**Name: Adwait S Purao**

**UID: 2021300101**

**Batch: B2**

**Aim:**

**1) To check whether string is a Palindrome or not**

**2) To reverse the string**

**3) To find the length of the string**

1. **Check whether string is a Palindrome or not**

**CASE 1: String is not a Palindrome**

**Code:**

Data Segment

str1 db 'Adwait','$'

strlen1 dw $-str1

strrev db 20 dup(' ')

strpali db "It is a palindrome","$"

strnotpali db "It is not a palindrome","$"

Data Ends

Code Segment

Assume cs:code, ds:data

start:

mov ax, data

mov ds, ax

mov es, ax

mov cx, strlen1

add cx, -2

lea si, str1

lea di, strrev

add si, strlen1

add si, -2

L1:

mov al, [si]

mov [di], al

dec si

inc di

loop L1

mov al, [si]

mov [di], al

inc di

mov dl, '$'

mov [di], dl

mov ax,data

mov ds,ax

lea si,str1

lea di,strrev

cld

mov cx,07h

repe cmpsb

jnz noteq

lea dx,strpali

mov ah,09h

int 21h

jmp endp1

noteq:

lea dx, strnotpali

mov ah,09h

int 21h

endp1:

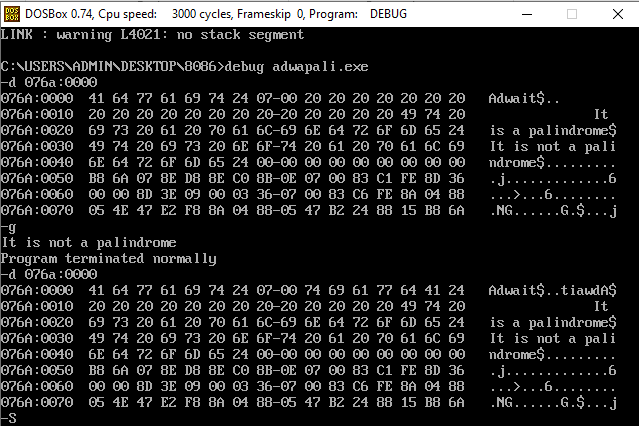
mov ax, 4c00h

int 21h

Code Ends

End start

**Output:**



**Case 2: String is a Palindrome**

**Code:**

Data Segment

str1 db 'racecar','$'

strlen1 dw $-str1

strrev db 20 dup(' ')

strpali db "It is a palindrome","$"

strnotpali db "It is not a palindrome","$"

Data Ends

Code Segment

Assume cs:code, ds:data

start:

mov ax, data

mov ds, ax

mov es, ax

mov cx, strlen1

add cx, -2

lea si, str1

lea di, strrev

add si, strlen1

add si, -2

L1:

mov al, [si]

mov [di], al

dec si

inc di

loop L1

mov al, [si]

mov [di], al

inc di

mov dl, '$'

mov [di], dl

mov ax,data

mov ds,ax

lea si,str1

lea di,strrev

cld

mov cx,07h

repe cmpsb

jnz noteq

lea dx,strpali

mov ah,09h

int 21h

jmp endp1

noteq:

lea dx, strnotpali

mov ah,09h

int 21h

endp1:

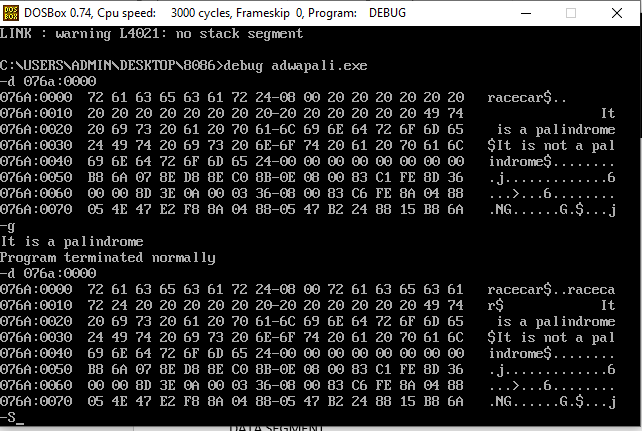
mov ax, 4c00h

int 21h

Code Ends

End start

**Output:**



1. **Reverse the given string**

**Code:**

DATA SEGMENT

STR1 db "VIRINCHI","$"

STRLEN1 dw $-STR1

STRREV db 20 DUP(' ')

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

START:

MOV AX, DATA

MOV DS, AX

MOV ES, AX

MOV CX, STRLEN1

ADD CX, -2

LEA SI, STR1

LEA DI, STRREV

ADD SI, STRLEN1

ADD SI, -2

L1:

MOV AL, [SI]

MOV [DI], AL

DEC SI

INC DI

LOOP L1

MOV AL, [SI]

MOV [DI], AL

INC DI

MOV DL, "$"

MOV [DI], DL

PRINT:

MOV AH, 09H

LEA DX, STRREV

INT 21H

EXIT:

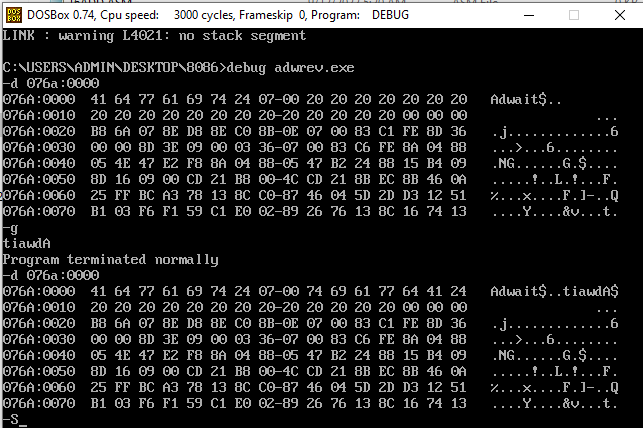
MOV AX, 4C00H

INT 21H

CODE ENDS

END START

**Output:**



**3) Find the length of the string**

**Code:**

data segment

Length1 db 01 dup(?)

String1 db "Richard","$"

data ends

code segment

assume cs:code, ds:data

start:

mov ax,data

mov ds,ax

lea si,String1

mov cl,00

AGAIN:

mov al,[si]

cmp al,"$"

jz dn

inc si

inc Length1

jnz AGAIN

dn:

mov al,Length1

add al,48

mov ah,2

int 21h

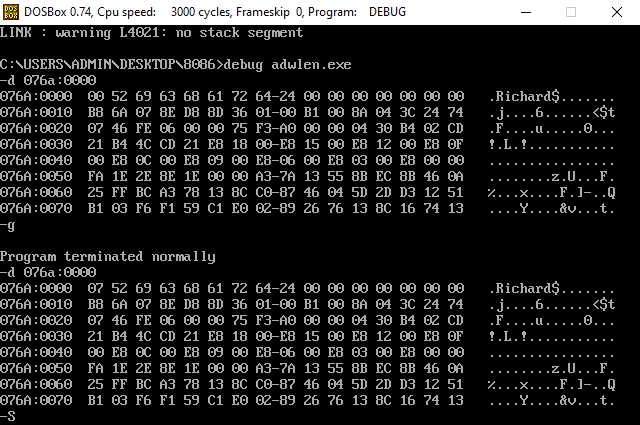
mov ah,4ch

int 21h

code ends

end start

**Output:**



**Conclusion:**

**In the following program we learned implement a code in 8086 to check whether a string is a Palindrome or not, Find the length of the string , Reverse the string.**