

Valid Palindrome

Try to solve the Valid Palindrome problem.

We'll cover the following ^

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

Statement

Write a function that takes a string, `s`, as an input and determines whether or not it is a palindrome.

Note: A **palindrome** is a word, phrase, or sequence of characters that reads the same backward as forward.

Constraints:

- $1 \leq s.length \leq 2 \times 10^5$
- The string `s` will not contain any white space and will only consist of ASCII characters.

Examples

Sample example 1

Input

A	B	C	B	A
---	---	---	---	---

Output

Output	TRUE
--------	------

1 of 2

<

>

▶

↶

+

⌂

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 are solving the correct problem:

Valid Palindrome

?

Tt

🌙

1

"abab" is a palindrome.

A) True

B) False

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 on how to solve this problem.

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

Initialize two pointers at the beginning and end of the string.

Check whether or not the current pair of characters is identical.

If they are not identical, return FALSE. Otherwise, move both pointers by one index toward the middle.

Keep traversing them toward the middle until they meet.

If we reach the middle of the string without finding a mismatch, return TRUE.

Reset

Show Solution

Submit

?

Tt

🔄

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.

main.java

TwoPointers.java

```
11         return false;
12     }
13     left= left+1;
14     right= right-1;
15
16 }
```

```
20 public static void main(String[] args)
21 {
22     String[] testCases=
23     {"kayak","hello","RACEACAR","A"
24     ,"ABCDABCD",
25     "DCBAABCD","ABCBA"
26     };
27     for(int k=0; k<testCases.length; k++){
28
29         System.out.println("\tThe input string is " + testCases[k] + "' and the
30         System.out.println("\n\tIs it a palindrome?..... " + isPalindrome(testC
31         }
32
33
34
35     }
36
37 }
```

Powered by AI

▶

Submit

Test Cases

Results

Case 1

Case 2

Case 3

Input #1

"kayak"

Valid Palindrome

← Back

Next →

Two Pointers: Introduc...

Solution: Valid Palindr...

✓ Completed