

## 125. Valid Palindrome

Solved ✓

Easy

Topics

Companies

A phrase is a **palindrome** if, after converting all uppercase letters into lowercase letters and removing all non-alphanumeric characters, it reads the same forward and backward.

Alphanumeric characters include letters and numbers.

Given a string `s`, return `true` if it is a **palindrome**, or `false` otherwise.

## Example 1:

**Input:** `s = "A man, a plan, a canal: Panama"`

**Output:** `true`

**Explanation:** "amanaplanacanalpanama" is a palindrome.

## Example 2:

**Input:** `s = "race a car"`

**Output:** `false`

**Explanation:** "raceacar" is not a palindrome.

## Example 3:

**Input:** `s = ""`

**Output:** `true`

**Explanation:** `s` is an empty string "" after removing non-alphanumeric characters.

## &lt;/&gt; Code

C++ Auto

```
1 class Solution {
2 public:
3     bool isPalindrome(string s) {
4         int n = s.length(); //Initializing variable 'n' with length of string 's'
5         bool result = true; //Taking variable 'result' as true initially
6         // Loop to traverse characters of string 's'
7         for (int i = 0; i < n; i++) {
8             if (isupper(s[i])) {
9                 s[i] = tolower(s[i]); // Convert uppercase character to lowercase
10            } else if (s[i] == ' ') { // Remove space character
11                s.erase(i, 1);
12                i--; // Decrement index to account for the removed character
13                n--; // Decrement length of the string after removal
14            } else if (!isalnum(s[i])) { // Remove non-alphanumeric character
15                s.erase(i, 1);
16                i--; // Decrement index to account for the removed character
17                n--; // Decrement length of the string after removal
18            }
19        }
20        //Loop to check whether string 's' is a palindrome or not
21        for (int j = 0; j < n / 2; j++) {
22            if (s[j] != s[n - j - 1]) { //Comparing characters from starting and end of string 's'
23                result = false; // If characters don't match, set result to false
24            }
25        }
26        return result; // Return the final result indicating whether the modified string is a palindrome
27    }
28 };
```

Saved to local

Ln 15, Co



Problem List



Run



Submit



0

[Description](#) | [Editorial](#) | [Solutions](#) | [Submissions](#)

## 242. Valid Anagram

Solved

Easy



Topics



Companies

Given two strings `s` and `t`, return `true` if `t` is an anagram of `s`, and `false` otherwise.

An **Anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

### Example 1:

**Input:** `s = "anagram", t = "nagaram"`

**Output:** `true`

### Example 2:

**Input:** `s = "rat", t = "car"`

**Output:** `false`

### Constraints:

Code

C++ Auto

```
1 class Solution {
2 public:
3     bool isAnagram(string s, string t) {
4         // Check if the lengths of the two strings are different
5         if (s.length() != t.length()) {
6             return false; // If lengths are different, they can't be anagrams
7         }
8         bool result = true; // Assume the strings are anagrams initially
9
10        // Sort both strings making it easier to compare character by character
11        sort(s.begin(), s.end());
12        sort(t.begin(), t.end());
13
14        int n = t.length(); // Initializing variable 'n' with length of string 't'
15
16        // Loop to compare characters of both strings after sorting
17        for (int i = 0; i < n; i++) {
18            if (s[i] != t[i]) {
19                result = false; // If characters at the same position are different, not anagrams
20            }
21        }
22
23        return result; // Return the final result indicating whether the strings are anagrams
24    }
25 };
26
```