Statement

↳ You are given a string, sentence, comprising words and leading or trailing spaces or multiple spaces between words. Your task is to reverse the order of its words without affecting the order of letters within the given word. Return the modified sentence.

Approach

↳ Remove leading and trailing whitespace from the sentence.

↳ Split sentence into words based on spaces and store in array result.

↳ Initialize two pointers:

- The left pointer starts from the first word in result (index 0)
- The right pointer starts from the last word in result

↳ Iterate over result until the left pointer becomes greater than the right pointer.

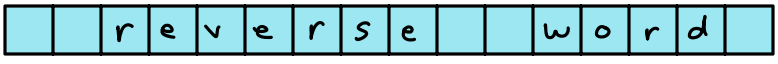
- Swap the words at left and right
- Increment left by one
- Decrement right by one

↳ Join the words in result with a single space between each word.

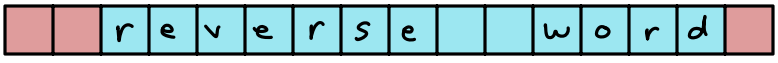
↳ Return final sentence

Visualization

i) Reverse the string " reverse word "

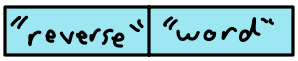


ii) First, remove any trailing or leading whitespaces in the string

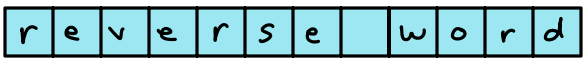


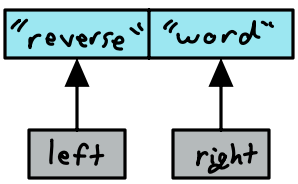
iii) Remove any multiple spaces in the whole string

After removing multiple spaces, store the remaining words in result array

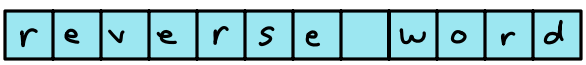


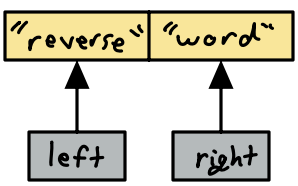
iv) Initialize two pointers left and right, on both sides of the string result.



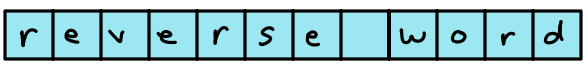


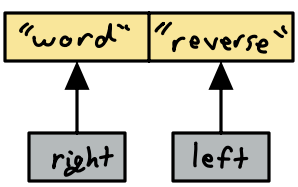
v) Swap result[left] with result[right] until left becomes greater than or equal to right.





vi) As left is now greater than right, exit the loop and return final output stored in result





Code

```
string ReverseWords(string sentence){
   int left = 0, right = sentence.size()-1;

   while (left<=right && sentence[left]==" "){
      left++;
   }

   while(right>=left && sentence[right]==" "){
      right--;
   }

   vector<string> words;
   stringstream ss(sentence.substr(left, right-left+1));
   string word;

   while (ss>>word){
      words.push_back(word);
   }

   int i=0, j=words.size()-1;

   while (i<j){
      swap(words[i], words[j]);
      i++;
      j--;
   }

   string result;
   for(int k=0; k<words.size(); k++){
      if(k>0){
         result += " ";
      }
      result += words[k];
   }

   return result;
}
```