

# 2019 CSP-S模拟赛Day2

题目名称	permutation	zuma	chessboard
输入文件	permutation.in	zuma.in	chessboard.in
输出文件	permutation.out	zuma.out	chessboard.out
空间限制	128MB	128MB	512MB
时间限制	1s	1s	4s
测试点数	10	10	10
题目类型	传统型	传统型	传统型
评测方式	全文比较	全文比较	全文比较

1. 考试时长为3.5个小时。
2. 评测环境为常用评测环境，评测时开启O2优化。
3. 全文比较时忽略多余空格、换行符和制表符。

# 1 *permutation*

## 1.1 Description

已知关于n的排列是有顺序的n个数，每个数为1到n且1到n中每个数出现且仅出现一次

现在，小T把一些排列混在了一起（也可能混入了一些别的数），他想让你把排列全部划分开

小T给了你一个n个数的序列，你需要判断这些数是否是若干个排列组成的，如果是，你需要先输出排列个数，再输出每个数属于哪个排列

为了加大难度，在输出每个数属于哪个排列，小T想让你用最小字典序输出。字典序的比较，即先比较第一个数属于排列的大小，再比较第二个数.....

## 1.2 Input

第一行一个数n，表示有n个数

接下来一行，共n个数,描述这个序列

## 1.3 Output

如果是由若干个排列组成,第一行输出排列的个数,第二行n个数,表示每个数属于哪个排列(注意要求最小字典序).否则输出-1

## 1.4 Sample Input

```
9
1 2 3 1 2 1 4 2 5
```

## 1.5 Sample Output

```
3
1 1 1 2 2 3 1 3 1
```

## 1.6 Sample Input

```
4
1 2 2 3
```

## 1.7 Sample Output

```
-1
```

## 1.8 Data

30 %,  $n \leq 10$

60 %,  $n \leq 1000$

100 %,  $n \leq 100000$ , 序列中数  $\leq n$

## 2 *zuma*

### 2.1 Description

小P有一行N个弹子，每一个都有一个颜色。每次可以让 $\geq K$ 个连续的同颜色的一段弹子消失，剩下的会重新紧凑在一起。现在小P有无限的所有颜色的弹子，他想在这行弹子中插入最少的弹子，使得弹子全部消失。

但他想了很久，也没想清最少需要插入多少个弹子，你能告诉他吗？

注意，对于 $\geq K$ 个连续的同颜色的一段弹子，小p可以让他先不消失

### 2.2 Input

第一行两个数n, K，分别表示弹子总个数和连续段消失至少要的弹子数

接下来一行，共n个正整数，分别表示每个弹子的颜色

### 2.3 Output

一个数，即最少需要插入的弹子数

### 2.4 Sample Input

```
5 3
2 2 3 2 2
```

### 2.5 Sample Output

```
2
```

### 2.6 Sample Input

```
10 4
3 3 3 3 2 3 1 1 1 3
```

## 2.7 Sample Output

4
---

## 2.8 Data

20 %,  $n \leq 10, K \leq 5$

50 %,  $n \leq 50, K \leq 5$

100 %,  $n \leq 100, K \leq 5$ , 颜色的值  $\leq 100$

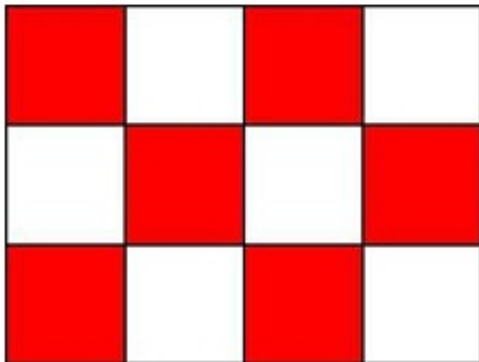
注意：数据有一定梯度，尽量优化自己的算法，同时，颜色的值可能比 $n$ 大

## 3 *chessboard*

### 3.1 Description

给出一个 $n \times n$ 的棋盘，每个格子有一种颜色，总共有 $K$ 种颜色，用整数 $1 \dots K$ 表示。现有三个操作：

①PAINT 给出五个正整数 $c, x_1, y_1, x_2, y_2$ ，将 $(x_1, y_1)$ 为左上角， $(x_2, y_2)$ 为右下角的矩形间隔染色，其中 $(x_1, y_1)$ 被染成颜色 $c$ ，与它相邻的格子不染色，具体如下图：



②SAVE 将当前棋盘的情况存储下来

③LOAD 给出一个正整数，将第 $x$ 次存储的棋盘载入，用于进行后面的操作

一开始棋盘上的格子都是1号颜色，给出 $M$ 条指令，每条指令是上面的一种。要求输出执行完所有指令后每个格子的颜色

### 3.2 Input

第一行三个正整数 $n, K, M$ ，分别表示棋盘大小，颜色总数和指令数

接下来 $M$ 行，表示每个指令，注意棋盘从0开始标号

### 3.3 Output

### 3.4 Sample Input

```
4 3 2
PAINT 2 0 0 3 3
PAINT 3 0 3 3 3
```

### 3.5 Sample Output

```
2 1 2 3
1 2 1 2
2 1 2 3
1 2 1 2
```

### 3.6 Sample Input

```
3 3 4
PAINT 3 0 0 1 1
SAVE
PAINT 2 1 1 2 2
LOAD 1
```

### 3.7 Sample Output

```
3 1 1
1 3 1
1 1 1
```

### 3.8 Sample Input

```
3 4 7
PAINT 2 0 0 1 1
SAVE
PAINT 3 1 1 2 2
SAVE
PAINT 4 0 2 0 2
```

```
LOAD 2
PAINT 4 2 0 2 0
```

### 3.9 Sample Output

```
2 1 1
1 3 1
4 1 3
```

### 3.10 Data

50 %,  $n \leq 50$ ,  $K \leq 100000$ ,  $M \leq 1000$ , 其中20 %的数据只有操作1

另有15 %的数据  $n \leq 100$ ,  $K \leq 100000$ ,  $M \leq 3500$

100 %,  $n \leq 1000$ ,  $K \leq 100000$ ,  $M \leq 100000$