

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。

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 <vector>
#include <string>
#include <algorithm>
#include <unordered_map>
#include <unordered_set>
#include <set>
#include <queue>
#include <map>
#include <deque>
#include <sstream>
#include <cstring>
#include <functional>

using namespace std;

string getInput(string line) {
    line.erase(remove(line.begin(), line.end(), ','), line.end());
    return line;
}

int getLessCar(string parking) {
    int n = parking.size();
    int count = 0;
    for(int i = 0; i < n; i++) {
        if(parking[i] == '1') {
            // 找到连续的 1
            int j = i;
            while(j < n && parking[j] == '1') j++;
        }
    }
    return count;
}
```

```

        // 此时 j 指向末尾或者是 0
        int len = j - i;
        // 卡车 3、货车 2、汽车 1
        if(len <= 3) count++;
        else count += (len + 2) / 3;
        i = j - 1;
    }
}
return count;
}

int main()
{
    string parking;
    getline(cin, parking);

    parking = getInput(parking);

    cout << getLessCar(parking) << endl;

    return 0;
}

```