

Reverse words in a string

BRUTE FORCE :

If a space occurs while traversing and a word is stored in a temporary string i.e temporary string is not empty then we push it into the stack. Stack is basically used for reversing.

Each word is popped and a space is added except for the last word.

Code :

```
#include<bits/stdc++.h>
string reverseString(string &str){
    // Write your code here.
    stack<string> s;
    str+=" ";
    string st="";
    string ans="";
    for(char ch: str)
    {
        if(ch==' ' && st!="")
        {
            s.push(st);
            st="";
        }
        else if(ch!=' ')
        {
            st+=ch;
        }
    }
    while(s.size()>1)
    {
        string word=s.top();
        s.pop();
        ans+=word+" ";
    }
    if(!s.empty())
    {
        string word=s.top();
        s.pop();
        ans+=word;
    }
    return ans;
}
```

- Time Complexity : $O(N)$
- Space Complexity : $O(N)$

OPTIMIZED APPROACH :

We eliminate stack and directly push into ans.

Code :

```
#include<bits/stdc++.h>
string reverseString(string &str){
    // Write your code here.
    stack<string> s;
    str+=" ";
    string st="";
    string ans="";
    for(int i=0;i<str.size();i++)
    {
        if(str[i]==' ' && st!="")
        {
            ans=st+" "+ans;
            st="";
        }
        else if(str[i]!=' ')
        {
            st+=str[i];
        }
    }
    return ans;
}
```

- Time Complexity : $O(N)$
- Space Complexity : $O(1)$