

题目描述：

游乐场里增加了一批摇摇车，非常受小朋友欢迎，但是每辆摇摇车同时只能有一个小朋友使用，如果没有空余的摇摇车，需要排队等候，或者直接离开，最后没有玩上的小朋友会非常不开心。请根据今天小朋友的来去情况，统计不开心的小朋友数量。

- 1、摇摇车数量为 N ，范围是： $1 \leq N < 10$ ；
- 2、每个小朋友都对应一个编码，编码是不重复的数字，今天小朋友的来去情况，可以使用编码表示为： $1\ 1\ 2\ 3\ 2\ 3$ 。（若小朋友离去之前有空闲的摇摇车，则代表玩耍后离开；不考虑小朋友多次玩的情况）。小朋友数量 ≤ 100
- 3、题目保证所有输入数据无异常且范围满足上述说明。

输入描述：

第一行：摇摇车数量
第二行：小朋友来去情况

输出描述：

返回不开心的小朋友数量

示例 1

输入：

1
1 2 1 2

输出：

0

说明：

第一行，1 个摇摇车

第二行，1号来 2号来（排队） 1号走 2号走（1号走后摇摇车已有空闲，所以玩后离开）

示例 2

输入：

1
1 2 2 3 1 3

输出：

1

说明：

第一行，1个摇摇车

第二行，1号来 2号来（排队） 2号走（不开心离开） 3号来（排队） 1号走 3号走
（1号走后摇摇车已有空闲，所以玩后离开）

```
import java.util.Arrays;
import java.util.LinkedList;
import java.util.List;
import java.util.Scanner;
import java.util.stream.Collectors;
```

```
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int num = Integer.parseInt(in.nextLine()); // 摇摇车数量
        List<Integer> list = Arrays.stream(in.nextLine().split(" ")).
            map(Integer::new).collect(Collectors.toList());
        LinkedList<Integer> carStack = new LinkedList<>();
        LinkedList<Integer> queue = new LinkedList<>();
        int rs = 0; // 不开心的数量
        for (int i = 0; i < list.size(); i++) {
            Integer child = list.get(i); // 小朋友

            // 小朋友第一次出现
            if (!carStack.contains(child) && !queue.contains(child)) {
                if (carStack.size() < num) {
                    carStack.add(child);
                } else {
                    queue.add(child);
                }
            }
        }
    }
}
```

```

        }else if(carStack.contains(child)){
            //小朋友在摇摇车中
            //从摇摇车中移除,若队列中存在别的小朋友,将其加入到摇摇车中,并从排
            队中移除

            carStack.remove(child);
            if(queue.size()!=0){
                Integer queueChild = queue.removeFirst();
                carStack.add(queueChild);
            }
        }else if (queue.contains(child)){
            //小朋友在排队
            //直接从排队中移除,并使结果+1
            queue.remove(child);
            rs++;
        }
    }
    System.out.println(rs);
}
}

```