI]  ; ESSMB  madam
          JNE  FAIL
          INC  SI
          INC  DI
          DEC  CX
          JZ  SUCESS
          JMP  AG
FAIL  :  DISPLAY  MSG4
            JMP  FINAL
SUCESS :  DISPLAY  MSG3
FINAL  :  MOV  AX, 4CH
            INT  21H
END
```

‖ ASCII

```
. MODEL small
. data
msg1 df 0dh, 0ah, " Enter alphanumeric char$"
res db 02 dup(0)
. Code
mov ax, @ data
mov ds, ax
lea dx, msg1
call disp
mov ah, 0lh
int 21h
mov bl, al
mov cl, 4
shr al, cl
cmp al, 0ah
JC digit
ADD AL, 07H
digit: add al, 30h
       mov res, al
       and bl, 0Fh
       cmp bl, 0ah
       JC digit1
       add bl, 07h
digit1: add bl, 30h
        mov res+1, bl

       mov ah, 00h
       mov al, 03h      ; TEXT MODE
       int 10h

       mov ah, 02h   ; set the cursor POS
       mov bh, 00h   ; Page no
       mov dh, 0ch   ; Row coo is TOP
       mov dl, 28h   ; Column val
       int 10h
```

```
mov  res+2, '$'
lea  dx, res
call disp
mov  ah, 4ch
int  21h

disp  proc  near
mov   ah, 09h
int   21h
ret
disp  endp
end.
```