

# Reverse Words in a String

Try to solve the Reverse Words in a String problem.

We'll cover the following ^

- Statement
- Examples
- Understand the problem
- Figure it out!
- Try it yourself

## Statement

Given a sentence, reverse the order of its words without affecting the order of letters within a given word.

### Constraints:

- Sentence contains English uppercase and lowercase letters, digits, and spaces.
- $1 \leq \text{sentence.length} \leq 10^4$
- The order of the letters within a word is not to be reversed.

**Note:** The input string may contain leading or trailing spaces or multiple spaces between words. The returned string, however, should only have a single space separating each word. Do not include any extra spaces.

## Examples

**Sample example 1**

**Input**

Input String

"Hello Friend"

-----

**Output**

Reversed String

"Friend Hello"

1 of 3



## Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

## Reverse Words in a String

1

What should be the output if the following sentence is given as an input?

“The quick brown fox jumped over a lazy dog”

A) lazy dog a over fox jumped brown quick The

B) dog lazy over a jumped fox brown quick The

C) dog lazy jumped jumped fox brown quick The

D) dog lazy a over jumped fox brown quick The

Submit Answer



Question 1 of 4  
0 attempted



Reset Quiz ↻

## Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.



Drag and drop the cards to rearrange them in the correct sequence.

Reverse the entire string.

Start iterating over the reversed string using two pointers, **start** and **end**, initially set at index 0.

While iterating over the string, when **end** points to a space, reverse the word pointed by **start** and **end-1**.

Once the word has been reversed, update the **start**



and **end** to the start index of the next word.

Repeat the process until the entire string is iterated and return the string.

Reset

Show Solution

Submit

## Try it yourself

Implement your solution in **main.java** in the following coding playground. We have provided a useful code template in the other file that you may build on to solve this problem.

Java

main.java  
TwoPointers.java

```
39 // Trim leading, trailing and multiple spaces
40 static String cleanSpaces(char[] a, int n) {
41     // Convert character array to string
42     String str = new String(a, 0, n);
43
44     // Trim extra spaces at the beginning and end of the string
45     // And replace multiple spaces with a single space
46
47
48     return str;
49 }
50
51
52 public static void main(String[] args) {
53     String[] inputs = {
54         "We love Java",
55         "To be or not to be",
56         "You are amazing", "Hello    World", "Hey"};
57
58     for(int i=0; i<inputs.length; i++){
59         System.out.print(i+1);
60         System.out.println("\tActual string:\t\t"+ inputs[i]);
61         System.out.println("\tReversed String:\t"+ reverseWords(inputs[i]));
62         System.out.println(new String(new char[100]).replace('\0', '-'));
63     }
64 }
65 }
66 }
```

Powered by AI

Submit

Test Cases

Results

Case 1Case 2Case 3

Input #1

"We love Java"

Reverse Words in a String

← Back

Next →

Solution: Sum of Thre...

Solution: Reverse Wor...

