

PASSWORD0--Graph

时间限制：100ms，空间限制：256MB

题目背景

此题为PASSWORD1，2的前奏。。。

题目描述

ZHT 十分的生气！他急于前往 LKY 家，可在他眼前出现了由 n 个城市， m 条道路组成的图，每条路通过都需要一定时间和一定费用，ZHT 想走最小路径到 LKY 家。但眼前的景象却难倒了兢兢业业学习计算机的他，他需要你的求助。

最小路径的定义如下：定义每条 ZHT 所在城市到 LKY 家所在城市的路径总时间是各条道路旅行时间的和，总费用是各条道路所支付费用的总和。一条路径越快，或者费用越低，该路径就越好。严格地说，如果一条路径比别的路径更快，而且不需要支付更多费用，它就比较好。反过来也如此理解。如果没有 LKY 一条路径比某路径更好，则该路径被称为最小路径。

最小路径可能不止一条，也有可能不存在路径

现在，ZHT 想知道他有多少种方法到 LKY 家。

输入格式

第一行有四个整数，城市总数 n ，道路总数 m ，起点和终点城市 s, e ；

接下来的 m 行每行描述了一条道路的信息，包括四个整数，两个端点 p, r ，费用 c ，以及时间 t ；

两个城市之间可能有多条路径连接。

输出格式

仅一个数，表示 ZHT 的走法总数。

样例输入

```
4 5 1 4
2 1 2 1
```

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

样例输出

```
2
```

数据范围与提示

对于全部数据，

$1 \leq n \leq 100$, $0 \leq m \leq 300$, $1 \leq s, e, p, r \leq n$, $0 \leq c, t \leq 100$, 保证
 $s \neq e, p \neq r$ 。

PASSWORD1--Maths

时间限制：200ms，空间限制：256MB

题目描述

蒟蒻 ZHT 同学历经千辛万苦（见PASSWORD0），来到了 LKY 同学家门口，却发现此门竟是一扇密码门，门上显示一个整数 n ，要求你将 $1 \sim n$ 的数排列，使得所有数都不在原位上，密码为这样排列的方案数，由于 LKY 为 ZHT 考虑，因此数太大，于是请将答案取模 10007 后再输入密码。ZHT 由于智商不够，他又找到了你，他急需你的帮助！

输入格式

输入整数 n 。

输出格式

输出 LKY 家密码门的密码。

样例输入

```
4
```

样例输出

```
9
```

数据范围与提示

对于10%的数据， $1 \leq n \leq 100$ ；

对于100%的数据， $1 \leq n \leq 10^7$ 。

PASSWORD2--Find

时间限制：200ms，空间限制：256MB

题目描述

由于 ZHT 频繁打开 LKY 家的门，于是 LKY 把旧密码门换成了全新的密码门。门上显示 n 个整数数对 (a_i, b_i) (i 为数对编号)，还有一整数 m ，让你选出 m 个数对，设选出的数对编号为 p_1, p_2, \dots, p_m ，密码为选出的 m 对 $\frac{\sum a_i}{\sum b_i}$ ($i \in p$) 的最大值。

ZHT 突然发现，密码要保留四位小数，他十分想进去，需要求助你，让你来帮他打开全新密码门。

输入格式

第 1 行输入两整数 n, m ；
第 2 行到 $n + 1$ 行为数对 (a_i, b_i) 。

输出格式

输出 LKY 家密码门的密码。

样例输入

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

样例输出

```
0.7142
```

数据范围与提示

对于 10% 的数据, $1 \leq m \leq n \leq 20$;

对于 100% 的数据, $1 \leq m \leq n \leq 10000$, $1 \leq |a_i|, b_i \leq 100$.

答案保留四位小数!!!

PASSWORD3--String

时间限制：200ms，空间限制：256MB

题目背景

HZY 巨佬AK完NOIP后，于是想学（tui）习（fei），无聊中的他构思出一款游戏，把游戏规则告诉了 LKY、ZHT 和你，但需要计算机来快速地判断输赢，由于 HZY 巨佬不屑于开发这种如此简单的游戏，于是想让你开发出判断输赢的程序。

题目描述

ZHT 终于走进了 LKY 家的家门，由于十分疲惫，想跟 LKY 玩 HZY 开发的游戏，游戏规则如下：

LKY 和 ZHT 分别给出字符串 A 、 B （只含大、小写字母）和整数 C 、 D ，由于他俩十分有默契， A 包含 B （保证 A 比 B 长）。

这时统计出 B 在 A 中出现的位置（位置从1开始计算，总是记 B 在 A 中每次开始的位置），组成了一个集合 $\{pos\}$ 。

定义"分契合值" $l_i = \min(|(int)A[pos_i] - (int)A[pos_j]|)(1 \leq j < i)$,

特殊的， $l_1 = A[pos_1]$ ，"总契合值" S 为 $\sum l_i$ 。

若 C 离 S 较近，输出"L"；

若 D 离 S 较近，输出"Z"；

若 C 离 S 与 D 离 S 一样近，输出"AK NOIP!"（有叹号）。

ZHT 十分贪玩，于是决定玩 T 局，现在请你写一个程序，让你给出每局的输赢情况。

输入格式

第一行一个整数 T 。

之后 $4 \times T$ 行，每四行中依次有：字符串 A 字符串 B 猜测数 C 猜测数 D 。

输出格式：

共 T 行，第 i 行表示第 i 局的输赢情况。

样例输入

```
3
AB
AB
1
1
AB
AB
1
5
AB
AB
5
1
```

样例输出

```
AK NOIP!
L
Z
```

数据范围与提示

B 在 A 中出现次数 $\leq 10^5$, $B.size() \leq A.size() \leq 10^6$, $T \leq 20$, C, D 均在 `int` 范围内。

PASSWORD4--Maintain

时间限制：800ms，空间限制：256MB

题目背景

ZHT 由于 HZY 的游戏而十分愤怒，于是他前往 HZY 家修改游戏源代码...

ZHT 来到 HZY 的电脑前，只见屏幕上显示：“回答如下问题，进入系统。”

这时，ZHT 发现的 HZY 桌上的纸条上有一道题：

题目描述

纸条上问题如下：有初始数列 a_1, a_2, \dots, a_n ，还有模数 MOD

操作：

- (1) 把数列中的一段数全部乘一个值；
- (2) 把数列中的一段数全部加一个值；
- (3) 把数列中的一段数全部取模一个值；

问题:数列中的一段数的和，由于答案可能很大，你只需回答这个数模MOD的值。

蒟蒻ZHT无能为力，只好求助拥有计算机的你。

输入格式

第一行两个整数 n 和 MOD ($1 \leq \text{MOD} \leq 10^9$)。 $n, m \leq 10^5$

第二行含有 n 个非负整数,从左到右依次为 a_1, a_2, \dots, a_n , ($0 \leq a_i \leq 10^9, 1 \leq i \leq n$)。

第三行有一个整数 m ，表示操作总数。

从第四行开始每行描述一个操作，输入的操作有以下三种形式：

操作1: “1 t g c ”(不含双引号)。表示把所有满足 $t \leq i \leq g$ 的 a_i 改为 $a_i \times c$ ($1 \leq t \leq g \leq n, 0 \leq c \leq 10^9$)。

操作2: “2 t g c ”(不含双引号)。表示把所有满足 $t \leq i \leq g$ 的 a_i 改为 $a_i + c$ (

$1 \leq t \leq g \leq n, 1 \leq c \leq 10^9$)。

操作3: “3 t g c”(不含双引号)。表示把所有满足 $t \leq i \leq g$ 的 a_i 改为 $a_i \% c$ ($1 \leq t \leq g \leq n, 2 \leq c \leq 10^9$)。

问题: “Q t g”(不含双引号)。询问所有满足 $t \leq i \leq g$ 的 a_i 的和模 MOD 的值 ($1 \leq t \leq g \leq n$)。

同一行相邻两数之间用一个空格隔开，每行开头和末尾没有多余空格。

输出格式

对每个问题，按照它在输入中出现的顺序，依次输出一行一个整数表示询问结果。

样例输入

```
7 43
1 2 3 4 5 6 7
5
1 2 5 5
Q 2 4
2 3 7 9
Q 1 3
Q 4 7
```

输出样例

```
2
35
8
```

数据范围与提示

对于 30% 的数据， $n, m \leq 1000$;

对于 100% 的数据， $n, m \leq 10^5$ 。

PASSWORD5--Count

时间限制：1s，空间限制：256MB

题目背景

经历过无数次密码考验的 ZHT 在PASSWORD4过后，决定向 HZY 发起总攻!!!

题目描述

ZHT & HZY 大战展开...

而你，既是 ZHT 的军师，又是 HZY 派遣 ZHT 方的间谍。

战场上，HZY 有 n 个堡垒，有 $n - 1$ 条堡垒间的道路（树形结构，以1为根），每个堡垒都驻扎着不同种类的军队，第 i 个堡垒驻扎 p_i 种军队。

ZHT 决定在首次进攻中，攻击 x, y （含 x, y ）两堡垒之间的所有堡垒，但 ZHT 的军队只能在预先针对 HZY 的某种军队配置相应部队进攻（不然打不过），且只能攻打不超过 k 个。

ZHT 军队只可以攻击 $x \leftrightarrow y$ 之间所有驻扎某一种军队堡垒，ZHT 军队攻打越多的堡垒，成功率与军队存活率便会越低。

若攻打超过 k 个，首次进攻便会 100% 失败。

你身为间谍，请回答 ZHT 的 m 个计划，每次对于计划 x, y ，选择使 ZHT 成功率最低的 HZY 军队种类编号 P ，并给出 ZHT 此次需攻击的次数 cnt 。

出于保密，只需回答 cnt 。

下面回答有无几率成功，答案为 str 。

- 当 $cnt > k$ ，再回答"NO"（不含引号）
- 否则，再回答"YES"（不含引号）

输入格式

第一行有三个整数 n, m, k ，分别表示堡垒个数，计划个数，最大攻击数；

第二行输入 $p_i, i \in [1, n]$ ，表示第 i 个堡垒驻扎军队的编号；

接下来 $n - 1$ 行，每行有两个整数 x, y ，表示 x, y 有道路相连；

最后 m 行，每行有两个整数 x, y ，表示 ZHT 的计划。

输出格式

输出有 m 行

第 i 行回答 ZHT 的第 i 个计划，请输出 cnt, str （输出中用空格隔开）

cnt 如题面所示， $str = \text{YES or NO}$ ，表示有无几率成功

样例输入

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

样例输出

```
3 NO
2 YES
```

数据范围与提示

共15组数据：

对于前3组数据， $2 \leq k \leq n, m \leq 10^3$ ；

对于第4 ~ 10组数据， $2 \leq k \leq n, m \leq 5 * 10^4$ ；

对于第11 ~ 15组数据， $2 \leq k \leq n, m \leq 10^5$ ；

对于100%的数据， $1 \leq p_i \leq 10^9$ 。

PASSWORD4.5--Calc

时间限制：1s，空间限制：256MB

题目背景

此题为PASSWORD5的序幕...

题目描述

ZHT & HZY 的大战即将展开!
你现在仍是神犇 HZY 的助手。

HZY 已侦察 ZHT 的营地，侦察结果如下：

由于 ZHT 十分喜欢建系，他的营地也排成了整齐的坐标形式：

$(i, j) (1 \leq i \leq n, 1 \leq j \leq m, i, j \in \mathbb{Z})$.

由于 ZHT 十分喜欢数学与数论，坐标为 (i, j) 的营地被 ZHT 安放了防御值为 $p_{ij} = i * j * \gcd(i, j)$ 的防御器。

由于 ZHT 十分想取得大战的胜利，他又派遣 k 支精锐部队驻守营地，第 k 个营地驻守在横坐标为 d_i 的所有营地，此时横坐标为 d_i 的营地无法被攻击。

现在 HZY 需要摧毁 ZHT 的 $n * m$ 个营地，对于摧毁每个营地需要的攻击值 $a_{ij} = p_{ij}$ ，无法攻击的营地无需计入答案中。

同时，HZY 有将攻击值控制在 $10^9 + 7$ 以下的能力（通过 $\text{mod } 10^9 + 7$ ）。

由于 ZHT 的营地规模十分庞大，需要你替 HZY 计算—(事实上 HZY 已用 $O(\frac{1}{n*m*k})$ 复杂度的算法算出了答案)。

请你帮 HZY 计算攻击值之和。

输入格式

第一行输入两个整数 n, m ，表示坐标最大的行和列；

第二行输入一个整数 k ，表示精锐部队的数量；

第三行有 n 个整数，第 i 个整数 d_i 表示第 i 支部队驻守的横坐标；

保证输入的 d_i 升序。

输出格式

输出需要的攻击值之和。

样例输入

```
3 3
1
2
```

输出样例

```
42
```

数据范围与提示

T 表示数据编号。

T	$n, m \leq$	$k \leq$	$n = 1 \text{ or } m = 1$
1 ~ 3	10	{0, 0, 5}	<i>false</i>
4 ~ 6	10^3	{0, 5, 10}	<i>false</i>
7 ~ 10	10^6	{0, 0, 5, 10}	<i>false</i>
11 ~ 12	{ $10^6, 10^9$ }	0	<i>true</i>
13 ~ 20	10^9	10	<i>false</i>

对于 100% 的数据, $1 \leq n, m \leq 10^9$, $0 \leq k \leq 10$, $1 \leq p_i \leq n$.

注

若要使用头文件, 请将原include语句换成以下语句:

```
#if(__cplusplus == 201103L)
#include <unordered_map>
#include <unordered_set>
#else
#include <tr1/unordered_map>
#include <tr1/unordered_set>
namespace std
```

```
{  
    using std::tr1::unordered_map;  
    using std::tr1::unordered_set;  
}  
#endif
```

PASSWORD6--Operate

时间限制：2s，空间限制：256MB

题目背景

一年后，他们之间的战争仍在继续.....

题目描述

ZHT 有 n 个防御器，HZY 有一台激光枪。第 i 个防御器的防御参数为 ω_i 。

当激光枪波长对应频率等于被攻击目标固有频率时能对防御器造成最大伤害。故 ZHT 为了自己的防御器不被一次性消灭，他计划要对他的防御器的防御参数进行修改

ZHT 有三种修改方式：

1. 将防御器 $l \sim r$ 的防御参数全部 $+v$ ($v > 0$)；
2. 将防御器 $l \sim r$ 的防御参数全部变成 v ；
3. 将防御器 $l \sim r$ 的防御参数中 v 的倍数全部除以 v ($v > 1$)。

在他们的世界中，激光枪每次能对编号在 $l \sim r$ 之间的全部防御器进行攻击。每次 HZY 攻击防御器 $l \sim r$ 时，ZHT 会把防御器 $l \sim r$ 合成。

每次攻击的激光频率需要对应合成后的总固有频率。

已知编号 $l \sim r$ 合成后的固有频率为

$f_{lr} = 3.846 \times 10^{14} + (\sum_{i=l}^{r-1} (\omega_i^{\sum_{j=i+1}^r (\omega_j)})) \bmod M$ ，HZY 攻击防御器 $l \sim r$ 的激光调整量为 $y_{lr} = \sum_{i=l}^{r-1} \omega_i \omega_{i+1} \bmod M$ ，其中 $M = 51011754393599$ 。

你作为神犇 HZY 的助手（贯穿始终的设定），你已经知道按时间顺序排列的事件列表（即 HZY 的攻击与 ZHT 的修改），请计算 HZY 的每次攻击使伤害最大化的激光频率（即攻击范围内的总固有频率）与激光调整量。

输入格式

第 1 行有 2 个正整数 n, m ，分别表示防御器个数与总操作个数。

第 2 行有 n 个正整数，为 n 个防御器的初始防御参数。

第 3 行到 $m + 2$ 行，每行对应一个事件（均按时间顺序）。

操作格式如下：

对于形如 " $Z\ l\ r\ v$ " 的输入 (Z 为字符 Z)，对应事件为 ZHT 将防御器 $l \sim r$ 的防御参数全部 $+v$ ($v > 0$)。

对于形如 " $Z\ 2\ l\ r\ v$ " 的输入 (Z 为字符 Z)，对应事件为 ZHT 将防御器 $l \sim r$ 的防御参数全部变成 v 。

对于形如 " $Z\ 3\ l\ r\ v$ " 的输入 (Z 为字符 Z)，对应事件为 ZHT 将防御器 $l \sim r$ 的防御参数中 v 的倍数全部除以 v ($v > 1$)。

对于形如 " $H\ l\ r$ " 的输入 (H 为字符 H)，对应事件为 HZY 攻击防御器 $l \sim r$ 。

输出格式

对应每一次 HZY 的攻击事件，输出使伤害最大化的激光频率 f 与激光调整量 y ，以空格隔开。

为了方便查看结果，请将 $f - 3.846 \times 10^{14}$ 后再输出。

每一攻击事件的输出对应一行。

样例输入

```
5 5
6 7 9 1 3
Z 3 1 3 4
H 3 4
Z 1 1 5 2
H 3 5
H 2 3
```

样例输出

```
9 9
214359124 48
31381059609 99
```

样例解释

此样例解释中都已将 $f - 3.846 \times 10^{14}$ 。

第一个事件后：序列不变；

第二个事件后： $f_{34} = \omega_3^{\omega_4} \bmod M = (9^1) \bmod M = 9$,

$y_{34} = (\omega_3 \times \omega_4) \bmod M = (9 \times 1) \bmod M = 9$;

第三个事件后：序列变为 $[8, 9, 11, 3, 5]$ ；

第四个事件后：

$$f_{35} = (\omega_3^{\omega_4 + \omega_5} + \omega_4^{\omega_5}) \bmod M = (11^{3+5} + 3^5) \bmod M = 214359124,$$
$$y_{35} = (\omega_3 \times \omega_4 + \omega_4 \times \omega_5) \bmod M = (11 \times 3 + 3 \times 5) \bmod M = 48;$$

第五个事件后: $f_{23} = \omega_2^{\omega_3} \bmod M = (9^{11}) \bmod M = 31381059609,$

$$y_{23} = (\omega_2 \times \omega_3) \bmod M = (9 \times 11) \bmod M = 99.$$

数据范围与提示

对于 30% 的数据, $1 \leq n, m \leq 100$;

对于 100% 的数据, $1 \leq n, m \leq 10^5$, $1 \leq \omega_i \leq 10^4$, 每种事件出现的概率均等;

为了方便查看结果, 请将 $f - 3.846 \times 10^{14}$ 后再输出。

PASSWORD-R1

时间限制：1s，空间限制：256MB

题目背景

ZHT 曾经遇到一道信息题.....

题目描述

题面如下：

给定序列 $a_{1\dots n}$, q 次询问。

每次询问给定 l, r , 请你求出下列式子的值：

$$\text{xor}_{i=l}^r \text{xor}_{j=i}^r \text{xor}_{k=i}^j a_k$$

xor (异或) : $\text{xor}_{i=l}^r$ 表示 $a_l \text{ xor } a_{l+1} \text{ xor } \dots \text{ xor } a_r$, xor 在 C++ 中为 \oplus 。

现在请你帮助 ZHT 求解这道题。

输入格式

第一行两个整数 n, q 。

第二行 n 个整数 $a_{1\dots n}$ 。

接下来 q 行, 每行两个整数 l, r 。

输出格式

对于每个询问, 一行一个整数, 表示答案。

样例输入

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

样例输出

```
2
6
```

数据范围与提示

对于 30% 的数据, $1 \leq n, q \leq 500$.

对于 60% 的数据, $1 \leq n, q \leq 5 * 10^3$.

对于 100% 的数据, $1 \leq n, q \leq 2 \times 10^5, 0 \leq a_i \leq 10^9, 1 \leq l \leq r \leq n$.

PASSWORD-R2

时间限制：2s，空间限制：512MB

题目背景

ZHT 学数列.....

题目描述

ZHT 的面前有一个长度为 n 的数组。初始时，第 i 个位置的元素为 i （即所有位置上的元素依次为从 1 到 n 的正整数）。

接下来，你需要帮 ZHT 依次完成 m 个操作。每个操作可能是：

$A\ p\ q$ ，表示你需要修改数组中的所有元素。具体地，第 i 个位置的元素被修改成 $p \times i + q$ 。

$B\ x\ y$ ，表示你需要修改数组中的单个元素。具体地，第 x 个位置的元素被修改成 y 。

其中上述 p, q, x, y 均为整数。

每次操作结束后，你需要输出数组中所有元素的和。

输入格式

输入的第一行包含两个数 n, m 。

接下来 m 行，每行依次表示一个操作，格式为 $A\ p\ q$ 或者 $B\ x\ y$ ，意义如上所述。

同一行输入的相邻两个元素之间，用一个空格隔开。

输出格式

输出共 m 行，每行表示每个操作结束之后，数组中所有数的和。

样例输入

```
5 4
B 1 5
```

```
A -1 0
A 1 2
A 2 -5
```

样例输出

```
19
-15
25
5
```

样例解释

样例 1 解释：

第 1 次操作结束后，数组变为 $\{5, 2, 3, 4, 5\}$ ；

第 2 次操作结束后，数组变为 $\{-1, -2, -3, -4, -5\}$ ；

第 3 次操作结束后，数组变为 $\{3, 4, 5, 6, 7\}$ ；

第 4 次操作结束后，数组变为 $\{-3, -1, 1, 3, 5\}$ 。

数据范围与提示

对于测试点 1, 2，保证 $n \leq 100$ 。

对于测试点 3, 4，保证所有的操作均为 B 类型。

对于测试点 5, 6, 7, 8，保证所有的操作均为 A 类型。

对于所有数据，保证 $n \leq 50,000,000, m \leq 500,000$ 。

对于每个操作，保证 $|p| \leq 1000, |q|, |y| \leq 10^9, 1 \leq x \leq n$ 。

PASSWORD7--Final

时间限制：2s，空间限制：128MB

题目背景

最终，ZHT 与 HZY 重归于好，而他的 OI 生涯也于今完结.....

问题描述

ZHT 与 HZY 举办宴会。

餐桌上有 n 盘菜，排成一排，从左到右数的第 i 盘菜编号为 i ，每盘菜有一个美味度 d_i 。

ZHT 会先吃掉编号为 a 的菜，这样位置 a 就空了。接下来每次他会选择一个与空出的位置相邻的菜中美味度最小的菜吃掉（要把好吃的留到最后）。你可以发现空出的位置一定是一个连续的区间。

为了让事情更加有趣，HZY 有时会给某一盘菜上加一点装饰，以增加它的美味度。他保证做完此操作后，这盘菜的美味度会变成所有菜中前 10 大的。而且在任何时候任意两盘菜的美味度都不同。

有时 HZY 好奇在 ZHT 吃掉某块特定的编号为 b 的菜之前，他会吃掉多少块菜。

请你帮助 HZY 编写一个程序，给出操作序列，回答 HZY 的询问。

输入格式

第一行两个正整数 n, a ，分别表示菜的盘数以及 ZHT 第一次会吃掉哪一盘菜。

第二行 n 个互不相同的正整数 d_1, d_2, \dots, d_n ，表示菜的初始美味度。

第三行一个正整数 q ，表示操作的个数。

接下来 q 行，每行形如 `E i e` 或 `F b` 表示一次操作：

- `E i e`：将编号为 i 的菜的美味度提升至第 e 美味的。保证在操作之前比编号为 i 的菜美味的菜的数量至少为 e 。
- `F b`：询问 ZHT 在吃掉编号为 b 的菜之前会吃掉多少盘菜。

输出格式

对于每个 **F** 操作，输出一行一个数表示答案。

样例输入

```
5 3
5 1 2 4 3
17
F 1
F 2
F 3
F 4
F 5
E 2 1
F 1
F 2
F 3
F 4
F 5
E 5 2
F 1
F 2
F 3
F 4
F 5
```

样例输出

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

数据范围与约定

对于所有数据，保证 $1 \leq n \leq 2.5 \times 10^5$ ， $1 \leq q \leq 5 \times 10^5$ ， $1 \leq d_i, a, i, b \leq n$ ， $1 \leq e \leq 10$ 。

子任务编号	分值	特殊限制
1	15	$n, q \leq 10^4$
2	15	$n \leq 2.5 \times 10^4$ 且 F 操作的数量不超过 500
3	20	$q \leq 10^5$ 且 E 操作的数量不超过 100
4	50	无特殊限制