

题目描述：

给定一个小写字母组成的字符串  $s$ ，请找出字符串中两个不同位置的字符作为分割点，使得字符串分成的三个连续子串且子串权重相等，注意子串不包含分割点。

若能找到满足条件的两个分割点，请输出这两个分割点在字符串中的位置下标，若不能找到满足条件的分割点请返回 0,0。

子串权重计算方式为：子串所有字符的 ASCII 码数值之和。

输入描述：

输入为一个字符串，字符串由  $a \sim z$ ，26 个小写字符组成， $5 \leq \text{字符串长度} \leq 200$ 。

输出描述：

输出为两个分割点在字符串中的位置下标，以逗号分隔

补充说明：

只考虑唯一解，不存在一个输入多种输出解的情况

示例 1

输入：

acdbbbca

输出：

2,5

说明：

以位置 2 和 5 作为分割点，将字符串分割为 ac，bb，ca 三个子串，每一个的子串权重都为 196，输出为：2,5

示例 2

输入：

abcabc

输出：

0,0

说明：

找不到符合条件的分割点，输出为 0,0

#coding=utf-8

# 本题为考试单行多行输入输出规范示例，无需提交，不计分。

import sys

```
def points(a):
```

```
    p1 = 0
```

```
    p2 = len(a) - 1
```

```
    str1_score = 0
```

```
    str2_score = 0
```

```
    while p1 < p2:
```

```
        str1_score += ord(a[p1])
```

```
        str2_score += ord(a[p2])
```

```

        if str1_score == str2_score:
            str3_score = 0
            if p1+1 < p2 - 1:
                tmp_str = a[p1+2:p2-1]
                for i in tmp_str:
                    str3_score += ord(i)

            if str3_score == str2_score:
                return p1+1, p2 - 1
            p1 += 1
            p2 -= 1
        else:
            p1 += 1
            p2 -= 1
    return 0,0

```

```

for line in sys.stdin:
    a = line
a = a[:-1]
result1, result2 = points(a)
print(str(result1) + ',' + str(result2))

```