AY: 2024-25

Class:	SE	Semester:	IV
Course Code:	CSL404	Course Name:	Microprocessor Lab

Name of Student:	Shravani Sandeep Raut
Roll No. :	48
Experiment No.:	6
Title of the Experiment:	Program to reverse word in a string
Date of Performance:	20/02/2025
Date of Submission:	06/03/2025

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Performance	5	
Understanding	5	
Journal work and timely submission	10	
Total	20	

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Performance	4-5	2-3	1
Understanding	4-5	2-3	1
Journal work and timely submission	8-10	5-8	1-4

Checked by

Name of Faculty : Ms. Sweety Patil

Signature :

Date:

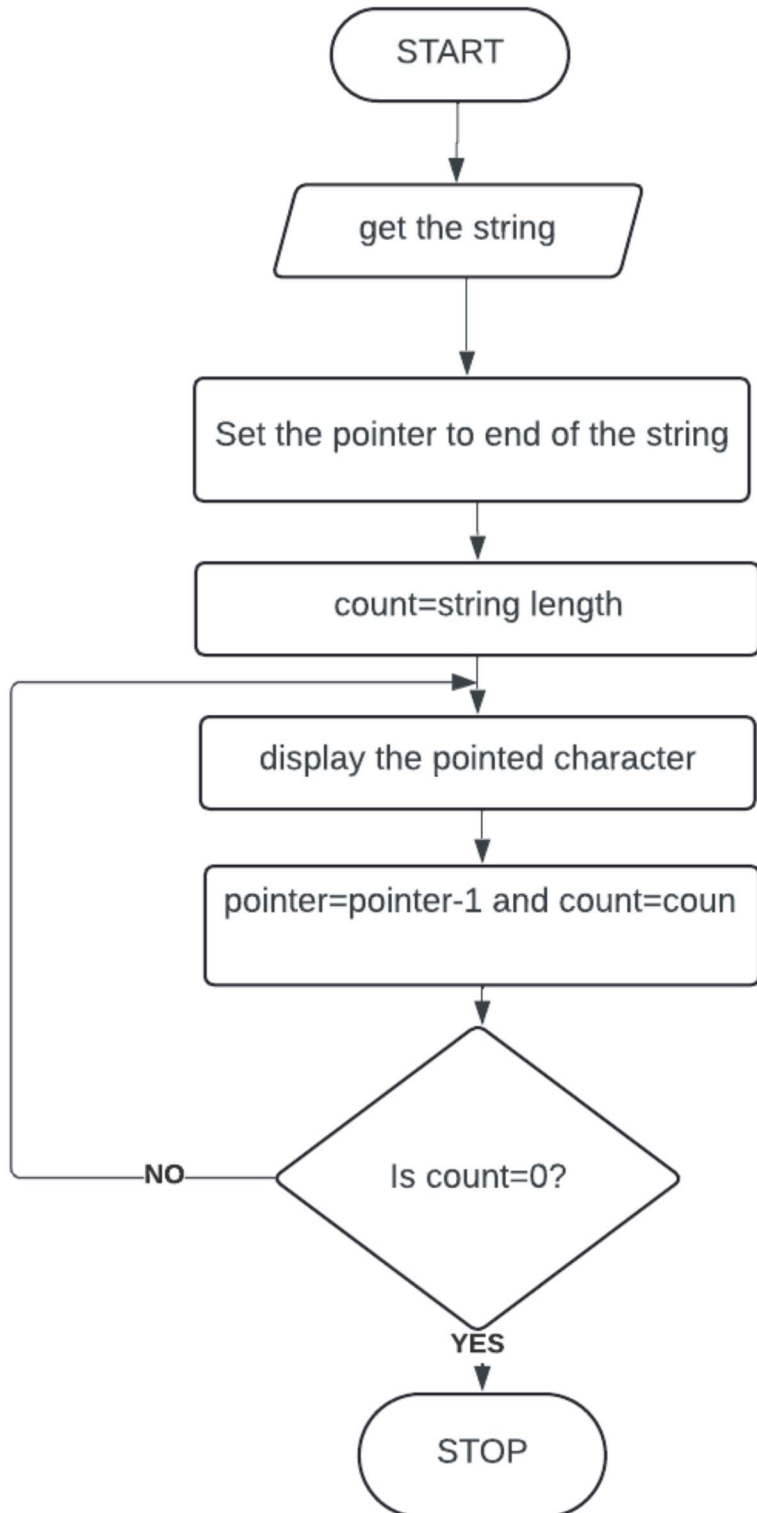**Aim:** Assembly Language Program to reverse the word in string.

Theory:

This program will read the string entered by the user and then reverse it. Reverse a string is the technique that reverses or changes the order of a given string so that the last character of the string becomes the first character of the string and so on.

Algorithm:

1. Start

2. Initialize the data segment

3. Display the message -1

4. Input the string

5. Display the message 2

6. Take characters count in DI

7. Point to the end character and read it

8. Display the character

9. Decrement the count

10. Repeat until the count is zero

11. To terminate the program using DOS interrupt

    a. Initialize AH with 4ch

    b. Call interrupt INT 21h

12. Stop

Flowchart:

```
                    ┌─────────────┐
                    │    START    │
                    └──────┬──────┘
                           │
                           ▼
                    ╱─────────────╲
                   ╱ get the string ╲
                   ╲                ╱
                    ╲──────┬───────╱
                           │
                           ▼
              ┌─────────────────────────────┐
              │ Set the pointer to end of the string │
              └──────────────┬──────────────┘
                             │
                             ▼
              ┌─────────────────────────────┐
       ┌─────▶│      count=string length     │
       │      └──────────────┬──────────────┘
       │                     │
       │                     ▼
       │      ┌─────────────────────────────┐
       │      │  display the pointed character │
       │      └──────────────┬──────────────┘
       │                     │
       │                     ▼
       │      ┌─────────────────────────────┐
       │      │ pointer=pointer-1 and count=coun │
       │      └──────────────┬──────────────┘
       │                     │
       │                     ▼
       │                  ◇─────────◇
       │                 ╱           ╲
       │   NO           ╱  Is count=0? ╲
       └───────────────◇               ◇
                        ╲             ╱
                         ╲           ╱
                          ◇─────────◇
                               │
                              YES
                               │
                               ▼
                        ┌─────────────┐
                        │    STOP     │
                        └─────────────┘
```

Code:

```
org 100h

.data
M1 db 10,13, 'Enter String:$'
M2 db 10,13, 'Reverse String is: $'
buff db 80

.code
lea dx, M1
mov ah, 09h
int 21h

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

lea dx, M2
mov ah, 09h
int 21h

mov ch,00h
mov cl, [buff+1]
lea bx, buff+1 ; Address of the first character
add bx, cx      ; Move to the end of the string

; Reverse
L1:
mov dl, [bx]    ; Load the current character into DL
mov ah, 02h
int 21h
dec bx
loop L1
```
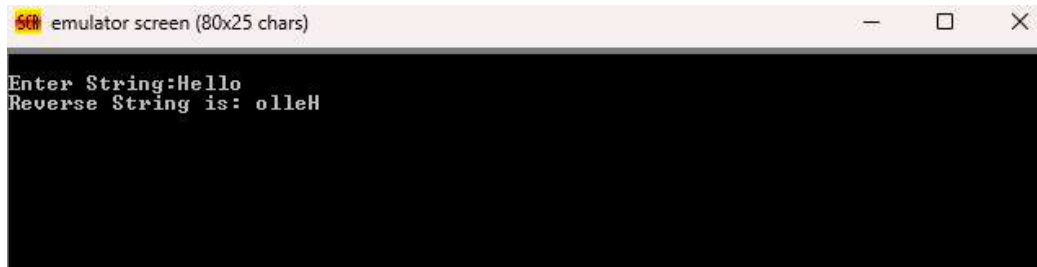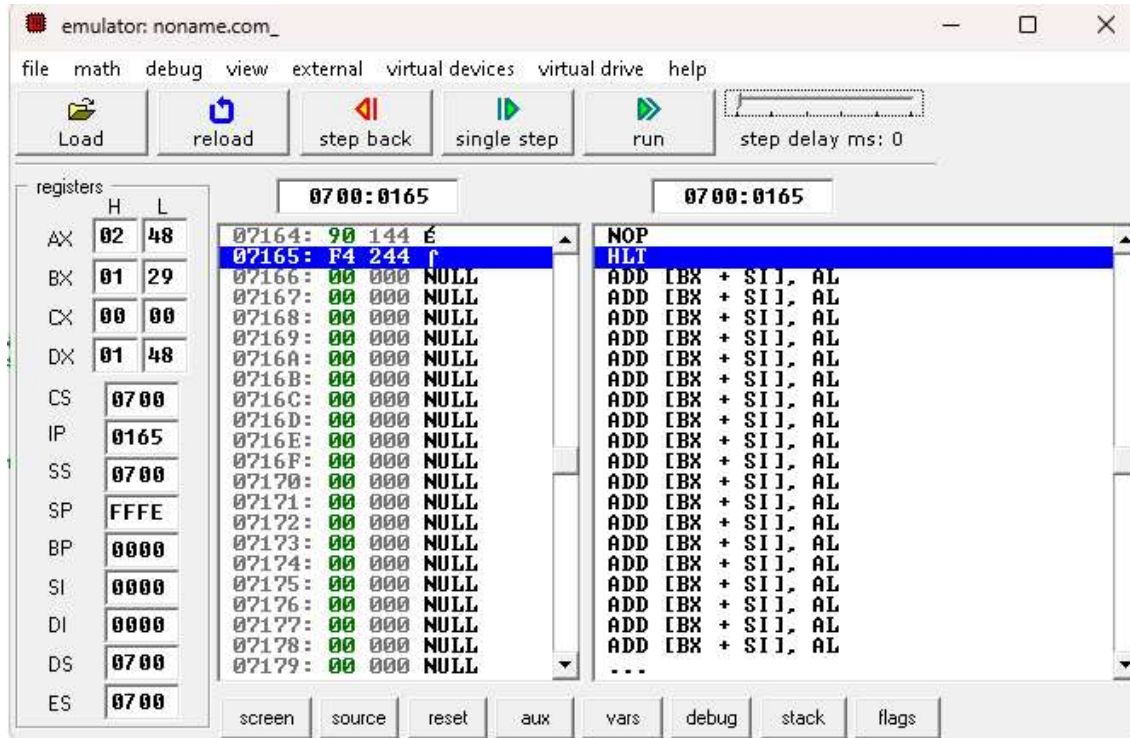
Output:

Conclusion: In conclusion, the assembly program for reversing a word in a string demonstrates the use of string manipulation and pointer arithmetic. By using registers to navigate through the string and swapping characters, the program effectively reverses the word. This highlights the power of assembly in directly managing memory and performing low-level operations efficiently.

1. Explain the difference between XLAT and XLATB

    The XLAT and XLATB instructions are used for table lookups in assembly language, but they have some differences:

    XLAT: Translates a byte in AL using a table located at the memory address in BX. The lookup is based on the value in AL, and the result is stored back in AL.

    XLATB: Works similarly but directly uses the BX register as the base address for the lookup and accesses a table in a different format, often in newer x86 processors.

    In most cases, XLATB is used as a more updated version of XLAT.

2. Explain the instruction LAHF

    The LAHF instruction is used to load the lower byte of the Flags register (the status flags) into the AH register. This includes flags like Zero (ZF), Sign (SF), Auxiliary Carry (AF), Parity (PF), and others. It's useful for saving the status of the flags to be later restored or used in decision-making within the program.