

MP - Lab1. Palindrome program

```
.model small
```

```
display macro msg
```

```
    lea dx, msg
```

```
    mov ah, 09h
```

```
    int 21h
```

```
endm
```

```
.data
```

```
msg1 db 0dh, 0ah, "enter String: $"
```

```
msg2 db 0dh, 0ah, "reverse string: $"
```

```
msg3 db 0dh, 0ah, "input String is a  
palindrome string. $"
```

```
msg4 db 0dh, 0ah, "input string is not a  
palindrome string. $"
```

```
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, 0dh
```

```
    je next
```

```
    mov [si], al
```

```
    inc si
```

```
    int cl
```

```

        jmp again
next:   mov [si], byte ptr '$'
        ; String input over .....
        dec si
        mov ch, cl
        ; reverse the string & store in rstring
        mov di, offset rstring
back:   mov al, [si]
        mov [di], al
        dec si
        inc di
        dec ch
        jnz back
        mov [di], byte ptr '$'
        display msg 2
        display rstring
        mov si, offset string
        mov di, offset rstring
ag:     mov al, [si]
        cmp al, [di]
        jne fail
        inc si
        inc di
        dec cx
        jz Success
        jmp ag
fail:   display msg 4
        jmp final
Success: display msg 3
final:  mov ah, 4Ch
        int 21h
        end Start

```

2) ASCII

.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 ; : 3a = 3341

shr al, cl ; al = 3a al = 03 bl = 3a

cmp al, 0ah ; bl = 3a 30 0f

jc digit ; 33 0a

add al, 07h ; 41

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 position  
mov bh, 00h ; page No  
mov dh, 0ch ; row (00 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
```