

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。

补充说明：

示例 1

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

示例 2

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

```
#include <iostream>
#include <sstream>
#include <string>
#include <vector>
using namespace std;
int main() {
    string input;
    getline(cin, input);
    vector<int> vec;
    while(input.find(",") != string::npos){
        int val = input.find(",");
        string str = input.substr(0, val);
        vec.emplace_back(atoi(str.c_str()));
        input = input.substr(val + 1);
    }
    vec.emplace_back(atoi(input.c_str()));
    vector<int> cars;
    int n = 0;
    for(auto & i : vec){
        if(i == 0 ){
            if(n != 0){
                cars.push_back(n);
            }
            n = 0;
            continue;
        }
        n++;
    }
    if(n != 0){
        cars.push_back(n);
    }
    int count = 0;
    for(auto it : cars){
        int m = it%3;
        if(m != 0){
            it -= m;
            count++;
        }
        count += it/3;
    }
}
```

```
    cout<<count;
    return 0;
}
```