

Full Name:

Tags Score:

Mock Test > contato@guifr.com.br

**GUILHERME FERNANDES** Email: contato@guifr.com.br Test Name: **Mock Test** 17 Jun 2022 20:39:42 IST Taken On: Time Taken: 14 min 23 sec/ 30 min Invited by: Ankush 17 Jun 2022 20:39:35 IST Invited on: Skills Score:

Algorithms 105/105

Problem Solving 105/105

problem-solving 105/105

Core CS 105/105 Easy 105/105

Strings 105/105

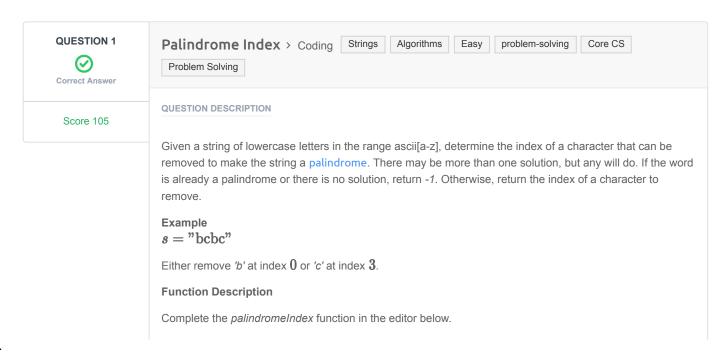
100% 105/105

scored in Mock Test in 14 min 23 sec on 17 Jun 2022 20:39:42 IST

#### **Recruiter/Team Comments:**

No Comments.





palindromeIndex has the following parameter(s):

• string s: a string to analyze

#### Returns

• int: the index of the character to remove or -1

### **Input Format**

The first line contains an integer q, the number of queries. Each of the next q lines contains a query string s.

### Constraints

- $1 \le q \le 20$
- $1 \le \text{length of } s \le 10^5 + 5$
- All characters are in the range ascii[a-z].

### Sample Input

```
STDIN Function

q = 3

aaab s = 'aaab' (first query)

baa s = 'baa' (second query)

aaa s = 'aaa' (third query)
```

### **Sample Output**

```
3
0
-1
```

### **Explanation**

Query 1: "aaab"

Removing 'b' at index 3 results in a palindrome, so return 3.

Query 2: "baa"

Removing 'b' at index 0 results in a palindrome, so return 0.

Query 3: "aaa'

This string is already a palindrome, so return -1. Removing any one of the characters would result in a palindrome, but this test comes first.

**Note:** The custom checker logic for this challenge is available here.

## **CANDIDATE ANSWER**

# Language used: Java 8

```
class Result {

/*

* Complete the 'palindromeIndex' function below.

* * The function is expected to return an INTEGER.

* The function accepts STRING s as parameter.

* //

public static int palindromeIndex(String s) {

StringBuilder order = new StringBuilder(s);

StringBuilder reverse = new StringBuilder(s);

reverse reverse();
```

```
reverse reverse () '
           if (order.toString().equals(reverse.toString())) {
              return -1;
           int newResult = -1;
           String[] split = s.split("");
          int start = 0;
          int last = s.length() - 1;
           while (start < last) {</pre>
               if (split[last].equals(split[start])) {
                  last--;
                  start++;
              } else {
                   if (newResult != -1) return -1;
                  order = new StringBuilder(s.substring(start + 1, last + 1));
                   reverse = new StringBuilder(order);
                   reverse.reverse();
                  if (order.toString().equals(reverse.toString())) {
                       newResult = start;
                       start++;
                  } else {
40
                      newResult = last;
                       last--;
44
              }
          }
          return newResult;
49
      }
52 }
54
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.1583 sec	29.6 KB
Testcase 2	Medium	Hidden case	Success	5	0.17 sec	29.8 KB
Testcase 3	Medium	Hidden case	Success	5	0.1495 sec	29.8 KB
Testcase 4	Medium	Hidden case	Success	5	0.1876 sec	29.5 KB
Testcase 5	Medium	Hidden case	Success	5	0.1576 sec	29.4 KB
Testcase 6	Medium	Hidden case	Success	5	0.3485 sec	42.2 KB
Testcase 7	Medium	Hidden case	Success	5	0.2377 sec	43.1 KB
Testcase 8	Medium	Hidden case	Success	5	0.3724 sec	56.4 KB
Testcase 9	Hard	Hidden case	Success	10	0.468 sec	41.3 KB
Testcase 10	Hard	Hidden case	Success	10	0.3464 sec	45.6 KB
Testcase 11	Hard	Hidden case	Success	10	0.3533 sec	46.1 KB
Testcase 12	Hard	Hidden case	Success	10	0.1333 sec	29.7 KB
Testcase 13	Hard	Hidden case	Success	10	0.3316 sec	42 KB

	Testcase 14	Hard	Hidden case	Success	10	0.5644 sec	47.1 KB
	Testcase 15	Hard	Hidden case	Success	10	0.3138 sec	46.5 KB
N	o Comments						

PDF generated at: 17 Jun 2022 15:25:33 UTC