

# 干饭协会的入场券

## 题目描述

This problem is the ticket of the Eat-More Union.

You are given an array of  $n$  positive integers  $a_1, a_2, \dots, a_n$ .

Please count the number of factors of  $\prod_{i=1}^n a_i$ .

## 输入格式

The first line contains a single integer  $n$  ( $1 \leq n \leq 10^5$ ).

The second line contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^7$ ).

## 输出格式

Print a single integer — the number of factors of  $\prod_{i=1}^n a_i$ . Since the answer can be very large, so print it modulo 998244353.

## 样例

输入：

```
1 | 5
2 | 1 2 3 4 5
```

输出：

```
1 | 16
```

## 提示

$$\prod_{i=1}^n a_i = a_1 \cdot a_2 \cdot a_3 \cdots a_n \quad (1)$$

# 干饭协会的副本侠

## Description

阿水是 EatMore 干饭协会里著名的副本侠，因为阿水可以很轻松的通关各式各样的副本。

今天游戏里出了一个新副本《云梦泽》，副本侠阿水早已跃跃欲试了。



新副本《云梦泽》地图

《云梦泽》里共有  $n$  只怪兽，由于 3202 年的怪兽十分强大，只有使用技能才有机会杀死怪兽。

阿水有两个技能，每个技能需要花费 1 秒的时间来释放，然后该技能会进入为期 1 秒的冷却时间（冷却时不能释放该技能，但可以释放另一个技能）。

每次释放技能可以指定一只怪兽，且阿水的两个技能分别有  $p_1\%$ ,  $p_2\%$  的概率杀死怪兽。

阿水会以 1 秒每次的频率轮流释放两个技能，直至所有怪兽都被杀死（即通关该副本）；在每次释放技能的过程中，阿水会受到所有存活怪兽的单次攻击，**即每只怪兽被杀死前会对阿水造成一次伤害**。

阿水听说你擅长计算，于是问你：在最优情况下，他通关副本所受到怪兽攻击次数的期望是多少？

## Input

本题包含多组测试数据。

第一行有一个整数  $T$  ( $1 \leq T \leq 10000$ )，表示一共有  $T$  组测试数据。

接下来每一行有三个整数，表示新副本中的怪兽数量  $n$  ( $1 \leq n \leq 5 \cdot 10^6$ )，以及阿水两个技能击杀怪兽概率的 100 倍  $p_1, p_2$  ( $0 \leq p_1, p_2 \leq 100$ ,  $p_1 + p_2 > 0$ )。

保证所有测试数据的怪兽数量之和  $\sum n$  不超过  $10^6$ 。

## Output

输出共  $T$  行，对于每组测试数据输出一行一个非负整数，表示最优情况下，阿水受到怪兽攻击次数的期望对 998244353 取模后的结果。

可以证明，所求期望一定是一个有理数，设其为  $\frac{p}{q}$  ( $\gcd(p, q) = 1$ )，那么你输出的非负整数  $x$  要满足  $p \equiv qx \pmod{998244353}$ 。

## Sample #1

## Sample Input #1

```
1 | 2
2 | 1 50 50
3 | 5 0 100
```

## Sample Output #1

```
1 | 2
2 | 25
```

## Hints

对于第一组测试数据，仅有一只怪兽，且阿水每秒都有  $\frac{1}{2}$  的概率杀死怪兽。

因此，阿水会受到怪兽的第 1 次攻击，有  $\frac{1}{2}$  的概率受到第 2 次攻击， $\dots$ ，有  $\frac{1}{2^{n-1}}$  的概率受到第  $n$  次攻击， $\dots$

所以，阿水受到怪兽攻击次数的期望为  $\sum_{i=0}^{\infty} \frac{1}{2^i} = 2$ 。

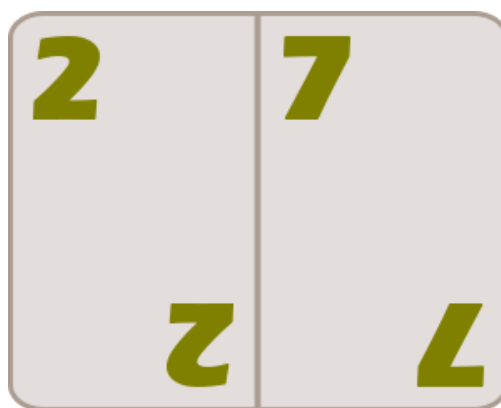
对于第二组测试数据，由于技能 1 不可能杀死怪兽，而技能 2 一定能杀死怪兽，因此阿水在释放完 5 次技能 2 后就能杀死所有怪兽。

阿水可以首先释放技能 2，即在第 1, 3, 5, 7, 9 秒初释放技能 2，在第 2, 4, 6, 8 秒初释放技能 1；于是在 9 秒末就能杀死所有怪兽，9 秒内各秒分别受到 5, 4, 4, 3, 3, 2, 2, 1, 1 次怪兽的攻击，共 25 次攻击。

# 干饭协会的双子牌

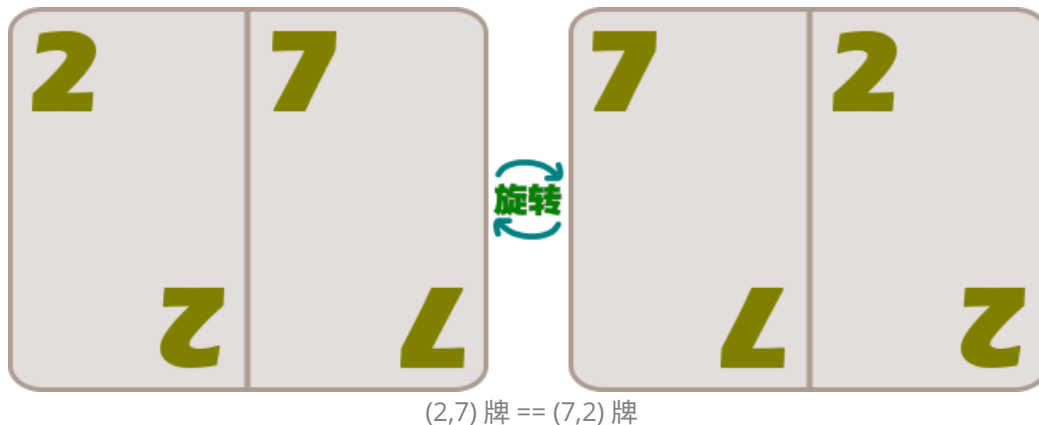
## 题目描述

EatMore 干饭协会发明了一款牌类游戏，会员们都管它叫双子牌，因为每张牌都有左右两个数字，如下图所示是一张 (2, 7) 牌（两个数字分别是 2 和 7）。



(2,7) 双子牌

当然，你可以将这张牌旋转  $180^\circ$ ，使其变成一张 (7, 2) 牌，如下图所示。



现有  $n$  张双子牌排成一行，其中从左数起的第  $i$  张牌是  $(a_i, b_i)$  牌。

你可以进行任意多次如下操作：

- 从  $n$  张牌里选择一张牌  $(a_i, b_i)$  牌，将其旋转  $180^\circ$  变成  $(b_i, a_i)$  牌。

在完成所有操作之后，把这一行双子牌变成一个长度为  $2n$  的数组

$$c = [c_1, c_2, c_3, c_4, \dots, c_{2n}] = [a_1, b_1, a_2, b_2, \dots, a_n, b_n]。$$

请问你能得到多少种相邻元素不相等的数组（即对于任意的  $1 \leq i < 2n$ ，满足  $c_i \neq c_{i+1}$ ）？可以证明，满足要求的不同数组数量有限。由于答案可能很大，输出其对  $10^9 + 7$  取模后的结果。

作为提示，两个数组相等当且仅当两个数组长度相等且相同下标的元素值相等。

## 输入格式

本题包含多组测试数据。

第一行有一个整数  $T$  ( $1 \leq T \leq 10000$ )，表示一共有  $T$  组测试数据。

对于每组测试数据：

- 第一行有 1 个整数  $n$  ( $1 \leq n \leq 10^6$ )，表示共有  $n$  张双子牌。
- 接下来  $n$  行中，第  $i$  行有 2 个整数  $a_i, b_i$  ( $1 \leq a_i, b_i \leq 10^9, 1 \leq i \leq n$ )，表示初始时刻第  $i$  张双子牌为  $(a_i, b_i)$  牌。

保证将所有测试数据的双子牌全部叠在一起后，总张数  $\sum n$  不超过  $10^6$ 。

## 输出格式

输出共  $T$  行，对于每组测试数据输出一行一个整数表示答案（对  $10^9 + 7$  取模）。

## 样例 #1

### 样例输入 #1

```
1 2
2 2
3 3 5
4 5 7
5 2
6 114514 114514
7 114514 233333
```

# 样例输出 #1

1	3
2	0

## 提示

对于样例第一组测试数据，下图列出了所有可能的数组，显然答案为 3。

3

5

5

7

3

5

5

7

✖

3

5

7

5

3

5

7

5

✔

5

3

5

7

5

3

5

7

✔

5

3

7

5

5

3

7

5

✔

第一组测试数据示意图

# 干饭协会的彩虹猫

## Description

EatMore 干饭协会有一只彩虹猫，名叫悠米，悠米从来不用腿走路，因为悠米拥有一本魔典，只要悠米吃够了协会食堂为她特供的彩虹猫粮，悠米就能站在魔典上释放魔法、自由飞行。



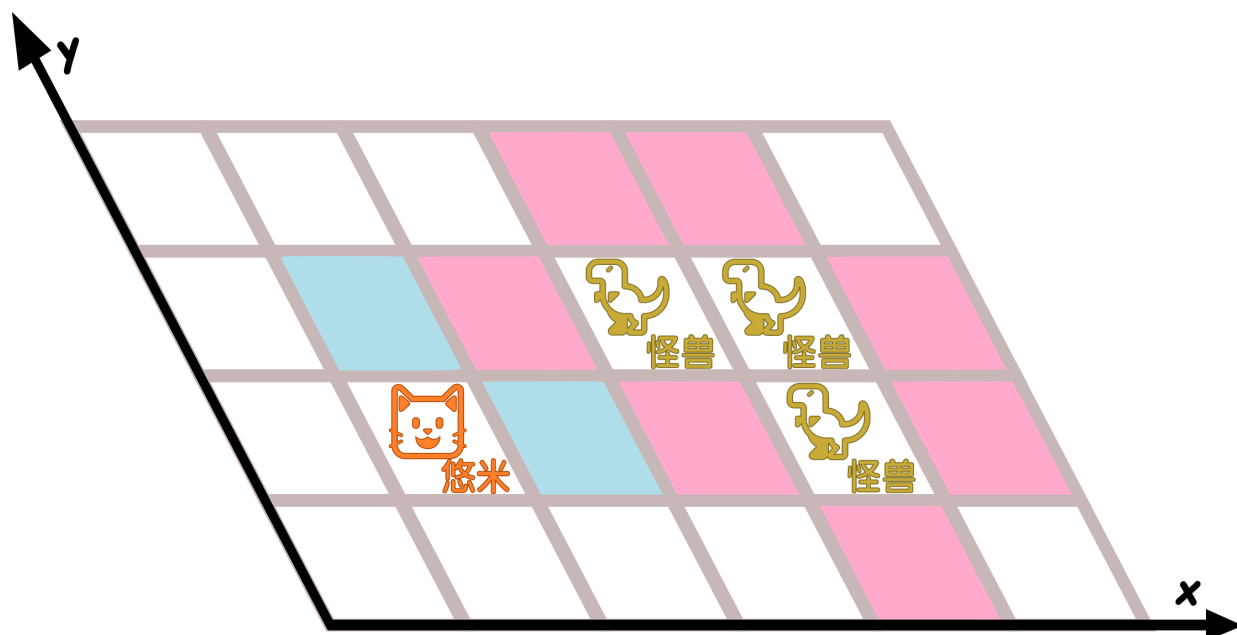
悠米正在飞行！

有一天，悠米吃完彩虹猫粮后困意袭来，打了个盹却开始做噩梦了，梦见她在一个无限大的二维棋盘上和恶魔进行一场回合制游戏，游戏规则如下：

- 在悠米的回合，悠米可以选择从  $(x, y)$  移动到  $(x + 1, y)$  或  $(x, y + 1)$ 。
- 在恶魔的回合，恶魔先选择棋盘上一个**已放置**怪兽的格子  $(x, y)$ ，然后从  $(x + 1, y), (x, y + 1), (x - 1, y), (x, y - 1)$  中**选择一格**再放置一只怪兽，并且**可以放在悠米所在的格子**。
- 如果某一时刻悠米与怪兽位于同一格，则悠米会被怪兽抓住，即逃离失败；否则悠米永远也不会遇到怪兽，即逃离成功。

假设悠米一开始位于  $(x_1, y_1)$ ，且初始时刻，棋盘上有 1 只怪兽位于  $(x_2, y_2)$ 。

如下图中浅蓝色区域表示悠米可以移动到的格子，浅红色区域表示恶魔可以放置新的怪兽的格子。



第一个回合是悠米的回合，已知悠米和恶魔均会以最优方式进行游戏。

问：悠米能否成功逃离恶魔的围困？

## Input

本题包含多组测试数据。

第一行有一个整数  $T$  ( $1 \leq T \leq 10000$ )，表示一共有  $T$  组测试数据。

接下来每行包括 4 个非负整数  $x_1, y_1, x_2, y_2$  ( $0 \leq x_1, y_1, x_2, y_2 \leq 10^9$ )，表示游戏开始时悠米的坐标  $(x_1, y_1)$  与初始棋子的坐标  $(x_2, y_2)$ ，保证两个坐标不相同，即  $(x_1, y_1) \neq (x_2, y_2)$ 。

注意，数据范围仅表示初始坐标在  $[0, 10^9]$  范围内；在游戏中，悠米可以走出该范围，怪兽也可以被放置在该范围之外，只要满足相应规则即可。

## Output

输出共  $T$  行，对于每组测试数据输出一行，若悠米能成功逃离恶魔的围困，输出 `YUUMI`，否则输出 `GG`。

## Sample #1

### Sample Input #1

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

### Sample Output #1

```
1 YUUMI
2 GG
3 YUUMI
```

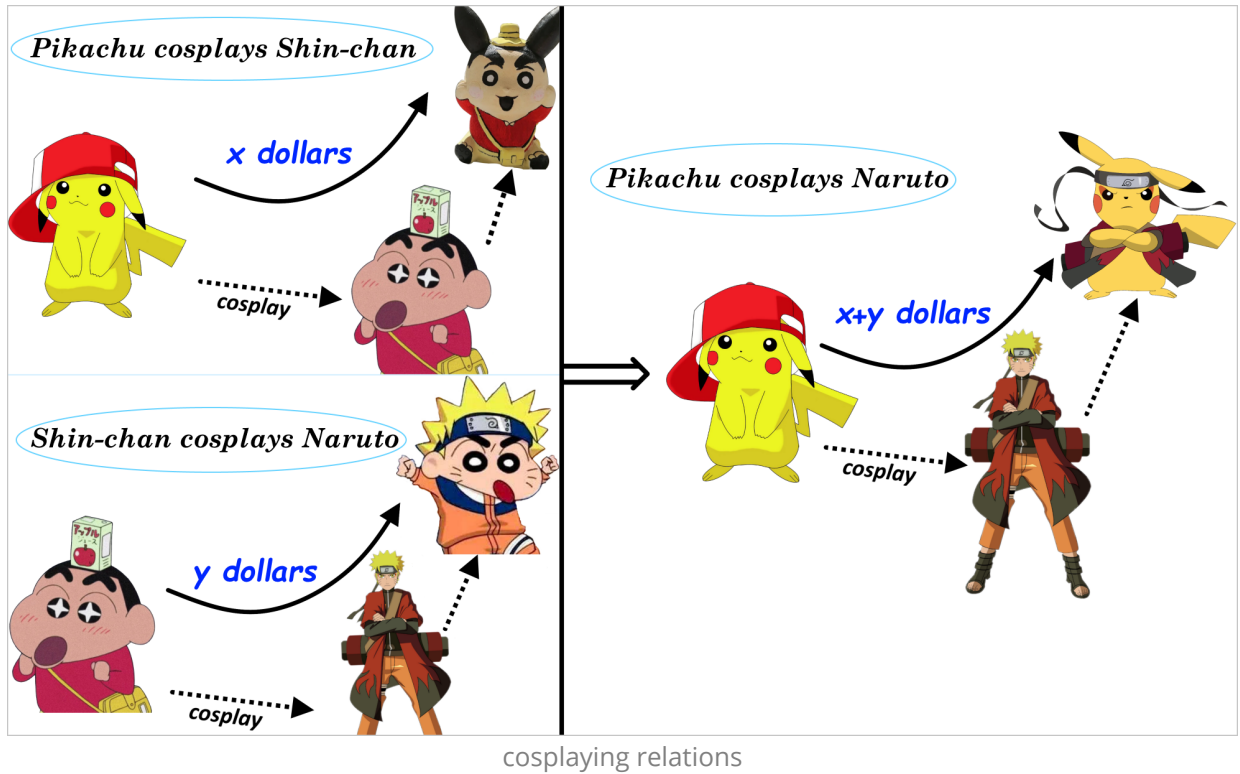
# 干饭协会的扮装节

## Description

In the EatMore Union, there are  $n$  characters present at the party to celebrate the *Cosplay Festival* (扮装节).

They enjoy cosplaying others in this party, and if  $a$  can cost  $x$  to cosplay  $b$ , and  $b$  can cost  $y$  to cosplay  $c$ , then we conclude that  $a$  can also cost  $x + y$  to cosplay  $c$ .

For example, let  $a$  be *Pikachu* (皮卡丘),  $b$  be *Shin-Chan* (蜡笔小新),  $c$  be *Naruto* (鸣人), then we conclude cosplaying relations as described in the figure below:



By the way, one can always cosplay himself/herself without any cost, i.e.  $a$  costs 0 to cosplay  $a$ .

You are given  $m$  cosplaying relations, described as 3 integers  $a, b, w$ , meaning that  $a$  can cost  $w$  to cosplay  $b$ .

You are also given  $q$  queries. In each query, there are 2 integers  $x, y$  — if  $x$  **cannot** cosplay  $y$ , output  $-1$ ; otherwise, output the minimal cost for  $x$  to cosplay  $y$ .

## Input Format

The first line contains 3 integers  $n, m$ , and  $q$  ( $1 \leq n \leq 600, 1 \leq m \leq 10^3, 1 \leq q \leq 10^5$ ) — the number of people in this town, the number of cosplaying relations, and the number of queries.

Then  $m$  lines follow, where  $i$ -th line contains 3 integers  $a_i, b_i, w_i$  ( $1 \leq a_i, b_i \leq n, 1 \leq w_i \leq 10^6, 1 \leq i \leq m$ ), meaning that  $a_i$  costs  $w_i$  to cosplay  $b_i$ .

Then  $q$  lines follow, where  $j$ -th line contains two integers  $x_j$  and  $y_j$  ( $1 \leq x_j, y_j \leq n, 1 \leq j \leq q$ ), representing a query — if  $x_j$  **cannot** cosplay  $y_j$ , output  $-1$ ; otherwise, output the minimal cost for  $x_j$  to cosplay  $y_j$ .

## Output Format

Output  $q$  lines, where  $j$ -th line represents the answer to  $j$ -th query.

## Sample #1



### Sample Input #1

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

### Sample Output #1

```
1 3
```

### Sample #2

#### Sample Input #2

```
1 4 5 3
2 2 4 9
3 1 3 6
4 3 4 1
5 1 2 2
6 2 3 3
7 2 1
8 1 1
9 1 4
```

### Sample Output #2

```
1 -1
2 0
3 6
```

# 干饭协会的招聘书

## Description

由于 EatMore 干饭协会日益壮大，现已无法依靠纯人工处理繁忙的业务。因此，会长 lzl 决定招聘一批编程高手，借助计算机程序来处理各项业务。

招聘要求：应聘者在协会自建的 EatMore Online Judge 网站上（简称 emoji），评分不低于  $m$  就可以投递简历。

EatMore Online Judge				
<a href="#">首页</a> <a href="#">问题</a> <a href="#">练习&amp;比赛</a> <a href="#">状态</a> <a href="#">询问</a> <a href="#">排名</a>				
<div> <div>问题列表</div> <div> <div>难度</div> <div>标签</div> <div>keyword</div> <div>重置</div> </div> </div>				
#	题目	难度	总数	通过率
AK10086	Problem A	低	2	0.00%
G114514	Problem B	低	0	0%
I1919810	Problem C	低	0	0%

emoj 网站界面

为了更好地帮助大家提升编程水平并合理赋分，emoj 有如下规定：

- emoj 上的每道题都有一定的难度  $d$ ，分值  $f$ 。
- 当一名用户**第一次**通过一道分值为  $f$  的题目时，该用户的评分增加  $f$ （过题即得分，但**不能重复**获得同一道题的分值）。
- 如果某题难度  $d$  与某用户评分  $r$  之差**超过**阈值  $k$ （即  $d - r > k$ ），则该用户**不能**做这道题目。

pyq 掌握了如上信息之后，立马就开始筹备他的“假应聘，真篡位”计划了。不过，在行动之前，他必须要能投得进简历才行。

已知 emoj 共有  $n$  道题，pyq 做完第  $i$  道题需要耗费时间  $t_i$ 。

现在 pyq 注册了一个评分为 0 的新的 emoj 账号，他想知道，至少要花多少时间才能使评分达到  $m$ ，从而拥有投递简历的机会呢？

如果 pyq 永远也不能使评分到达  $m$ ，则输出  $-1$ 。

## Input

第一行有 3 个整数，分别表示 emoj 的题目总数  $n$  ( $1 \leq n \leq 3000$ )，投递简历需要的评分  $m$  ( $1 \leq m \leq 3000$ )，以及阈值  $k$  ( $1 \leq k \leq 3000$ )。

第二行有  $n$  个整数，其中第  $i$  个整数表示第  $i$  道题的难度  $d_i$  ( $1 \leq d_i \leq 3000$ )。

第三行有  $n$  个整数，其中第  $i$  个整数表示第  $i$  道题的分值  $f_i$  ( $1 \leq f_i \leq 3000$ )。

第四行有  $n$  个整数，其中第  $i$  个整数表示 pyq 做第  $i$  道题需耗时  $t_i$  ( $1 \leq t_i \leq 10^{12}$ )。

## Output

输出一个整数  $T$ ，表示 pyq 至少要花费时间  $T$  才能够拥有投递简历的机会；如果 pyq 永远也不能使评分到达  $m$ ，则输出  $-1$ 。

## Sample #1

### Sample Input #1

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

## Sample Output #1

```
1 | 3
```

## Sample #2

### Sample Input #2

```
1 | 2 1 1
2 | 2 9
3 | 3 8
4 | 9876543210 123467890
```

## Sample Output #2

```
1 | -1
```

# 干饭协会的捏脸师

## 题目描述

*hsy* 是 EatMore 干饭协会里著名的捏脸师，凡是曾经捏过的脸型，*hsy* 都可以再为他人捏出来（因为这个脸型从此以后就被 *hsy* 记住了，这样一来，*hsy* 想捏多少个就能捏多少个，随心所欲）。



*hsy* 温柔捏脸ing...

不过，如果 *hsy* 从来没有捏过脸型  $x$ ，但是有顾客想要捏出这种脸型  $x$ ，那么 *hsy* 需要先花费  $p_x$  元设计脸型  $x$ ，然后再为这位顾客捏出该脸型，并且以后也就能直接捏出脸型  $x$  了（因为 *hsy* 已经记住了脸型  $x$ ）。

现在有  $m$  位顾客，第  $i$  位顾客想把自己的脸型  $u_i$  捏成脸型  $v_i$ ，且顾客们的捏脸顺序可以由 *hsy* 自由安排（保证每一顾客  $u_i \neq v_i$ ，不然就没必要来捏脸了）。

请问 *hsy* 至少需要花费多少元才能把所有顾客捏成想要的脸型？

## 输入格式

本题包含多组测试数据。

第一行有一个整数  $T$  ( $1 \leq T \leq 10000$ )，表示一共有  $T$  组测试数据。

对于每组测试数据：

- 第一行是空行。
- 第二行有 2 个整数  $n, m$  ( $2 \leq n \leq 5 \cdot 10^5, 1 \leq m \leq 5 \cdot 10^5$ ), 表示不同的脸型数量为  $n$ , 要捏脸的顾客数量为  $m$ 。
- 第三行有  $n$  个整数  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq 10^9$ ), 表示设计脸型  $i$  的花费为  $p_i$  元。
- 接下来  $m$  行的第  $i$  行有 2 个整数  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ , 且  $u_i \neq v_i$ ), 表示第  $i$  位顾客想把当前脸型  $u_i$  捏成目标脸型  $v_i$ 。

保证所有测试数据的脸型总数  $\sum n$ 、顾客总数  $\sum m$  均不超过  $5 \cdot 10^5$ 。

## 输出格式

输出共  $T$  行, 对于每组测试数据输出一行一个整数, 表示 *hsy* 把所有顾客捏成想要的脸型的最小花费。

## 样例 #1

### 样例输入 #1

```

1 2
2
3 3 3
4 3 7 9
5 1 2
6 2 3
7 3 1
8
9 5 4
10 5 6 7 8 9
11 2 1
12 3 1
13 4 1
14 1 5
```

### 样例输出 #1

```

1 3
2 9
```

## 提示

**样例 #1 的第 1 组测试数据:**

- 花费 3 元设计脸型 1, 此后 *hsy* 能捏出脸型 1;
- 为第 3 位顾客捏出脸型 1, 此后 *hsy* 能捏出脸型 1, 3 (第 3 名顾客原来的脸型是 3);
- 为第 2 位顾客捏出脸型 3, 此后 *hsy* 能捏出脸型 1, 2, 3 (第 2 名顾客原来的脸型是 2);
- 为第 1 位顾客捏出脸型 2, 此时所有顾客均已捏成目标脸型。

总花费 3 元, 可以证明这是最小花费。

**样例 #1 的第 2 组测试数据:**

- 花费 9 元设计脸型 5, 此后 *hsy* 能捏出脸型 5;
- 为第 4 位顾客捏出脸型 5, 此后 *hsy* 能捏出脸型 1, 5;
- 分别为第 1, 2, 3 位顾客捏出脸型 1, 此时所有顾客均已捏成目标脸型。

总花费 9 元，可以证明这是最小花费。

# 干饭协会的排队论

## 题目描述

“叮铃铃，叮铃铃……”。

随着干饭铃的响起，干饭协会的成员们冲向食堂。



很快食堂窗口就排起了长队。大家边排队边看着窗口的菜单，讨论午餐吃什么。

没过多久，有同学就发现，因为前面有人比自己高，所以自己一直看不见菜单。所以这个同学想麻烦前面的同学一下低头，让他瞄一眼今天的菜单。

- 我们定义，对于某个参数  $k$ ，编号为  $i$  的同学（身高为  $a_i$ ）看不见菜单，当且仅当在他前面有一个编号为  $j$  ( $j < i$ ) 的同学，他的身高  $a_j$  满足  $a_j > a_i + k$ 。

现在有  $q$  次询问，每一次询问给出两个整数  $pos, k$ ，表示编号为  $pos$  的同学想知道，对于参数  $k$  第一个挡着他看菜单的同学的编号。

## 输入格式

第一行两个整数  $n, q$  ( $1 \leq n, q \leq 10^5$ )，分别表示队伍长度为  $n$  和询问次数为  $q$ 。

第二行有  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ )，第  $i$  个整数  $a_i$  表示第  $i$  个人的身高。

接下来  $q$  行，每行 2 个整数  $pos, k$  ( $1 \leq pos \leq 10^5, 0 \leq k \leq 10^9$ )，表示编号为  $pos$  的同学想知道，对于参数  $k$  第一个挡着他看菜单的同学的编号。

## 输出格式

输出  $q$  行，每行一个整数表示编号为  $pos$  的同学想知道，对于参数  $k$  第一个挡着他看菜单的同学的编号。如果不存在这样的同学，输出  $-1$ 。

## 样例

输出：

```
1 6 5
2 1 9 1 9 8 10
3 1 1
4 4 5
5 1 4
6 3 7
7 5 0
```

输出：

```
1 -1
2 -1
3 -1
4 2
5 4
```

# 干饭协会的整向量

## 题目描述

最近干饭协会的会长 *lzl* 痴迷于研究整向量。

整向量是这样定义的：

- 对于一个向量  $\vec{x} = (a, b)$ ，如果它的每一分量  $a, b$  都是整数，那么我们就称之为 整向量。

这一天 *lzl* 发现了三个非零整向量，它们分别是  $\vec{x} = (x_1, x_2), \vec{y} = (y_1, y_2), \vec{z} = (z_1, z_2)$ 。

*lzl* 很好奇，是否存在整数  $a, b$  使得  $a\vec{x} + b\vec{y} = \vec{z}$ 。

如果存在，你能告诉 *lzl* 一组满足上述条件的整数  $a, b$  吗。

## 输入格式

本题包含多组测试数据。

第一行一个整数  $t$  ( $1 \leq t \leq 10^5$ )，表示有  $t$  组测试数据。

接下  $t$  行，每行 6 个整数  $x_1, x_2, y_1, y_2, z_1, z_2$  ( $1 \leq |x_1|, |x_2|, |y_1|, |y_2|, |z_1|, |z_2| \leq 10^6$ )，表示 *lzl* 发现的三个非零整向量  $\vec{x} = (x_1, x_2), \vec{y} = (y_1, y_2), \vec{z} = (z_1, z_2)$  ("非零" 就是保证  $|x|, |y|, |z| > 0$ )。

## 输出格式

对于每组测试数据：

- 如果答案存在，第一行输出 YES，然后第二行输出  $a, b$  的值，且须满足  $|a|, |b| \leq 2 \cdot 10^{12}$ 。可以证明，如果答案存在，那么就一定存在至少一组答案  $a, b$  满足以上约束；如果有多组答案满足要求，输出任意一组即可。
- 如果答案不存在，直接输出一行 NO。

## 样例 #1

输入:

```
1 4
2 1 2 1 2 3 6
3 1 2 1 2 3 3
4 1 1 4 5 1 4
5 1 0 0 2 1 1
```

输出:

```
1 YES
2 1 1
3 NO
4 YES
5 -11 3
6 NO
```

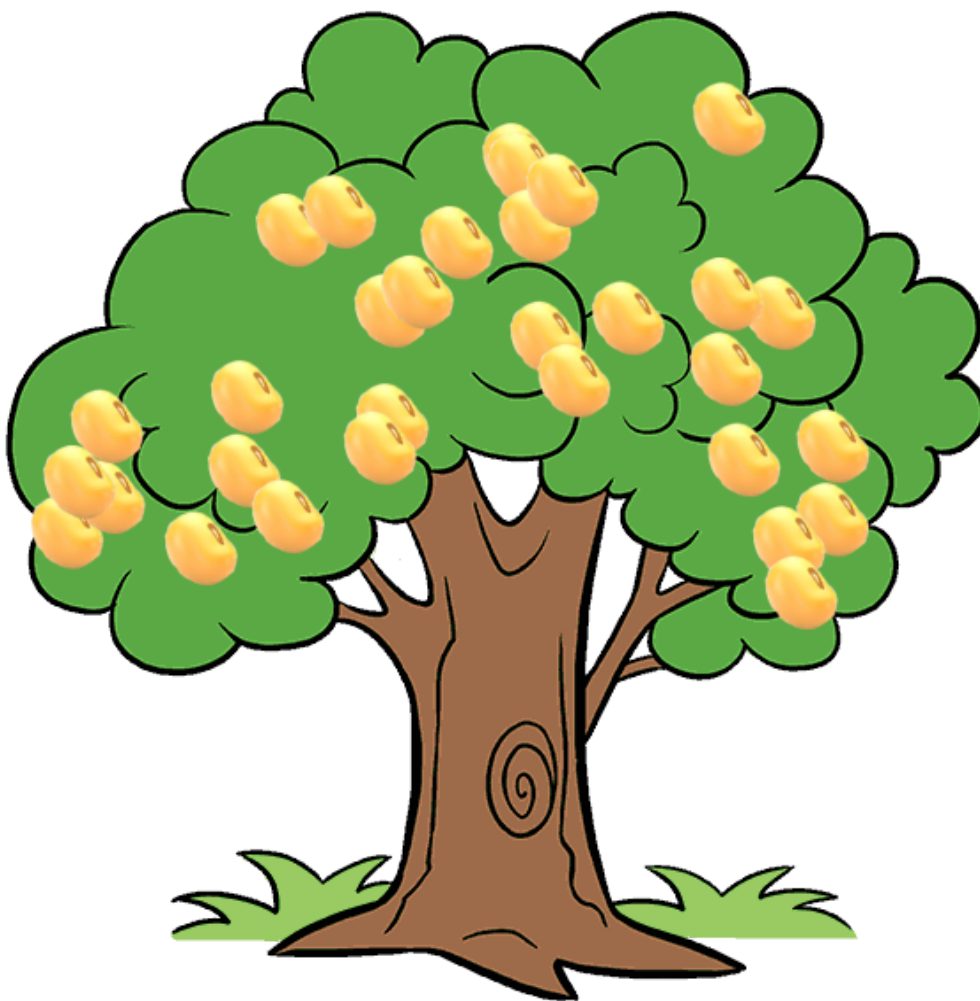
## Note

# 干饭协会的欢乐树

## 题目描述

众所周知干饭协会会长 *lzl* 是斗地主高手，他特别喜欢玩欢乐斗地主。其中欢乐豆就是欢乐斗地主中的货币。

这一天 *lzl* 不小心输掉了他所有的欢乐豆，所以他准备去欢乐树上摘欢乐豆。



欢乐树

为了简化问题，我们把欢乐树看作是一棵以 1 为根节点的树。

这颗树共有  $n$  个结点，编号为  $i$  的结点上有  $a_i$  个欢乐豆（由于某些原因，结点上的欢乐豆可以是负数）。

为了让摘欢乐豆的过程看起来更欢乐，lzl 将进行  $q$  次操作。

每次操作都是以下两种操作中的一个：

- 1 pos d：选定以编号为  $pos$  的结点为根结点的子树，从该子树中的每个结点摘取  $d$  个欢乐豆（如果  $d$  为负数则表示把欢乐豆放回该结点）。
- 2 pos：查询编号为  $pos$  的结点有多少个欢乐豆（可能为负）。

## 输入格式

第一行两个整数  $n, q$  ( $1 \leq n, q \leq 10^5$ )，表示结点数为  $n$ ，操作次数为  $q$ 。

接二行  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $-10^9 \leq a_i \leq 10^9$ )，表示初始时刻第  $i$  个节点上的欢乐豆数量。

接下来  $n - 1$  行，每行两个整数  $u, v$  ( $1 \leq u, v \leq n$ )，表示节点  $u, v$  之间有一条边。保证输入的数据形成一棵树。

接下来  $q$  行，每行表示下列两种操作之一：

- 1 pos d：选定以编号为  $pos$  的结点为根结点的子树，从该子树中的每个结点摘取  $d$  个欢乐豆（如果  $d$  为负数则表示把  $|d|$  欢乐豆放回该结点，输入保证  $1 \leq pos \leq n$ ，且  $-10^9 \leq d \leq 10^9$ ）。
- 2 pos：查询编号为  $pos$  的结点有多少个欢乐豆（答案可能为负，输入保证  $1 \leq pos \leq n$ ）。



## 输出格式

对于每一次查询操作，输出一个整数，表示查询结点的欢乐豆个数。

## 样例

输入：

```
1 5 6
2 7 3 7 8 0
3 1 2
4 1 3
5 3 4
6 3 5
7 2 3
8 1 1 -10
9 2 3
10 1 3 20
11 2 3
12 2 5
```

输出：

```
1 7
2 -3
3 17
4 10
```

## 干饭协会的签到题

### Description



EatMore 干饭协会在招纳新成员时，会对新人进行考核，要求其解决一道“入会资格题”。

不过，这道题总是那么简单，以至于大家都称之为签到题。

就在今天，干饭协会又开始招新啦！这道签到题是这样的：

给定一个包含  $n$  个元素的数组  $a$ ，你可以对其进行如下操作：

- 从  $a$  中任选两个数  $x, y$ , 将它们从  $a$  中移除, 并向  $a$  中添加一个数  $x - y$ 。

显然, 经过  $n - 1$  次操作后, 数组  $a$  恰好只剩一个元素; 请问: 最后这个元素最大是多少呢?

注意, 从数组  $a$  中移除  $x$  时, 尽管可能有多个值等于  $x$  的元素, 但是也**只删除其中一个**。

## Input

第一行有 1 个整数, 表示数组  $a$  的初始大小  $n$  ( $2 \leq n \leq 5 \cdot 10^5$ )。

第二行有  $n$  个整数, 表示数组  $a$  的初始  $n$  个元素  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9, 1 \leq i \leq n$ )。

## Output

输出一个整数, 表示进行  $n - 1$  次操作后, 数组  $a$  中唯一一个元素的最大可能值。

## Sample #1

### Sample Input #1

```
1 | 2
2 | 1 1
```

### Sample Output #1

```
1 | 0
```

## Sample #2

### Sample Input #2

```
1 | 3
2 | 23333 114514 5201314
```

### Sample Output #2

```
1 | 5292495
```

## Sample #3

### Sample Input #3

```
1 | 4
2 | 1 1 1 1
```

## Sample Output #3

```
1 | 2
```

## Hint

### 样例 #1:

初始数组  $a$  中仅含两个数 1 和 1，故进行如下操作：

先把它们从  $a$  中移出，此时  $a = []$ ，再把  $1 - 1 = 0$  插入到  $a$  中，于是  $a = [1]$ ，此时已经无法再操作了，因此答案为 1。

# 干饭协会的金币树

## Description

EatMore 干饭协会的会员大食堂门口有一颗金币树，树上每一个结点都有一定量的金币。

一个正午，富哥们 chd 正打算来食堂干饭，但他突然看着这棵树出了神——

chd 认为，只有**从根节点开始到每个叶子结点的路径上的金币数量之和都相等**的树才是吉利的。

chd 想把这棵树变成吉利的，但他只能在任意结点上添加金币，不能减少、移动树上已有的金币。

chd 想知道，最少要花费多少个金币才能使得这棵树是吉利的？当然也可以不花费金币，此时输出 0 即可。

## Input

第一行有 1 个整数  $n$  ( $2 \leq n \leq 5 \cdot 10^5$ )，表示金币树的结点数  $n$ 。

第二行有  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9, 1 \leq i \leq n$ )，表示初始时刻，编号为  $i$  的结点有  $a_i$  个金币。

接下来  $n - 1$  行中的第  $i$  行有 2 个整数  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ )，表示结点  $u_i, v_i$  之间有一条边。

数据保证这些边构成一棵树，并规定编号为 1 的结点是根节点。

## Output

输出一个整数，表示 chd 最少需花费的金币数。

## Sample #1

### Sample Input #1

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

## Sample Output #1

```
1 | 0
```

## Sample #2

### Sample Input #2

```
1 | 4
2 | 100 1 60 60
3 | 1 2
4 | 1 3
5 | 3 4
```

## Sample Output #2

```
1 | 119
```

# 干饭协会的金手帕

## 题目描述

相传干饭协会有一个祖传的金手帕，每一次出去聚餐都会带上它，并用它来做游戏，据说会带来好运。



金手帕

某次比赛后，干饭协会全体成员一起去聚餐。由于菜还没有上齐，大家就开始玩丢手帕的游戏了。

游戏规则是这样的：

- 有  $n$  名成员参加游戏，编号从 1 到  $n$ 。
- 如果手帕在编号为  $x$  的人手中，他就会把手帕传给编号为  $a_x$  的人。

现在  $pyq$  哥哥有  $q$  个想法，每个想法表示为两个数字  $pos, d$  表示  $pyq$  哥哥想知道起始位置为  $pos$ ，传递  $d$  次之后手帕在谁手中。

由于  $pyq$  哥哥和吉吉国王一样急，所以你能在  $1s$  内告诉他答案吗？

## 输入格式

第一行两个整数  $n, q$  ( $1 \leq n, q \leq 10^5$ ), 表示有  $n$  个人做游戏,  $pyq$  哥哥有  $q$  个想法。

第二行  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n, 1 \leq i \leq n$ ), 表示第  $i$  个人下一次会把手帕传给  $a_i$ 。

接下来  $q$  行每行两个整数  $pos, k$  ( $1 \leq pos \leq n, 1 \leq k \leq 10^9$ ), 表示  $pyq$  哥哥想知道起始位置为  $pos$ , 传递  $d$  次之后手帕在谁手中。

## 输出格式

输出  $q$  行, 每行一个整数表示答案。

## 样例

输入:

```
1 6 3
2 2 3 1 1 6 5
3 3 5
4 4 2
5 5 999999
```

输出:

```
1 2
2 2
3 6
```

# 干饭协会的魔法棒

## 题目描述

有一天, EatMore 干饭协会会长 `lzl` 突然不见了! 大家都非常着急!

一查监控发现, 原来是 `lzl` 拿书时, 书柜略微倾斜, 柜顶的魔法棒滚下来, 把 `lzl` 变成了白纸上的字符串 `lzl`。

但 `lzl` 似乎因为魔法棒的非正常释放, 意外地掌握了操控字符串  $s$  的能力, 正愁无题可出的 `lzl` 突然有了灵感, 于是 `ta` 将计就计, 在不经意间逃离了这张白纸躲了起来, 然后在白纸的顶部写下了下面这道题:

- 下面是一个长度为  $n$  且仅含小写字母的字符串  $s$ , 接下来会发生  $m$  个事件, 事件分为以下两种:
  - `change p c`: 将  $s$  的第  $p$  个字符  $s_p$  改为  $c$ , 其中  $c$  为单个小写字母;
  - `count-lzl l r`: 输出在  $s$  的子串  $s_l s_{l+1} \dots s_r$  中不同的等于 `lzl` 的子序列个数。

你能解决这道题吗? 只要你解决了这道题, `lzl` 就会忍不住发出“哇喔”的声音从而无所遁形。

作为提示, 字符串  $t$  的子串是  $t$  中某一段连续的字符串, 而子序列不必连续。

例如, 记字符串  $t$  为 `csust`, 那么 `sus` 是  $t$  的子串和子序列, `css` 是  $t$  的子序列但并非子串, `stc` 既不是  $t$  的子序列也不是  $t$  的子串。

## 输入格式

第一行包含一个长度为  $n$  的仅含小写字母的字符串  $s$  ( $3 \leq n \leq 2 \cdot 10^5$ ), 表示初始字符串。

第二行有 1 个整数  $m$  ( $1 \leq m \leq 2 \cdot 10^5$ ), 表示总事件数。

接下来  $m$  行，每行表示下面两者中的一个事件：

- `change p c`：将  $s$  的第  $p$  个字符  $s_p$  改为  $c$  ( $1 \leq p \leq n$ , 保证  $c$  为单个小写字母)；
- `count-lzl 1 r`：输出在  $s$  的子串  $s_l s_{l+1} \cdots s_r$  中不同的等于 `lzl` 的子序列个数 ( $1 \leq l \leq r \leq n$ )。

保证对于每组测试数据，`count-lzl` 事件至少出现 1 次。

## 输出格式

对于所有 `count-lzl` 事件，输出一行一个整数表示答案。

## 样例 #1

### 样例输入 #1

```
1 | 11111
2 | 5
3 | change 3 z
4 | count-lzl 2 4
5 | count-lzl 1 5
6 | change 3 x
7 | count-lzl 1 5
```

### 样例输出 #1

```
1 | 1
2 | 4
3 | 0
```