



## Vidyavardhini's College of Engineering &amp; Technology

Department of Artificial Intelligence and Data Science (AI&amp;DS)

|                             |  |
|-----------------------------|--|
| <b>Name:</b>                | BARI ANKIT VINOD                                 |
| <b>Roll No:</b>             | 65   |
| <b>Class/Sem:</b>           | SE/IV  |
| <b>Experiment No.:</b>      | 6  |
| <b>Title:</b>               | To perform program to reverse the word in string |
| <b>Date of Performance:</b> | 06/03/24   |
| <b>Date of Submission:</b>  | 06/03/24   |
| <b>Marks:</b>               |  |
| <b>Sign of Faculty:</b>     |  |



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

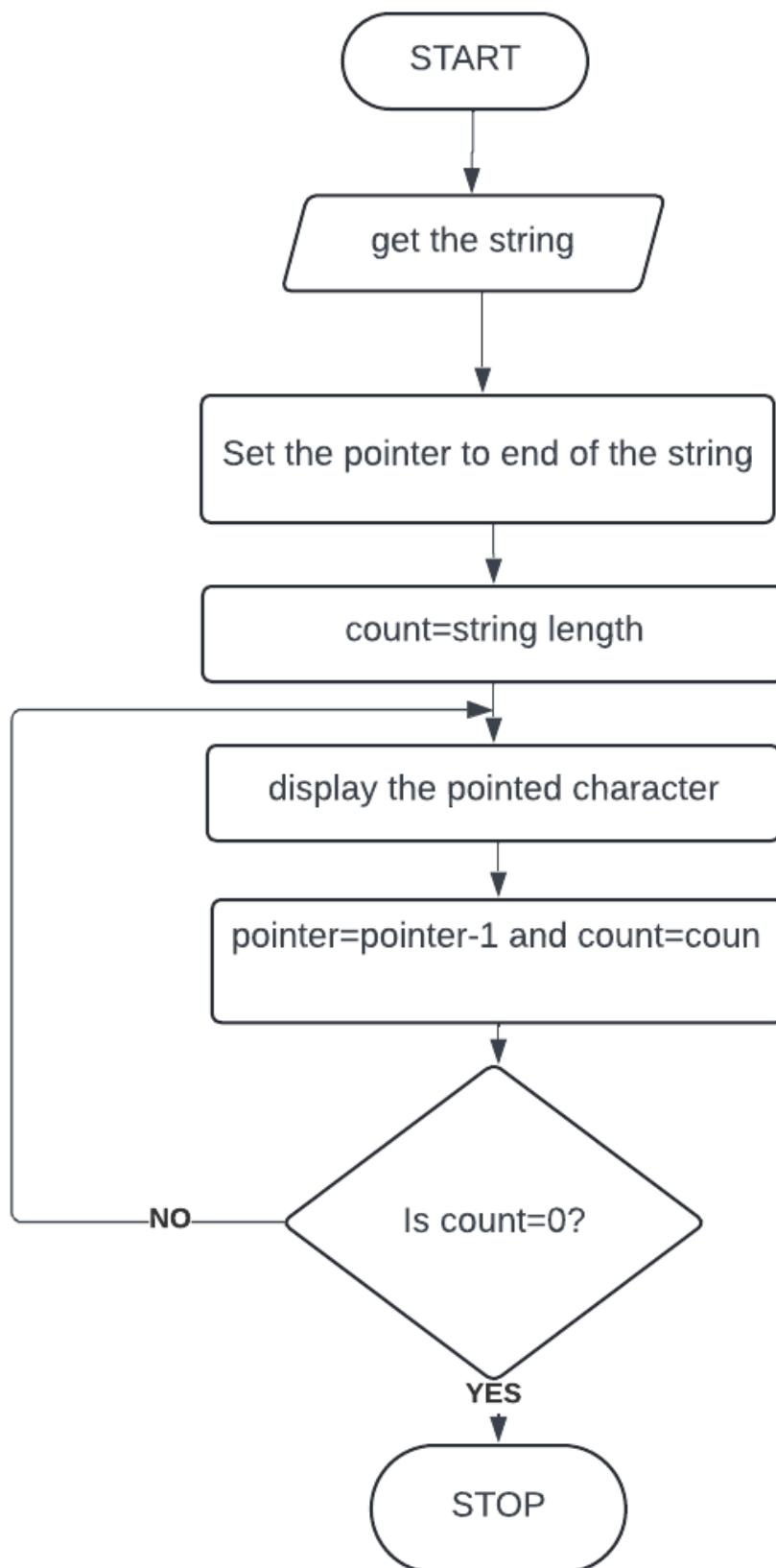
**Theory:**

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:**





**Code :**

org 100h

.data

m1 db 10, 13, 'Enter the string :\$'

m2 db 10, 13, 'The 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 cl, [buff+1]

lea bx, buff+2

l1:

mov dx, [bx]

mov ah, 02h

int 21h

inc bx



### Output :

```
01 org 100h
02
03 .data
04 m1 db 10, 13, 'Enter the string :$'
05 m2 db 10, 13, 'The string is :$', '$'
06 buff db 80
07
08 .code
09 lea dx, m1
10
11 mov ah, 09h
12 int 21h
13
14 lea dx, buff
15
16 mov ah, 0ah
17 int 21h
18
19 lea dx, m2
20
21 mov ah, 09h
22 int 21h
23
24 mov cl, [buff+1]
25 lea bx, buff+2
26
27 l1:
28 mov dx, [bx]
29
30 mov ah, 02h
31 int 21h
32
33 inc bx
34 loop l1
```

emulator: exp6.com

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

|    | H    | L  |
|----|------|----|
| AX | 02   | 69 |
| BX | 01   | 2D |
| CX | 00   | 00 |
| DX | 00   | 69 |
| CS | 0700 |    |
| IP | 0161 |    |
| SS | 0700 |    |
| SP | FFFE |    |
| BP | 0000 |    |
| SI | 0000 |    |
| DI | 0000 |    |
| DS | 0700 |    |
| ES | 0700 |    |

0700:0161

07160: 90 144 6

07161: E4 244 6

07162: 00 000 NULL

07163: 00 000 NULL

07164: 00 000 NULL

07165: 00 000 NULL

07166: 00 000 NULL

07167: 00 000 NULL

07168: 00 000 NULL

07169: 00 000 NULL

0716A: 00 000 NULL

0716B: 00 000 NULL

0716C: 00 000 NULL

0716D: 00 000 NULL

0716E: 00 000 NULL

0716F: 00 000 NULL

07170: 00 000 NULL

07171: 00 000 NULL

07172: 00 000 NULL

07173: 00 000 NULL

07174: 00 000 NULL

07175: 00 000 NULL

original source co...

17 int 21h

18

19 lea dx, m2

20

21 mov ah, 09h

22 int 21h

23

24 mov cl, [buff+1]

25 lea bx, buff+2

26

27 l1:

28 mov dx, [bx]

29

30 mov ah, 02h

31 int 21h

32

33 inc bx

34 loop l1

emulator screen (80x25 chars)

Enter the string:bari

The string is:bari

clear screen change font 0.16

### Conclusion :

In conclusion, the task of reversing a word in a string requires careful consideration of string manipulation and algorithmic efficiency. By implementing a systematic approach, we can successfully reverse the order of characters within a word while preserving the integrity of the overall string. Through this process, we enhance our understanding of string manipulation techniques and sharpen our problem-solving skills in programming. As we continue to explore and tackle similar challenges, we reinforce our ability to navigate complex problems and produce effective solutions in the realm of software development.

