# Documentation: Valid Palindrome

## Problem Statement

A phrase is considered a palindrome if, after converting all uppercase letters to lowercase and removing all non-alphanumeric characters, it reads the same forward and backward. Alphanumeric characters include both letters and numbers.

Given a string s, determine if it is a palindrome. Return true if it is a palindrome, or false otherwise.

## Example 1:

- **Input:** s = "A man, a plan, a canal: Panama"
- **Output:** true
- **Explanation:** After processing, the string becomes "amanaplanacanalpanama", which is a palindrome.

## Example 2:

- **Input:** s = "race a car"
- **Output:** false
- **Explanation:** After processing, the string becomes "raceacar", which is not a palindrome.

## Example 3:

- **Input:** s = " "
- **Output:** true
- **Explanation:** After processing, the string becomes an empty string "", which reads the same forward and backward. Therefore, it is a palindrome.

## Constraints

- The length of the string s is between 1 and 200,000 characters (inclusive).
- The string s consists only of printable ASCII characters.

## Solution Explanation

*The approach to solve this problem involves the following steps:*

1. **Filtering and Conversion:**

- Convert all uppercase letters to lowercase.
- Remove all non-alphanumeric characters.
- This is done using a list comprehension that iterates through each character of the string, checking if it is alphanumeric, and converting it to lowercase if it is.

2. **Palindrome Check:**

- Compare the filtered list of characters with its reverse.
- If they are the same, the string is a palindrome; otherwise, it is not.

## Example Usage

1. **Example 1:**

- *Input:* "A man, a plan, a canal: Panama"
- *Output:* true

2. **Example 2:**

- *Input:* "race a car"
- *Output:* false

3. **Example 3:**

- *Input:* " "
- *Output:* true

The provided solution efficiently checks if a given string is a palindrome by focusing only on its alphanumeric characters and ignoring case differences.