

# NOIP 模拟赛

题目名称	购票方案	直线相交	记录数列	模式匹配
目录	buy	intersect	record	select
可执行文件名	buy	intersect	record	select
输入文件名	buy.in	intersect.in	record.in	select
输出文件名	buy.out	intersect.out	record.out	select
每个测试点时限	1.0s	1.0s	1.0s	2.0s
内存限制	512MB	512MB	512MB	512MB
试题总分	100	100	100	100
测试点数目	20	20	20	20
每个测试点分值	5	5	5	5
是否有部分分	否	否	否	否
题目类型	传统型	传统型	传统型	传统型

提交的源程序文件名:

对于 C++ 语言	buy.cpp	intersect.cpp	record.cpp	select.cpp
对于 C 语言	buy.c	intersect.c	record.c	select.c
对于 Pascal 语言	buy.pas	intersect.pas	record.pas	select.pas

编译开关:

对于 C++ 语言	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11
对于 C 语言	-O2 -std=c11	-O2 -std=c11	-O2 -std=c11	-O2 -std=c11
对于 Pascal 语言	-O2	-O2	-O2	-O2

# 1 购票方案 (buy)

## 1.1 题目描述

小 D 家附近新开了一家游乐园。小 D 经常光顾这家游乐园，他想要找到最优的购票方案。

小 D 想要进入游乐园  $N$  次，第  $i$  次在时刻  $T_i$ ，这个游乐园有两种购票方式：

1. 购买单次票：价格为  $One$ ，该票每张只能进入游乐园一次。

2. 购买时限票：时限票一共有  $K$  种类型，第  $i$  种类型的时限票可以使用的时长为  $num_i$ ，价格为  $cost_i$ 。具体来说，若一张时限票在第  $t$  时刻被使用，则在  $[t, t + num_i - 1]$  这个时刻区间内小 D 都可以进入游乐园。

这两种票都可以购买任意多张。记  $sum_i$  为小 D 前  $i$  次进入游乐园的最小花费（不考虑第  $i$  次之后的进入），小 D 定义第  $i$  次进入游乐园的代价为  $spend_i = sum_i - sum_{i-1}$ 。小 D 要知道每次进入游乐园的代价，即  $spend_1, \dots, spend_n$ 。

## 1.2 输入格式

从文件 buy.in 中读取数据。

第一行两个非负整数  $N, K$ ，表示进入游乐园的总次数和时限票的种类数。

第二行  $N$  个正整数  $T_1, \dots, T_N$ ，表示小 D 每次进入游乐园的时刻，保证单调增。

第三行到第  $K + 2$  行，每行两个整数  $num_i, cost_i$  表示每种时限票的时长和价格。

第  $K + 3$  一个整数  $One$ ，表示单次票的价格。

## 1.3 输出格式

输出到文件 buy.out 中。

$N$  行，每行一个整数  $Spend_i$ ，表示第  $i$  次进入游乐园的代价。

## 1.4 样例输入 1

```
3 2
10 20 30
90 50
1440 120
20
```

## 1.5 样例输出 1

```
20
```

20  
10

## 1.6 样例解释

对于前 1 次进入，购买单次票，总价格为 20，故第 1 次进入的代价为  $20 - 0 = 20$ ；

对于前 2 次进入，购买两张单次票，总价格为 40，故第 2 次进入的代价为  $40 - 20 = 20$ ；

对于前 3 次进入，购买一张类型 1 时限票，总价格为 50，故第 3 次进入的代价为  $50 - 40 = 10$ 。

## 1.7 样例输入 2

10 2  
13 45 46 60 103 115 126 150 256 516  
90 45  
450 100  
20

## 1.8 样例输出 2

20  
20  
5  
0  
20  
0  
0  
20  
15  
20

## 1.9 数据范围和约定

对于前 10% 的数据， $K = 0$ ；

对于前 40% 的数据， $K \leq 2, N \leq 10^3$ ；

对于前 60% 的数据， $K \leq 2, N \leq 10^5$ ；

对于前 80% 的数据， $K \leq 30, N \leq 10^5$ ；

对于 100% 的数据， $K \leq 500, N \leq 10^5, 1 \leq T_i, num_i, cost_i, One \leq 10^9, T_i$  单调增。

## 1.10 提示

注意虽然  $Spend_i$  不会超过 `int` 范围，但是  $Sum_i$  可能会超出 `int` 范围。

## 2 直线相交 (intersect)

### 2.1 题目描述

游乐园内有这样一个项目：扔出  $N$  根木棒，猜这些木棒之间会有多少个交点。

形式化地，我们可以假设木棒的长度近似为无穷，木棒可以互相平行，但是不存在木棒相互重合或者三条及以上木棒交于同一点的情况。

小 D 想知道，如果扔出  $N$  根木棒，是否有可能出现总交点数为  $M$  的情况。

本题多组询问。

### 2.2 输入格式

从文件 `intersect.in` 中读取数据。

第一行一个整数  $Q$ ，表示询问的数量。

第二行到第  $Q + 1$  行，每行两个整数  $N, M$  表示一组询问：在扔出  $N$  根木棒的情况下是否有可能出现交点数为  $M$  的情况。

### 2.3 输出格式

输出到文件 `intersect.out` 中。

$Q$  行，每行一个整数 0 或者 1：1 表示有可能出现，0 表示不可能出现。

### 2.4 样例输入 1

```
4
3 0
3 1
3 3
4 4
```

### 2.5 样例输出 1

```
1
0
1
1
```

## 2.6 样例解释

下图分别是 3 根木棒摆出 0、3 个交点以及 4 根木棒摆出 4 个交点的情况。



## 2.7 样例输入 2

```
8
40 99
41 100
42 782
43 777
44 888
45 1000
46 1212
47 2333
```

## 2.8 样例输出 2

```
0
0
1
1
1
0
0
0
```

## 2.9 数据范围与约定

对于前 20% 的数据,  $N \leq 5, M \leq 20$ ;

对于前 50% 的数据,  $N \leq 10, M \leq 100$ ;

对于前 80% 的数据,  $N \leq 50, M \leq 2500, Q \leq 10$ ;

对于 100% 的数据,  $N \leq 500, 0 \leq M \leq 10^5, Q \leq 10^5$ 。

## 3 记录数列 (record)

### 3.1 题目描述

对于一个  $1 - N$  的排列  $p$ ，小 D 如下定义长度为  $N$  的阶梯数组  $a$ ：

$a_i$  为包含下标  $i$  的由单调且连续的元素构成的区间的最大长度，即在  $p$  中这个区间要么是  $[x, x+1, x+2, \dots, y-1, y]$  或者是  $[y, y-1, \dots, x+1, x]$  的形式。

例如，对于排列  $p = [4, 1, 2, 3, 7, 6, 5]$ ，我们有  $a = [1, 3, 3, 3, 3, 3, 3]$ 。

现在，小 D 已经知道了阶梯数组  $a$ ，他想知道有多少个排列的阶梯数组等于  $a$ ，由于答案可能很大，小 D 只想知道答案对 998244353 取模后的结果。

### 3.2 输入格式

从文件 record.in 中读取数据。

第一行一个整数  $N$ ，表示阶梯数组的长度。

第二行  $N$  个正整数  $a_1, \dots, a_N$ ，描述阶梯数组。

### 3.3 输出格式

输出到文件 record.out 中。

一行一个整数  $Ans$ ，表示排列的数量对 998244353 取模后的结果。

### 3.4 样例输入 1

```
6
3 3 3 1 1 1
```

### 3.5 样例输出 1

```
6
```

### 3.6 样例解释

可能的排列  $p$  如下：

```
2 3 4 6 1 5
3 4 5 1 6 2
3 4 5 2 6 1
```

```
4 3 2 6 1 5
4 3 2 5 1 6
5 4 3 1 6 2
```

### 3.7 样例输入 2

```
8
2 2 2 2 2 2 1 1
```

### 3.8 样例输出 2

```
370
```

### 3.9 数据范围与约定

对于前 20% 的数据,  $N \leq 8$ ;

对于前 50% 的数据,  $N \leq 16$ ;

对于另外 10% 的数据, 保证  $a_i = 1$ ;

对于另外 20% 的数据, 保证  $a_i > 1$ ;

对于 100% 的数据,  $N \leq 2000$ ,  $1 \leq a_i \leq N$ 。



## 4 模式匹配 (select)

### 4.1 题目描述

小 D 最近迷上了 'abab' 类型的四元组，即第一个位置和第三个位置相同，第二个位置和第四个位置相同。

给定一个数列  $a$ ，小 D 想要找到它的一个子序列  $b$ ，满足：

1. 序列  $b$  的长度  $m$  为 4 的倍数；

2. 对于任意  $0 \leq k < m/4$ ，有  $b_{4k+1} = b_{4k+3}$  且  $b_{4k+2} = b_{4k+4}$ 。（注意  $b_{4k+1}$  可以与  $b_{4k+2}$  相同也可以不相同。）

现在小 D 想要知道数列  $b$  的长度最长可以是多少。

### 4.2 输入格式

从文件 `select.in` 中读取数据。

第一行一个整数  $N$ ，表示数列  $a$  的长度。

第二行  $N$  个整数  $a_1, \dots, a_N$ ，描述数列  $a$ 。

### 4.3 输出格式

输出到文件 `select.out` 中。

一行一个整数  $Ans$ ，表示数列  $b$  的最大长度。

### 4.4 样例输入 1

```
10
3 1 2 1 2 3 4 5 4 5
```

### 4.5 样例输出 1

```
8
```

### 4.6 样例解释

序列  $b$  可以为：1, 2, 1, 2, 4, 5, 4, 5，长度为 8。

## 4.7 数据范围与约定

对于前 10% 的数据,  $N \leq 100$ ;

对于前 30% 的数据,  $N \leq 3000$ ;

对于另外 20% 的数据, 保证每个  $a_i$  最多出现两次。

对于前 80% 的数据,  $N \leq 10^5$ ;

对于 100% 的数据,  $N \leq 5 \times 10^5$ ,  $1 \leq a_i \leq N$ 。