LAB 6.

Date / /

```
3.  .model small
    .data
    msg1 db 0dh, 0ah, "Enter alphanum
                       character $"

    res db 02 dup(0)
    .code
     mov  ax, @data
     MOV  ds, ax
     lea  dx, msg1
     Call disp
     mov  ah, 01h          //input in AL regist
     int  21h
     mov  bl, al
     mov  cl, 4
    shr al, cl             //only three will
    cmp al, 0ah
    Jc  digit
    ~~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

              moh  ah, 00h
              mov  al, 03h        ;Text mod
              int  ~~21h~~ 10h
```

Page No.

```
mov   ah, 02h      ; Set cursor position
mov   bh, 00h      ; Page number
mov   dh, 0ch      ; Row (00 is Top).
mov   dl , 28h     ;, Colum Val
int   10h          ; Bios Interrupt
mov   res+2, '$'   ; to show the ~~co~~ of array
lea   dx, res
call  disp
mov   ah, 4ch
int   21h

disp proc near
 mov ah, 09h   } Display the string.
 int 21h
 ret
 disp endp
 end
```

Enter the alphanumeric character 1

34

Palindrome

Date ___/___/___

```
4o    .model Small
Display Macro MSG
          lea  DX ,MSG
          MOV  Ah, 09h
          int  21h

End m


. data
MSG1   db   odh, oah,    "Enter String:
MSG2   db   odh ,oah,     " Reverse Strin
MSG3   db   odh, oah,     " Input String
                                    Palindr

MSG4   db   odh, oah      "Input string i
                                not palin


string   db   80h   Dup(?)
rstring  db   80h   DUp(?)
.Code
start :    MOV   AX, @data
           MOV   DS,   AX
           Display  MSG1
           MOV   SI , offset string
           Xor   cl, cl


Again :    MOV Ah, 01h
           int 21h
           cmp  al, odh
           JE   Next
           MOY  [SI] ,AL
```

```
        INC   SI
        INC   CI
        Jmp   Again

Next:       MOV [SI], Byte Ptr   '$'
                ; 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 Rstring
```

Date ___/___/___

```
AG:     MOV   AL,  (SI]
        cmp   Al,  (DI]
        JNE   fail
        INC   SI
        INC   DI
        Dec   CX
        JZ    Success
        Jmp   ag

Fail:       Display MSG4
            Jmp  final

Sucass:     Display  MSG3
final:      MOV ah, 4Ch
             int 21h

end
```