

#### 题目描述：

停车场有一横停车位， $0$  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。

#### 补充说明：

##### 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。

```
const rl = require("readline").createInterface({ input: process.stdin });
var iter = rl[Symbol.asyncIterator]();
const readline = async () => (await iter.next()).value;
```

```
void async function () {
    while(line=await readline()){
        //就是找出最长的连续 0 序列
        let a=line.split(',');
        //n 是字符串长度,拆成数组
        let n=a.length;
        let ans=0;
```

```

//遍历数组,找有车的位置
for(let i=0;i<n;i++){
    if(a[i]=='1') continue;
    let mi=n;
    for(let j=i-1;j>=0;j--){
        if(a[j]=='1'){
            mi=Math.min(mi,i-j);
            break;
        }
    }
    for(let j=i+1;j<n;j++){
        if(a[j]=='1'){
            mi=Math.min(mi,j-i);
        }
    }
    ans=Math.max(ans,mi);
}
console.log(ans);
}
// Write your code here
while(line = await readline()){
    let tokens = line.split(' ');
    let a = parseInt(tokens[0]);
    let b = parseInt(tokens[1]);
    console.log(a + b);
}
}()

```