

|  |  |
| --- | --- |
| **Name:** | BARI ANKIT VINOD |
| **RollNo:** | 65 |
| **Class/Sem:** | SE/IV |
| **ExperimentNo.:** | 6 |
| **Title:** | Toperformprogramtoreversethewordinstring |
| **DateofPerformance:** | 14/02/24 |
| **DateofSubmission:** | 24/02/24 |
| **Marks:** |  |
| **SignofFaculty:** |  |



**Aim:**AssemblyLanguageProgramtoreversethewordinstring.

**Theory:**

Thisprogramwillreadthestringenteredbytheuserandthenreverseit.Reverseastringisthe technique thatreversesorchangestheorderofagivenstringsothatthelastcharacterofthestringbecomesthefirst characterofthestringandsoon.

**Algorithm:**

1. Start

2. Initializethedatasegment

3. Displaythemessage-1

4. Inputthestring

5. Displaythemessage2

6. TakecharacterscountinDI

7. Pointtotheendcharacterandreadit

8. Displaythecharacter

9. Decrementthecount

10.Repeatuntilthecountiszero

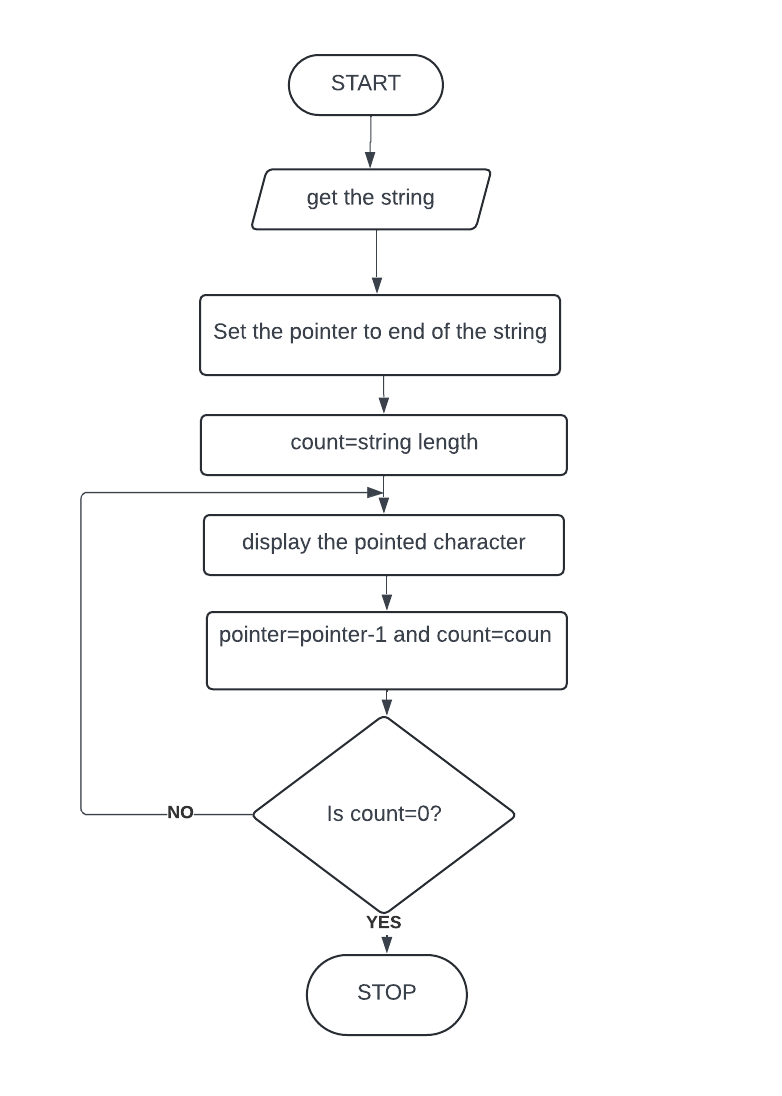
11. ToterminatetheprogramusingDOSinterrupt

a. InitializeAHwith4ch

b. CallinterruptINT21h

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]

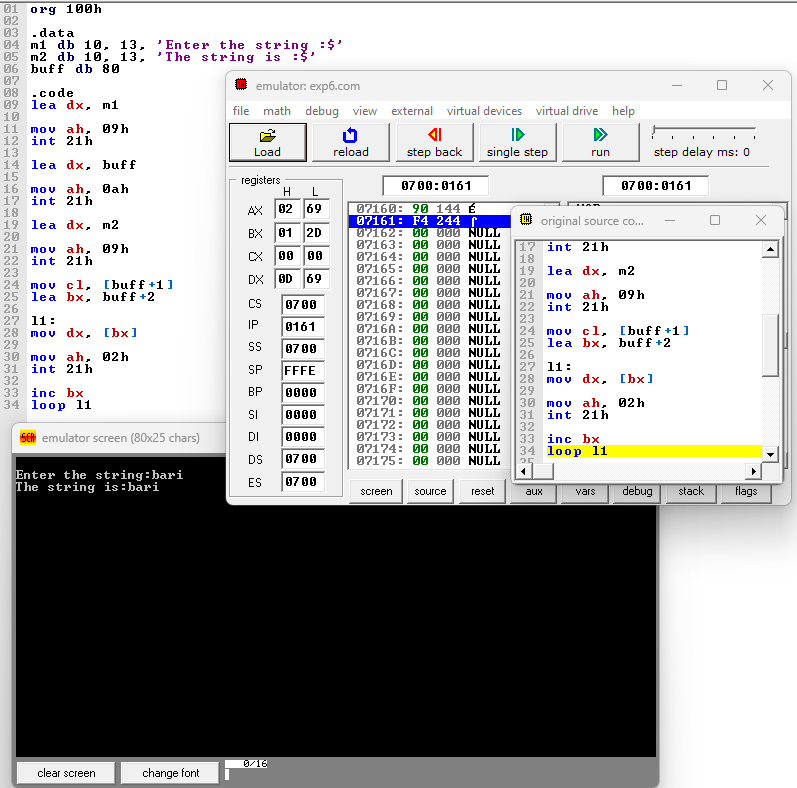
mov ah, 02h

int 21h

inc bx

loop l1

**Output :**



**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.

