

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，请你在环中找出 'l'、'o'、'x' 字符都恰好出现了偶数次最长子字符串的长度。

输入描述：

输入是一串小写的字母组成的字符串。

输出描述：

输出是一个整数

补充说明：

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

s 只包含小写英文字母。

示例 1

输入：

alolobo

输出：

6

说明：

最长子字符串之一是 "alolob"，它包含 'l'、'o' 各 2 个，以及 0 个 'x'。

示例 2

输入：

looxdolx

输出：

7

说明：

最长子字符串是 *"oxdolxl"*，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 *'l'*，2 个 *'o'*，2 个 *'x'*。

示例 3

输入：

bcbcbcb

输出：

6

说明：

这个示例中，字符串 *"bcbcbcb"* 本身就是最长的，因为 *'l'*、*'o'*、*'x'* 都出现了 0 次。

```
import sys
from collections import Counter

st = input()
merge_st = st + st
n = len(st)
counter = Counter(st)
if (
    counter.get("l", 0) % 2 == 0
    and counter.get("o", 0) % 2 == 0
    and counter.get("x", 0) % 2 == 0
):
    print(n)
    sys.exit()
flag = False
for i in reversed(range(1, n + 1)):
    for j in range(n):
        tmp_c = Counter(merge_st[j : j + i])
        if (
            tmp_c.get("l", 0) % 2 == 0
            and tmp_c.get("o", 0) % 2 == 0
            and tmp_c.get("x", 0) % 2 == 0
        ):
            ret = i
            flag = True
            break
```

```
    if flag:
        break
print(ret)
```