

题目描述

特定大小的停车场，数组 `cars[]` 表示，其中1表示有车，0表示没车。

车辆大小不一，**小车** 占一个车位（长度1），**货车**占两个车位（长度2），**卡车**占三个车位（长度3）。

统计停车场最少可以停多少辆车，返回具体的数目。

输入描述

整型 **字符串数组** `cars[]`，其中1表示有车，0表示没车，数组长度小于1000。

输出描述

整型数字字符串，表示最少停车数目。

用例

输入	1,0,1
输出	2
说明	1个小车占第1个车位 第二个车位空 1个小车占第3个车位 最少有两辆车

输入	1,1,0,0,1,1,1,0,1
输出	3
说明	1个货车占第1、2个车位 第3、4个车位空 1个卡车占第5、6、7个车位 第8个车位空 1个小车占第9个车位 最少3辆车

题目解析

这道题的意思应该是：给定了车位占用情况，如 1,1,0,0,1,1,1,0,1，这种车位占用情况，可能停了6辆车，即每个1都停了一个小车，这是最多的情况，但是现在要求最少可能停几辆车。

解题思路也很简单，先把卡车，即111的停车情况先弄出来，再将货车，即11的停车情况弄出来，最后再弄小车1的情况。

Java算法源码

```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         String str =
8             sc.nextLine()
9                 .replaceAll(",", "")
10                .replaceAll("111", "x")
11                .replaceAll("11", "x")
12                .replaceAll("1", "x");
13
14         int ans = 0;
15         for (int i = 0; i < str.length(); i++) {
16             if (str.charAt(i) == 'x') {
17                 ans++;
18             }
19         }
20
21         System.out.println(ans);
22     }
23 }
```

JS算法源码

```
1  /* JavaScript Node ACM模式 控制台输入获取 */
2  const readline = require("readline");
3
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout,
7  });
8
9  rl.on("line", (line) => {
10    let str = line
11      .replaceAll(",", "")
12      .replaceAll("111", "x")
13      .replaceAll("11", "x")
14      .replaceAll("1", "x");
15
16    let count = 0;
17
18    while (true) {
19      let index = str.indexOf("x");
20      if (index !== -1) {
21        count++;
22        str = str.slice(index + 1);
23      } else {
24        break;
25      }
26    }
27
28    console.log(count);
29  });
```

Python算法源码

```
1  s = input().replace(",", "").replace("111", "x").replace("11", "x").replace("1", "x")
2
3  ans = 0
4  for c in s:
5      if c == 'x':
6          ans += 1
7
8  print(ans)
```