



## 骰码提高组10月模拟赛

### 提高级

题目名称	ZOM	骰子	忍俊不禁	追逐极光者
题目类型	传统型	传统型	传统型	传统型
目录	zom	dice	laugh	aurora
可执行文件名	zom	dice	laugh	aurora
输入文件名	zom.in	dice.in	laugh.in	aurora.in
输出文件名	zom.out	dice.out	laugh.out	aurora.out
每个测试点时限	2000ms	1000ms	1000ms	1000ms
内存限制	1024mb	256mb	256mb	256mb
子任务数目	20	20	10	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于C++语言	zom.cpp	dice.cpp	laugh.cpp	aurora.cpp
---------	---------	----------	-----------	------------

编译选项

对于C++语言	-O2 -lm
---------	---------

### 注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++中函数main()的返回值类型必须是int，程序正常结束时的返回值必须是0。
3. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
4. 程序可使用的栈空间内存限制与题目的内存限制一致。
5. 只提供Linux格式附加样例文件。

# ZOM (zom)

## 题目背景

居住在 ZOM 国度的小 S 已经完成了种树的基本目标，现在他要对他的树进行一些修改。

## 题目描述

给定一棵  $n$  个点的树和  $q$  次操作，1 为树的根，分为两种：

1. 给定  $u, v$ ，将  $u$  的父亲修改为  $v$ ，保证  $u$  是叶子结点。
2. 给定  $u, v$ ，查询  $u$  和  $v$  的最近公共祖先。

## 输入格式

第一行输入两个正整数  $n, q$ 。

第  $2 \sim n$  行，每行输入两个正整数  $u, v$ ，表示一条树边。

第  $n + 1 \sim n + q$  行，每行三个正整数  $op, u, v$ 。其中  $op$  表示操作种类， $u, v$  含义如题面所述。

## 输出格式

对于每一个查询操作，输出一行一个整数，表示你的答案。

## 样例 #1

### 样例输入 #1

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

### 样例输出 #1

```
1
2
```

## 提示

对于 30% 的数据，保证  $1 \leq n, q \leq 1000$

对于 60% 的数据，保证  $1 \leq n, q \leq 2 \times 10^5$

对于另外 10% 的数据，保证不存在操作 1。

对于 100% 的数据，保证  $1 \leq n, q \leq 10^6$ ，所有输入均合法。

# 骰子 (dice)

## 题目描述

小 C 喜欢骰子，尽管这些骰子非常奇怪。

小 C 的手中有  $n$  个不同的骰子，其中第  $i$  个骰子是  $a_i$  面骰子，每个面上的数字依次为  $1, 2, 3, \dots, a_i$ 。

现在，小 C 要选择其中的  $k$  个骰子，并对于这  $k$  个骰子中的每一个骰子，选择一个面朝上。

小 C 想要知道，共有多少种不同的选择骰子和选择朝上的面的方案，可以使所有选择的骰子的朝上的面上的数字之和不大于  $p$ 。

两种选择骰子和选择朝上的面的方案不同，当且仅当**存在一个只在其中一种选择方案中出现的骰子或存在一个同时出现在这两种选择方案的骰子的朝上的面上的数字不同**。

由于答案可能很大，所以你只需要输出答案对 998244353 取模的结果。

## 输入格式

第一行三个整数  $n, k, p$ 。

第二行  $n$  个整数，表示数组  $a$ 。

## 输出格式

一行一个整数，表示满足条件的不同的选择骰子和选择朝上的面的方案数对 998244353 取模的结果。

## 样例 #1

### 样例输入 #1

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

### 样例输出 #1

```
28
```

## 样例 #2

### 样例输入 #2

```
4 4 13
6 4 5 3
```

### 样例输出 #2

```
296
```

# 提示

## 【样例 #3】

见附加文件中的 `dice/dice3.in` 与 `dice/dice3.ans`。

该样例满足测试点 8 的限制。

## 【样例 #4】

见附加文件中的 `dice/dice4.in` 与 `dice/dice4.ans`。

该样例满足测试点 11 的限制。

## 【样例 #5】

见附加文件中的 `dice/dice5.in` 与 `dice/dice5.ans`。

该样例满足测试点 13 的限制。

## 【样例 #6】

见附加文件中的 `dice/dice6.in` 与 `dice/dice6.ans`。

该样例满足测试点 20 的限制。

## 【数据范围】

对于 100% 的数据,  $1 \leq k \leq n \leq 200$ ,  $1 \leq p \leq 1000$ ,  $1 \leq a_i \leq 10^9$ 。

测试点编号	$n \leq$	$p \leq$	$a_i \leq$	特殊性质
1	8	8	8	是
2 ~ 3	8	1000	8	是
4 ~ 5	8	12	$10^9$	否
6 ~ 7	50	80	80	否
8 ~ 10	50	80	$10^9$	否
11 ~ 12	200	1000	1	否
13 ~ 15	200	400	$10^9$	是
16 ~ 20	200	1000	$10^9$	否

特殊性质: 保证  $k = n$ 。

# 忍俊不禁 (laugh)

## 题目背景

忍俊不禁题。会做之后，你会笑出声来。

## 题目描述

给定  $A_1$ ，求一个序列  $A$  的通项。

序列  $A$  满足  $A_n = qA_{n-1} + \sum_{i=0}^m n^i x_i$ 。

其中  $q$  和序列  $x$  都是给定的。

如果输出  $A$  的通项见输出格式。

## 输入格式

第一行三个整数  $m, A_1, q$ ，含义见题面。

第二行  $m + 1$  个整数，用空格隔开，第  $i$  个整数代表  $x_{i-1}$ 。

## 输出格式

我们保证  $A$  的通项为  $A_n = t \times q^n + \sum_{i=0}^m y_i n^i$  的格式。且所有输出的数字都保证为有理数，要求输出的结果对于 998244353 取模。

第一行输出一个正整数  $t$ ，含义见上段。

第二行输出  $m + 1$  个正整数，用空格隔开，第  $i$  个数代表  $y_{i-1}$ 。

## 样例 #1

### 样例输入 #1

```
0 3 3
0
```

### 样例输出 #1

```
1
0
```

### 样例输入 #2

```
见下发文件中的 ex_laugh1.in。
```

## 样例输出 #2

见下发文件中的 `ex_laugh1.ans`。

## 提示

---

对于 10% 的数据，保证  $m = 0$ 。

对于 30% 的数据，保证  $m \leq 1$ 。

对于 100% 的数据，保证  $1 \leq m \leq 300$ ,  $0 \leq a_1, q, x_i (0 \leq i \leq m) < 998244353$ 。

数据有一定的梯度。

# 追逐极光者 (aurora)

## 题目背景

「这人真有意思，明明就在极光城里，却在寻找极光城。」

「不……他从未抵达极光城……」

从此以后，极光城里多了个找极光城的疯子。

## 题目描述

极光城形如一颗  $n$  个点的树，每个点最多只连接了 10 条双向的街道。

「你在找什么？」

「我在找极光城。」

「这里就是极光城。」

「我快找到了……我一定能找到的！」

同时，有  $m$  种风筝在极光城上空飘荡，在  $u_i$  到  $v_i$  间飘荡着  $w_i$  只第  $i$  种风筝。

Donald 想要顺着风筝找极光城，他可以在极光城每条街道停留并追寻风筝，且 Donald 在每条街道只能追寻其中一种风筝，如果他完整的经过了第  $i$  种风筝飘荡的路线，则 he 可以获得这  $w_i$  只风筝。

Donald 希望获得最多的风筝，他认为这样就能找到极光城了，请「你」告诉他最多能获得多少风筝。

「极光城有风筝……我也有风筝……顺着风筝……我一定能去极光城！」

「我就快找到极光城了……」

「我就快找到了！」

## 形式化题意

给定一颗  $n$  个点的树，树上每个点最大度数为 10，以及树上  $m$  条路径，路径有权值  $w$ ，你需要选出若干路径使得选出的路径边无交，问最大价值。

$$n, m \leq 1000, w_i \leq 10^9$$

## 输入格式

第一行两个整数  $n, m$ ，表示点数以及路径数。

接下来  $n - 1$  行，每行两个整数  $u, v$  表示  $u, v$  之间有条双向边。

接下来  $m$  行，每行三个整数  $u, v, w$  表示  $u, v$  为路径的两个端点， $w$  为这条路径的权值。

## 输出格式

输出一行一个整数，表示最大价值。

# 输入输出样例

## 样例#1

### 样例输入#1

```
7 3
1 2
1 3
2 4
2 5
3 6
5 7
1 6 1
1 4 3
4 7 1
```

### 样例输出#1

```
4
```

## 样例 #2

### 样例输入 #2

```
见下发文件中的 ex_aurora1.in。
```

### 样例输出 #2

```
见下发文件中的 ex_aurora1.ans。
```

## 样例 #3

### 样例输入 #3

```
见下发文件中的 ex_aurora2.in。
```

### 样例输出 #3

```
见下发文件中的 ex_aurora2.ans。
```

## 提示

对于所有数据满足  $n, m \leq 1000, w \leq 10^9$

数据点编号			特殊性质
1~2	10	10	



数据点编号	$n$	$m$	特殊性质
2~4	$10^3$	10	
5~8	100	100	
9~10	$10^3$	$10^3$	保证树是一条链
11~12	$10^3$	$10^3$	保证每个点度数最大为 3
13~20	$10^3$	$10^3$	每个点的度数不超过 10