

## 找车位

题目描述：停车场有一横排车位，**0** 代表没有停车，**1** 代表有车。至少停了一辆车在车位上，也至少有一个空位没有停车。

为了防剐蹭，需为停车人找到一个车位，使得距停车人的车最近的车辆的距离是最大的，返回此时的最大距离。

输入描述：**1**、一个用半角逗号分割的停车标识字符串，停车标识为 **0** 或 **1**，**0** 为空位，**1** 为已停车。

**2**、停车位最多 **100** 个。

输出描述：输出一个整数记录最大距离。

### 示例 1

输入：

1,0,0,0,0,1,0,0,1,0,1

输出：

2

说明：

当车停在第 **3** 个位置上时，离其最近的的车距离为 **2**（**1** 到 **3**）。

当车停在第 **4** 个位置上时，离其最近的的车距离为 **2**（**4** 到 **6**）。

其他位置距离为 **1**。

因此最大距离为 **2**。

```
import sys
```

```
line = sys.stdin.readline().strip('\n')
```

```
p = line.split(',')
```

```
count_zeros = 0
```

```
max_zeros = 0
```

```
head_zeros = None
```

```
for i, v in enumerate(p):
```

```
    if v == '0':
```

```
count_zeros += 1
```

```
elif v == '1':
```

```
if head_zeros is None:
```

```
head_zeros = count_zeros
```

```
max_zeros = max(max_zeros, count_zeros)
```

```
count_zeros = 0
```

```
end_zeros = count_zeros
```

```
side_zeros = max(head_zeros, end_zeros)
```

```
print(max((max_zeros + 1) // 2, side_zeros))
```