SONA ROOPA Page No.
Date

```
.model small
.data
msg1 db 0dh, 0ah," Enter alphanumeric character $"
res db 02 dup(0)
.code
mov ax,@data
mov ds, ax
lea dx, msg1
call disp
mov ah, 01h
int 21h
mov bl, al.
mov cl, 4
mov 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
        int 20h

        mov ah, 02h
   mov  bh, 00h
mov dh, 0ch
mov dl. 28 h
int 10h

mov res+2, '$'
lea dx, res
call disp
mov ah, 4ch
int 21 h.
~~end~~.
disp proc near.
mov ah, 09 h.
int 21h.
ret
disp endp.
end.
```

```
display  macro  msg
    LEA   Dx , msG
    mov  AH, 09H
    int  21H
.data
msg1  db  0dh , 0ah, "Enter String : $"
msg2  db  0db, 0ab , " Reserve string: $ "
msg 3  db  0 db, 0ah," Input string  is polindrome. $
msg4  ds  0dh, 0ah, "Input string is not a polindrome
                        string . $".
string  db 80 h dup(?)
Rstring  db 80h dup(? ),
.code,
start  :  mov  an , @data.
          mov  ds , an
          Display  msg1
       ; take the string from keyboard character by character
       mov  SI, OFFSET STRING
       XOR cL, CL
AGAIN :  MOV  ah, 01 H.
         Int  21h
        cmp  AL, 0dh.
        JE   next
        mov  [SI], Al
        inc  si
        inc  cl
        jmp  again
NEXT : mov  [si], BYTEPTR '$'
        , string input over.
       Dec  si
      mov ch, cl
       : Reverse the string and store in Rstring
       mov  DI, OFFSET RSTRING
Back  : mov  al, [SI]
      mov  [DI] , Al
       Dec  SI
```

```
        inc DI
        DEC CH
        JNZ BACK
        mov [DI], byte ptr '$'.
        Display msg2.
        Display Rstring
        mov SI, OFFSET STRING
        mov DI, OFFSET STRING.
AG:     mov AL, [SI]
        cmp AL, [DI]
        JNE FAIL
        INC SI
        INC DI
        DEC CX
        JZ SUCCESS
        Jmp AG
FAIL : Display MSG4
        JMP FINAL
SUCCESS: DISPLAY MSG3
FINA : mov ah, 4ch
end     ~~end~~ .
```