Name:	Rithesh Shetty
Roll No:	54
Class/Sem:	SE/IV
Experiment No.:	7
Title:	Program to find whether given string is palindrome or not
Date of Performance:	05ss/03/2024
Date of Submission:	22/03/2024
Marks:	
Sign of Faculty:	

Aim: Assembly Language Program to find given string is Palindrome or not.

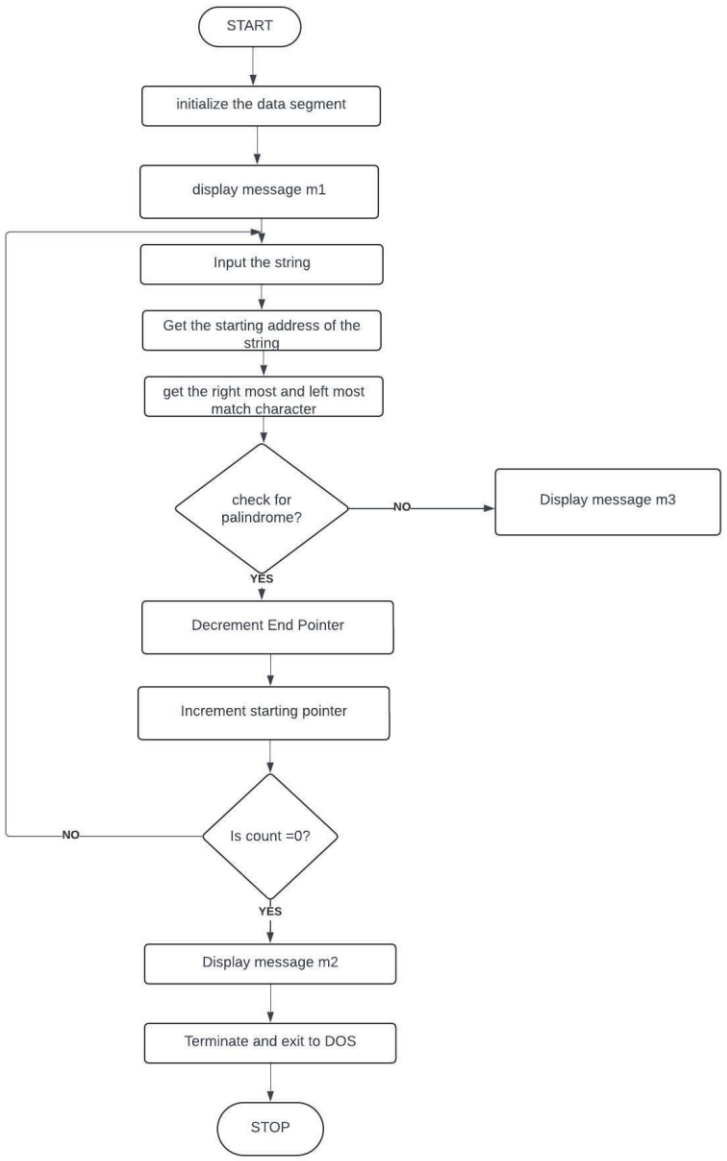
Theory:

A palindrome string is a string when read in a forward or backward direction remains the same. One of the approach to check this is iterate through the string till middle of the string and compare the character from back and forth.

Algorithm:

1. Initialize the data segment.
2. Display the message M1
3. Input the string
4. Get the string address of the string
5. Get the right most character
6. Get the left most character
7. Check for palindrome.
8. If not Goto step 14
9. Decrement the end pointer
10. Increment the starting pointer.
11. Decrement the counter
12. If count not equal to zero go to step 5
13. Display the message m2
14. Display the message m3
15. To terminate the program using DOS interrupt
    a. Initialize AH with 4ch
    b. Call interrupt INT 21h
16. Stop

Flowchart:



code:

```
org 100h

.data
buff db 80
m1 db 10,13, 'Enter a string:$'
m2 db 10,13, 'String entered is a palindrome:$'
m3 db 10,13, 'string entered is not palindrome:$'

.code

lea dx,m1
mov ah,09h
int 21h

lea dx,buff
mov ah,0ah
int 21h

lea bx,buff+1
mov si ,01h

mov ch,00h
mov cl,[buff+1]

sar cl, 1

PAL:
mov ah,[buff+si]
mov al,[buff+di]
cmp al,ah
jc L1
inc si
dec di

LOOP PAL

lea dx, m2
mov ah,09h
int 21h

jmp L2

L1:
lea dx, m3
mov ah, 09h
int 21h

L2:
mov ah,4ch
int 21h

ret
```

OUTPUT:

Conclusion:

- The assembly program checks whether a given input string is a palindrome or not. It prompts the user to enter a string, reads the input, and then compares the characters from both ends towards the middle. If the string is a palindrome, it displays a message confirming it; otherwise, it shows a message indicating that it's not. Finally, it terminates after displaying the result.

1. Explain SAR INSTRUCTION
    - SAR (Arithmetic Shift Right) in x86 assembly shifts each bit of a signed binary number to the right by a specified count while preserving the sign bit. It's used with syntax `SAR destination, count`, where `destination` is the operand to be shifted, and `count` is the shift amount. SAR effectively divides the operand by 2, rounding towards negative infinity for negative numbers.

2. Explain DAA instruction.
    - The DAA (Decimal Adjust for Addition) instruction in x86 assembly language is used to adjust the result of an addition operation performed on binary-coded decimal (BCD) numbers. It adjusts the AL register to ensure that after addition, it represents a valid two-digit packed decimal number. If the lower four bits of AL contain a value greater than 9 or the Auxiliary Carry flag (AF) is set, DAA adds 6 to AL. If the upper four bits of AL are greater than 9 or the Carry flag (CF) is set, DAA adds 60h to AL. This instruction is typically used in conjunction with addition operations on BCD numbers to ensure correct representation in the BCD format.

Vidyavardhini's College of Engineering & Technology