

# Strings – Palindrome Check

## 1. Introduction

A **palindrome** is a string that **reads the same forwards and backwards**.

Checking whether a string is a palindrome is a **common string problem** and helps learners understand **string traversal, comparison, and logical thinking**.

---

## 2. What is a Palindrome?

A string is called a **palindrome** if the sequence of characters remains the same when reversed.

### Examples:

- "madam" → Palindrome
  - "racecar" → Palindrome
  - "hello" → Not a palindrome
- 

## 3. Why is Palindrome Checking Important?

Palindrome checking is important because:

- It strengthens understanding of string traversal
  - It introduces comparison logic
  - It is frequently asked in interviews
  - It is used in pattern recognition and validation problems
  - It forms the base for advanced string algorithms
- 

## 4. Basic Idea Behind Palindrome Check

The basic idea is to **compare characters from the beginning and end** of the string:

- If all corresponding characters match → palindrome
  - If any mismatch occurs → not a palindrome
- 

## 5. Logic for Palindrome Check (Plain English)

### Two-Pointer Approach:

1. Place one pointer at the start of the string
  2. Place another pointer at the end of the string
  3. Compare characters at both pointers
  4. If they are equal, move pointers inward
  5. If they are not equal, stop and declare not a palindrome
  6. Continue until pointers cross or meet
- 

## 6. Visualization of Palindrome Checking

Example string:

"LEVEL"

Comparison flow:

L == L → OK  
E == E → OK  
V → Middle reached

Since all comparisons match, the string is a palindrome.

---

## 7. Alternative Method: Using Reversal (Conceptual)

Another way to check palindrome is:

1. Reverse the string
2. Compare the reversed string with the original

3. If both are equal → palindrome

This method is simple but uses extra space.

---

## 8. Time and Space Complexity

Method	Time Complexity	Space Complexity
Two-pointer method	$O(n)$	$O(1)$
Reverse & compare	$O(n)$	$O(n)$

The **two-pointer approach** is preferred due to better space efficiency.

---

## 9. Special Cases

- **Empty string:** Considered a palindrome
  - **Single character:** Always a palindrome
  - **Case sensitivity:** "Madam" may not be palindrome unless case-handled
  - **Spaces & symbols:** Often ignored in advanced versions
- 

## 10. Real-World Applications

- Word games and puzzles
  - DNA sequence analysis
  - Text validation
  - Pattern recognition
  - Competitive programming problems
- 

## 11. Common Variations of Palindrome Problems

- Palindrome ignoring case and spaces
- Numeric palindrome
- Longest palindromic substring

- Palindrome using recursion
  - Palindrome in linked lists
- 

## 12. Summary

- A palindrome reads the same forwards and backwards
  - Comparison is done from both ends
  - Two-pointer technique is efficient
  - Time complexity is  $O(n)$
  - A fundamental string problem in DSA
-