1.Easy\02.Reverse_words_in_string.cpp

```
/*
Question:
Given an input string s, reverse the order of the words.
A word is defined as a sequence of non-space characters. The words in s will be separated by at least one space.
Return a string of the words in reverse order concatenated by a single space.
Note that s may contain leading or trailing spaces or multiple spaces between two words. The returned string should only have a single space separating the words. Do not include any extra spaces.

Example:
Input: s = "the sky is blue"
Output: "blue is sky the"

Approach:
- Initialize an empty string 'ans' to store the reversed words.
- Initialize 'start' and 'end' variables to keep track of the start and end indices of each word.
- Iterate through the input string 's'.
- Ignore leading spaces by advancing the iterator 'i' until a non-space character is found.
- Set 'start' to the current index 'i'.
- Find the end index 'end' of the current word by advancing 'i' until a space or the end of the string is encountered.
- Extract the current word using the substr() function and store it in a temporary string 'temp'.
- Reverse the characters in 'temp'.
- Append 'temp' to 'ans' with a space delimiter.
- Reverse the characters in 'ans' to get the reversed order of words.
- Remove any leading or trailing spaces in 'ans'.
- Return the resulting string 'ans'.

Code:
*/

string reverseWords(string s) {
    string ans = "";
    int start = -1, end = -1;
    for(int i=0; i<s.size(); i++){
        while(s[i]==' ')
            i++;
        start = i;
        while(i<s.size() && s[i]!=' ')
            i++;
        end = i;
        string temp = s.substr(start,end-start);
        reverse(temp.begin(),temp.end());
        ans = ans+" "+temp;
    }
    reverse(ans.begin(),ans.end());
    int i=0, j=ans.size()-1;
    while(ans[i]==' ')
        i++;
```

```cpp
    while(ans[j]==' ')
        j--;
    ans = ans.substr(i,j-i+1);
    return ans;
}

/*
Time Complexity: O(n), where n is the length of the input string 's'.
Space Complexity: O(n), where n is the length of the input string 's'.
*/

```