

#### 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。

#### 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  $2$ 。

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
```

```
#define MAX 101
```

```
int main()
{
    int n = 0, i, j, sum = 0, max = 0, flag = 0;
    int num[MAX];
    do{
        scanf("%d", &num[n++]);
```

```

}while(getchar() != '\n');

for(i=0; i<n; i++)
{
    if(num[i] == 1)
    {
        sum = 0;
        flag = 1;
    }
    else sum++;

    if(flag == 0)
        if(max < sum)
            max = sum;
    if(flag == 1)
        if(max < (sum+1)/2)
            max = (sum+1)/2;
    //printf("%d ",sum);
    //printf("%d ",max);
}

flag = 0;
sum = 0;
for(i=n-1; i>0; i--)
{
    if(num[i] == 1)
    {
        sum = 0;
        flag = 1;
    }
    else sum++;

    if(flag == 0)
        if(max < sum)
            max = sum;
    if(flag == 1)
        if(max < (sum+1)/2)
            max = (sum+1)/2;
    //printf("%d ",sum);
    //printf("%d ",max);
}

printf("%d",max);

```

```
    return 0;  
}
```