

# Is Palindrome

str = "abcba"

The string "abcba" is shown in red. The first 'a' and the last 'a' are circled. Below the string, five upward-pointing arrows are labeled with indices: 'i' under the first 'a', 'i' under the first 'b', 'j' under the first 'c', 'j' under the second 'b', and 'j' under the second 'a'.

pseudo  
code

→ declare  $i = 0$   
and  $j = \text{str.length}() - 1$

→ iterate until  $i < j$

→ check if  $\text{char at } i \neq \text{char at } j$   
    (  $\text{Syso}(\text{False})$  )

→  $i++$ ,  $j--$

→  $\text{Syso}(\text{True})$

code

```
public static void checkPal(String str) {  
    int i = 0;  
    int j = str.length() - 1;  
    while ( i < j ) {  
        char ch1 = str.charAt(i);  
        char ch2 = str.charAt(j);  
        if ( ch1 != ch2 ) {  
            System.out.println("Not a Palindrome");  
            return;  
        }  
        i++;  
        j--;  
    }  
    System.out.println("Palindrome");  
}
```

# code for leetcode 125

```
class Solution {
    public boolean isPalindrome(String s) {
        String str = "";
        for (int i = 0; i < s.length(); i++) {
            char c = s.charAt(i);
            if ( (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') || (c >= '0' && c <= '9') ) {
                str += c;
            }
        }
        str = str.toLowerCase();
        return checkPal(str);
    }

    public boolean checkPal(String str) {
        int i = 0;
        int j = str.length() - 1;
        while ( i < j ) {
            char ch1 = str.charAt(i);
            char ch2 = str.charAt(j);
            if ( ch1 != ch2 ) {
                return false;
            }
            i++;
            j--;
        }
        return true;
    }
}
```

## Locate the Target String

0 1 2 3 4 5 6 7  
str = "geekster"  
                        

target = "st"  
            

ans = 4

0 1 2 3 4 5 6 7 8 9 10 11  
str = "programingra"  
          ↑     └─┘           └─┘

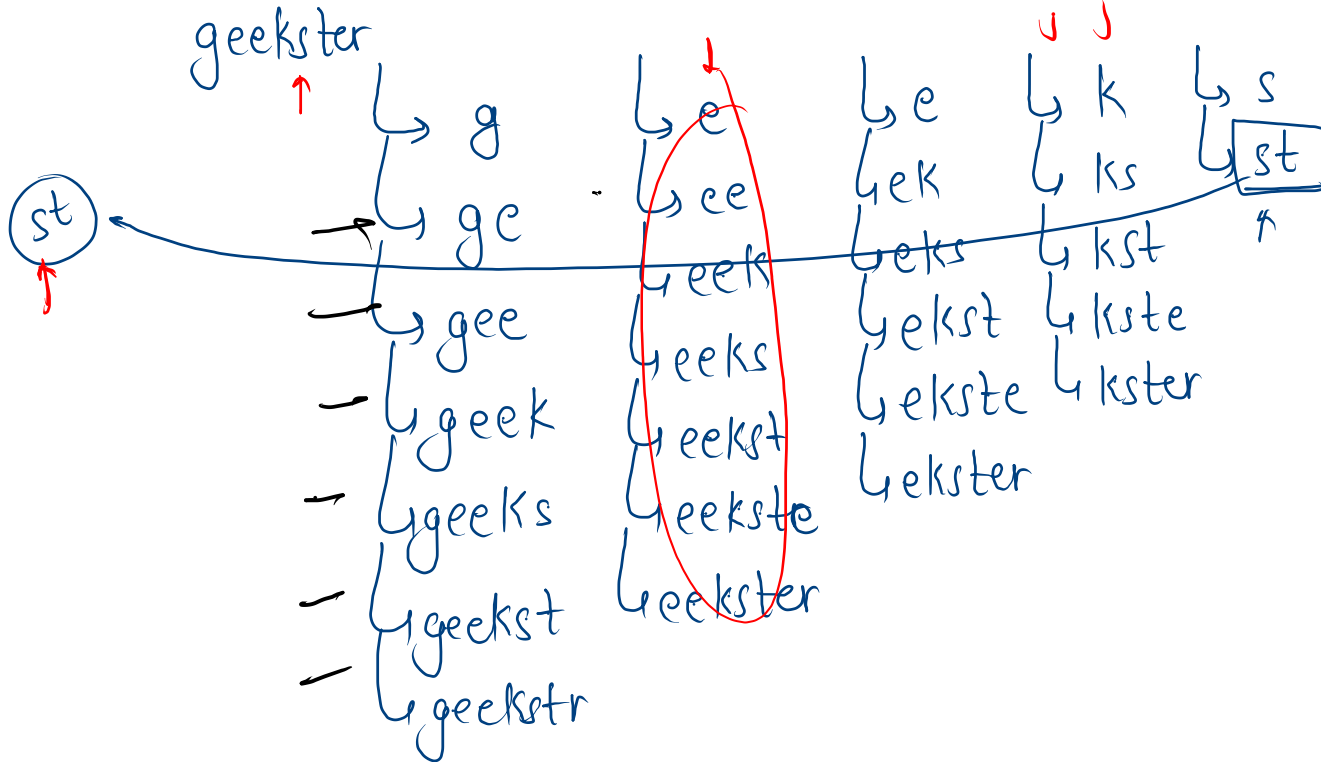
target = "ra"

ans = 4

str = "geekster"

geekspst

۱۰۰



## approch 2

```
public static int locateTarget(String str, String tar) {  
    for (int i = 0; i <= str.length() - tar.length(); i++) {  
        for (int j = 0; j < tar.length(); j++) {  
            if (tar.charAt(j) != str.charAt(i + j)) {  
                break;  
            }  
            if (j == tar.length() - 1) {  
                return i;  
            }  
        }  
    }  
    return -1;  
}
```

Confirmation

$i=0, j=0, (s \neq g)$  break

$i=1, j=0, (s \neq e)$  break

$i=2, j=0, (s \neq e)$  break

$i=3, j=0, (s \neq k)$  break

$i=4, j=0, (s = s)$

$j=1, (t = t)$

return 4

Diagram illustrating the string matching process:

str = "g e e k s t e r"  
target = "s t"

The diagram shows the indices i and j. i points to the start of the substring "st" in str (index 4). j points to the start of the target string "st" (index 0). A blue bracket under "st" in str and another blue bracket under "st" in target indicate the matching substring.

```

public static int locateTarget(String str, String tar) {
    for (int i = 0; i <= str.length() - tar.length(); i++) {
        for (int j = 0; j < tar.length(); j++) {
            if (tar.charAt(j) != str.charAt(i + j)) {
                break;
            }
            if (j == tar.length() - 1) {
                return i;
            }
        }
    }
    return -1;
}

```

ans = 4

$T.C = O(\text{str.length()} * \text{tar.length()})$

$S.C = O(1)$

$i$   
 $\downarrow$   
 $\text{str} = \text{"abcde"}$   
 0 1 2 3 4 5 6  
 $\text{tar} = \text{"cde"}$   
 0 1 2  
 $\uparrow$   
 $j$

$i=0, j=0$  ( $c \neq a$ ) break

$i=1, j=0$  ( $c \neq b$ ) break

$i=2, j=0$  ( $c = c$ )

$j=1$  ( $d \neq d$ )

$j=2$  ( $e \neq c$ ) break

$i=3, j=0$  ( $c \neq d$ ) break

$i=4, j=0$  ( $c = c$ )

$j=1$  ( $d \neq d$ )

$j=2$  ( $e \neq e$ )