**PROGRAM**

print macro m

mov ah,09h

mov dx,offset m

int 21h

endm

.model small

.data

menu db 10,13, "Menu: "

db 10,13, "1. Enter the string to find length "

db 10,13, "2. Check for palindrome "

db 10,13, "3. Display Entered String"

db 10,13, "4. Exit "

db 10,13, " "

db 10,13, "Your choice:$"

msg1 db 10,13, "Your choice is:$"

mc1 db 10,13, "case 1 $"

mc2 db 10,13, "case 2 $"

mc3 db 10,13, "case 3 $"

mc4 db 10,13, "Exiting the program $"

mc6 db 10,13, "Invalid choice$"

mc7 db 10,13, "Exiting the program $"

empty db 10,13, " $"

str1 db 25,?,25 dup('$')

str2 db 25,?,25 dup('$')

len db ?

str3 db 0ah,0dh,"Enter name: $"

mstring db 10,13, "Enter the string: $"

notpalin db 10,13, "String is not a palindrome. $"

palin db 10,13, "String is a palindrome. $"

mstring2 db 10,13, "Enter second string: $"

mlength db 10,13, "Length is: $"

scount db ?

.code

start:

mov ax,@data

mov ds,ax

again:

print menu

call accept

mov bl,al

case1:

cmp bl,"1"

jne case2

print mc1

print empty

print mstring

call accept\_string

mov cl,str1+1

mov bl,cl

print mlength

call display1

print empty

jmp again

case2:

cmp bl,"2"

jne case3

print mc2

print empty

print mstring

call accept\_string

mov si,offset str1

mov cl,str1+1 ;store the length

mov ch,00h

mov len,cl

inc si

add si,cx

mov di,offset str1

add di,0002h

mov cl,len

cmpagain:

mov al,[si]

mov ah,[di]

inc di

dec si

cmp al,ah

jne nopalin

dec cl

jnz cmpagain

print palin

print empty

jmp again

nopalin:

print notpalin

print empty

jmp again

case3:

cmp bl,"3" ;check for case 3

jne case4

print mc3

lea dx,str1+2

mov ah,09h

int 21h

print empty

jmp again

case4: cmp bl,"4"

jne case5

print mc4

jmp exit

case5:

print mc7

exit:

mov ah,4ch

int 21h

accept proc near

mov ah,01

int 21h

ret

accept endp

display1 proc near

mov al,bl

mov bl,al

and al,0f0h

mov cl,04

rol al,cl

cmp al,09

jbe number

add al,07

number:

add al,30h

mov dl,al

mov ah,02

int 21h

mov al,bl

and al,00fh

cmp al,09

jbe number2

add al,07

number2:

add al,30h

mov dl,al

mov ah,02

int 21h

ret

display1 endp

accept\_string proc near

mov ah,0ah ;accept string from user function

mov dx,offset str1 ; store the string in memory pointed by "DX"

int 21h

ret

accept\_string endp

print\_string proc near

mov ah,09h

lea dx,[mstring]

int 21h

ret

print\_string endp

print\_string\_ds\_dx proc near

mov ah, 09h

int 21h

ret

print\_string\_ds\_dx endp

end start

end

**OUTPUT**





