

# 【华为OD机考 统一考试机试C卷】 环中最长子串/字符成环找偶数O（C++ Java JavaScript Python C语言）

## 华为OD机考:统一考试 C卷 + D卷 + B卷 +A卷

目前在考C卷，经过两个月的收集整理，**C卷真题已基本整理完毕**

抽到原题的概率为2/3到3/3，**也就是最少抽到两道原题。请注意：大家刷完C卷真题，最好要把B卷的真题刷一下，因为C卷的部分真题来自B卷。**

另外订阅专栏还可以联系笔者开通在线 **OJ** 进行刷题，提高刷题效率。

**真题目录：**华为OD机考机试 真题目录（C卷 + D卷 + B卷 + A卷） + 考点说明

**专栏：**2023华为OD机试( B卷+C卷+D卷) (C++JavaJSPy)

**华为OD面试真题精选：**华为OD面试真题精选

**在线OJ：**点击立即刷题，模拟真实机考环境

## 注意

本题2022年考过

## 题目描述

给你一个字符串  $s$ ，字符串 $s$ 首尾相连成一个环形，请在环中找出 'o' 字符出现了偶数次最长子字符串的长度。

给你一个字符串  $s$ ，字符串 $s$ 首尾相连成一个环形，请在环中找出 'o' 字符出现了偶数次最长子字符串的长度。

## 输入描述

输入是一串小写字母组成的字符串

## 备注

$1 \leq s.length \leq 5 \times 10^5$

$s$  只包含小写英文字母

输出描述

输出是一个整数

示例1

输入

1 | alolobo

输出

1 | 6

说明：  
最长子字符串之一是“alolob”，它包含'o' 2个。

示例2

输入

1 | looxdolx

输出

1 | 7

说明：  
最长子字符串是“oxdolx”，由于是首尾连接在一起的，所以最后一个‘x’和开头的‘l’是连接在一起的，此字符串包含 2 个‘o’。

示例3

输入

1 | bcbcbc

输出

说明：

这个示例中，字符串“bcbcbc”本身就是最长的，因为‘o’都出现了0次。

## 解题思路

这段Java代码的核心解题思路主要包括以下几个步骤：

1. **读取用户输入**：首先，代码创建了一个Scanner对象，用于读取用户输入的字符串。
2. **获取字符串长度**：代码获取了字符数组的长度，这个长度值在后续的输出中会用到。
3. **统计‘o’字符的数量**：遍历字符数组，每遇到一个‘o’字符，就将计数器加1，以统计‘o’字符的总数量。
4. **根据‘o’字符的数量决定输出**：最后，代码检查了‘o’字符的数量。如果‘o’字符的数量是偶数，那么就输出字符串的长度；如果‘o’字符的数量是奇数，那么就输出字符串长度减1，因为字符串是环形的，只需要删除其中一个o然后就是偶数次最长子串。

## C++

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main() {
6      // 创建一个字符串变量，用于存储用户输入
7      string input;
8      // 读取用户输入的字符串
9      getline(cin, input);
10     // 获取字符串的长度
11     int len = input.size();
12     // 初始化‘o’字符的计数器
13     int num = 0;
14     // 遍历字符串，统计‘o’字符的数量
15     for (char chr : input) {
16         if (chr == 'o') {
17             num += 1;
18         }
19     }
20     // 如果‘o’字符出现的次数是偶数，则输出字符串的长度
21 }
```

```

21     if (num % 2 == 0) {
22         cout << len << endl;
23     } else {
24         // 如果'o'字符出现的次数是奇数, 则输出字符串长度减1
25         cout << len - 1 << endl;
26     }
27     return 0;
28 }

```

## Java

```

1  import java.util.Scanner;
2
3  public class Main {
4      public static void main(String[] args) {
5          // 创建一个Scanner对象, 用于读取用户输入
6          Scanner in = new Scanner(System.in);
7          // 读取用户输入的字符串
8          String input = in.nextLine();
9          // 将输入的字符串转换为字符数组
10         char[] chrs = input.toCharArray();
11         // 获取字符串的长度
12         int len = chrs.length;
13         // 初始化'o'字符的计数器
14         int num = 0;
15         // 遍历字符数组, 统计'o'字符的数量
16         for (char chr : chrs) {
17             if (chr == 'o') {
18                 num += 1;
19             }
20         }
21         // 如果'o'字符出现的次数是偶数, 则输出字符串的长度
22         if (num % 2 == 0) {
23             System.out.println(len);
24         } else {
25             // 如果'o'字符出现的次数是奇数, 则输出字符串长度减1
26             System.out.println(len - 1);
27         }
28     }
29 }

```

## JavaScript

```
1 // 引入readLine模块, 用于读取用户输入
2 const readline = require('readline').createInterface({
3   input: process.stdin,
4   output: process.stdout
5 });
6
7 // 询问用户输入字符串
8 readline.on('line', (input) => {
9   // 获取字符串的长度
10  const len = input.length;
11  // 初始化'o'字符的计数器
12  let num = 0;
13  // 遍历字符串, 统计'o'字符的数量
14  for (let chr of input) {
15    if (chr === 'o') {
16      num += 1;
17    }
18  }
19  // 如果'o'字符出现的次数是偶数, 则输出字符串的长度
20  if (num % 2 === 0) {
21    console.log(len);
22  } else {
23    // 如果'o'字符出现的次数是奇数, 则输出字符串长度减1
24    console.log(len - 1);
25  }
26  readline.close();
27 });
```

## Python

```
1 # 读取用户输入的字符串
2 input_str = input("")
3 # 获取字符串的长度
4 len_str = len(input_str)
5 # 初始化'o'字符的计数器
6 num = 0
7 # 遍历字符串, 统计'o'字符的数量
8 for chr in input_str:
```

```

9         if chr == 'o':
10             num += 1
11 # 如果'o'字符出现的次数是偶数，则输出字符串的长度
12 if num % 2 == 0:
13     print(len_str)
14 else:
15     # 如果'o'字符出现的次数是奇数，则输出字符串长度减1
16     print(len_str - 1)

```

## C语言

```

1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      char input[500001]; // 创建一个字符数组，用于存储用户输入的字符串，最大长度500000
6      scanf("%s", input); // 读取用户输入的字符串
7
8      int len = strlen(input); // 获取字符串的长度
9      int num = 0; // 初始化'o'字符的计数器
10
11 // 遍历字符串，统计'o'字符的数量
12 for (int i = 0; i < len; i++) {
13     if (input[i] == 'o') {
14         num++;
15     }
16 }
17
18 // 如果'o'字符出现的次数是偶数，则输出字符串的长度
19 if (num % 2 == 0) {
20     printf("%d\n", len);
21 } else {
22     // 如果'o'字符出现的次数是奇数，则输出字符串长度减1
23     printf("%d\n", len - 1);
24 }
25
26 return 0;
27 }

```

## 完整用例

### 用例1

looxdolx

### 用例2

alolobo

### 用例3

bcbcbc

### 用例4

o

### 用例5

oooooo

### 用例6

abcdefg

### 用例7

ooabcd

### 用例8

oabcd

### 用例9

oooabcd

### 用例10

oooabcdoo

## 文章目录

华为OD机考:统一考试 C卷 + D卷 + B卷 +A卷

注意

题目描述

输入描述

备注

输出描述

示例1

示例2

示例3

解题思路

C++

Java

JavaScript

Python

C语言

完整用例

用例1

用例2

用例3

用例4

用例5

用例6

用例7

用例8

用例9

用例10



# 机考真题 华为OD



CSDN @算法大师