H

Full Name:

Đào Nguyễn Huy Hoàng

Email:

hoang.dnh210380@sis.hust.edu.vn

Test Name:

Mock Test

Taken On:

13 Apr 2023 22:18:07 IST

Time Taken:

3 min 11 sec/ 30 min

Invited by:

Ankush

Invited on:

13 Apr 2023 22:17:58 IST

Skills Score:

Tags Score:

Algorithms 105/105

Core CS 105/105

Easy 105/105

Problem Solving 105/105
Strings 105/105

problem-solving 105/105

100% 105/105

scored in **Mock Test** in 3 min 11 sec on 13 Apr 2023 22:18:07 IST

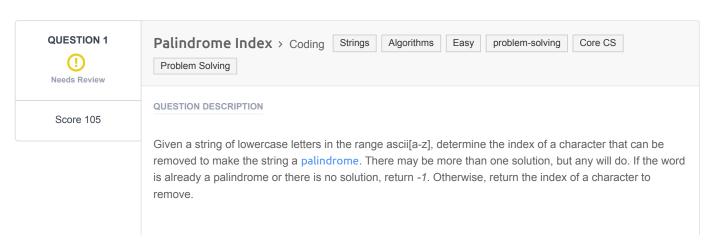
Recruiter/Team Comments:

No Comments.

Plagiarism flagged

We have marked questions with suspected plagiarism below. Please review.





Example s = "bcbc"

Either remove b' at index 0 or c' at index 3.

Function Description

Complete the *palindromeIndex* function in the editor below.

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 ${\it q}$, 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

----

3  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 b' are sults in a palindrome, so return b'.

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: C

```
// palindrome-index
#include <stdio.h>
#include <string.h>
int main()

{
   int i, j, k, t, n, flag;
   char str[100010];
```

```
8
       scanf("%d", &t);
9
       while (t--)
           flag = 0;
           scanf("%s", str);
           n = strlen(str);
14
           i = 0;
           j = n - 1;
           while (i < j)
               if (str[i] != str[j])
                   if (str[i] == str[j - 1] && str[i + 1] != str[j])
                       printf("%d\n", j);
                       flag = 1;
24
                       break;
                   else if (str[i] == str[j - 1] && str[i + 1] == str[j - 2])
                       printf("%d\n", j);
                       flag = 1;
                       break;
                   else if (str[i] == str[j - 1] \&\& str[i + 1] == str[j])
                       printf("%d\n", i);
                       flag = 1;
                       break;
                   else
                       printf("%d\n", i);
41
                       flag = 1;
                       break;
43
44
45
               i++;
               j--;
           if (flag == 0)
               printf("-1\n");
       }
       return 0;
52 }
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.032 sec	7.4 KB
Testcase 2	Medium	Hidden case	Success	5	0.055 sec	7.39 KB
Testcase 3	Medium	Hidden case	Success	5	0.0311 sec	7.4 KB
Testcase 4	Medium	Hidden case	Success	5	0.0235 sec	7.25 KB
Testcase 5	Medium	Hidden case	Success	5	0.0244 sec	7.35 KB
Testcase 6	Medium	Hidden case	Success	5	0.0333 sec	7.45 KB
Testcase 7	Medium	Hidden case	Success	5	0.0579 sec	7.1 KB
Testcase 8	Medium	Hidden case	Success	5	0.0387 sec	7.48 KB
Testcase 9	Hard	Hidden case	Success	10	0.0354 sec	6.92 KB
Testcase 10	Hard	Hidden case	Success	10	0.0286 sec	7.41 KB

Testcase 11	Hard	Hidden case	Success	10	0.0475 sec	7.58 KB				
Testcase 12	Hard	Hidden case	Success	10	0.0246 sec	7.52 KB				
Testcase 13	Hard	Hidden case	Success	10	0.0536 sec	7.61 KB				
Testcase 14	Hard	Hidden case	Success	10	0.0268 sec	7.57 KB				
Testcase 15	Hard	Hidden case	Success	10	0.0263 sec	7.39 KB				
No Comments										

PDF generated at: 13 Apr 2023 16:52:55 UTC