

北京理工大学第十一届程序设计新生赛

The 11-th BIT Campus Programming Contest for Junior Grade Group

现场赛 Onsite Round



题目列表 Problem List

A	魔法练习
B	玩牌
C	数三元图
D	博弈
E	Before the deadline
F	随机数破解程序
G	鲭鱼圣者
H	回文串分割
I	猜数字
J	圆神
K	数据结构基本功
L	数路径
M	猫猫虫虫虫

请勿在比赛开始前翻阅试题!
Do not open before the contest has started.

2023 年 06 月 03 日

Problem A. 魔法练习

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

欢迎来到北京理工大学第十一届程序设计新生赛！作为北理工的新生，想必大部分同学对魔法（魔法）并不是很熟悉，因此接下来将教大家一些基本的魔法技巧，希望你能在本场比赛中用到。

首先，让我们复习一下 C 语言程序设计的基本知识——基本数据类型能存储的数值范围：

- int：32 位有符号整数，可以表示 $-2^{31} \sim 2^{31} - 1$ 范围内的整数（ $2^{31} = 2,147,483,648$ ）
- long long：64 位有符号整数，可以表示 $-2^{63} \sim 2^{63} - 1$ 范围内的整数（ $2^{63} = 9,223,372,036,854,775,808$ ）

同时，介绍一个基本的标记： $x \bmod y$ 表示 x 除以 y 的余数，读作 x 模 y 。在 C/C++ 中，你可以使用 `a % b` 求出 a 除以 b 的余数。

对于任意的正整数 x, y, p ，以下公式总是成立：

- $(x + y) \bmod p = ((x \bmod p) + (y \bmod p)) \bmod p$
- $(x \times y) \bmod p = ((x \bmod p) \times (y \bmod p)) \bmod p$

因此，如果你试图求一个很大的数字除以 p 的余数，你可以放心的在过程中的每次运算后都对 p 取模，以防止数值超出数据类型能够表示的范围。

现在，我们来做一个小练习。给定长度为 n 的数组 a ，你需要求出 a 中所有元素的乘积除以 p 的余数。

Input

第一行两个整数 n, p ($1 \leq n \leq 10^5, 1 \leq p \leq 10^9$)，表示数组 a 的长度，以及所求答案的除数。

第二行 n 个整数，第 i 个整数为 a_i ($0 \leq a_i < p$)，表示数组中的第 i 个元素。

Output

输出一行一个整数，表示数组 a 中所有元素的乘积除以 p 的余数。

Examples

standard input	standard output
3 2035 2023 6 3	1819
3 1000000000 999999999 999999998 999999997	999999994

Note

在样例 1 中， $2023 \times 6 \times 3 = 36414$ ， $36414 \div 2035 = 17 \cdots 1819$ ，因此你需要输出 1819。

在样例 2 中，两个数的乘积会超出 `int` 类型的范围，因此你应该使用 `long long` 类型。同时，三个数的乘积超出了 `long long` 类型的表示范围，因此你需要在每次乘完后，把答案对 1000000000 取模。如果你的程序输出了 -957777926 ，说明你没有使用 `long long` 类型计算乘积。如果你的程序输出了 265065466，说明你没有在计算过程中及时对结果取模，导致了 `long long` 类型的溢出。

Problem B. 玩牌

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

桌面上共有 n 个非空牌堆，第 i 个牌堆中有 a_i 张牌，第 i 个牌堆的第 j 张牌上的权值记为 $b_{i,j}$ 。

现在进行若干轮游戏。具体而言，游戏的第 t 轮需要从左往右按顺序选出 k 个牌堆，记为 $c_{t,1}, c_{t,2}, \dots, c_{t,k} (1 \leq c_{t,1}, c_{t,i} < c_{t,i+1}, c_{t,k} \leq n)$ 。从这些牌堆中各选择一张牌并取出。从第 $c_{t,i}$ 个牌堆中取出的牌记为 $d_{t,i}$ ，要求满足 $\{d_{t,i}\}$ 单调递增。

此外，对于游戏的任意两个相邻轮次，还需要满足前一轮的 第 k 张牌 小于 后一轮的 第 1 张牌，即 $d_{t,k} < d_{t+1,1}$ 。

当某一轮无法取出满足条件的 k 张牌时，游戏结束。

DLee 希望这个游戏能玩到天荒地老，聪明的你能告诉他这个游戏最多能进行多少轮吗？

Input

第一行 2 个正整数 $n (1 \leq n \leq 2 \times 10^5), k (1 \leq k \leq n)$ ，表示牌堆数量和每轮取出的卡牌数量。

接下来 n 行，第 $i + 1$ 行表示第 i 个牌堆的信息，格式为 $a_i, b_{i,1}, b_{i,2}, \dots, b_{i,a_i} (1 \leq a_i, \sum a_i \leq 10^6)$ ，含义如上文所示。

数据保证所有 $b_{i,j}$ 互不相同，且 $1 \leq b_{i,j} \leq \sum a_i$ ，即所有 $b_{i,j}$ 构成一个排列。

Output

一行一个整数，表示游戏的最大轮数。

Example

standard input	standard output
5 3 2 4 1 1 2 1 3 1 5 1 6	2

Note

样例 1 的解释如下：

第 1 轮： $c = 1, 2, 3, d = 1, 2, 3$ ；

第 2 轮： $c = 1, 4, 5, d = 4, 5, 6$ 。

因此最多进行两轮游戏。

Problem C. 数三元图

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

“学霸题，数三元图。 n 点有向图，点点一条边，图含三元环，共有多少图？”

以下是人话：

有一张 n 个点的有向图，已知每两个点之间有且仅有一条有向边连接，即对于任意两个点 $u, v (u \neq v)$ ，有且仅有 $u \rightarrow v$ 或是 $v \rightarrow u$ 。且该图包含至少一个三元环，即存在三个不同的点 a, b, c ，他们的连边关系为 $a \rightarrow b, b \rightarrow c, c \rightarrow a$ 。

询问可能的图有多少种？你需要输出答案除以 p 的余数。

Input

输入包含多组数据。

第一行一个整数 $T (1 \leq T \leq 10^4)$ ，表示数据的组数。

接下来 T 行，每行两个正整数 $n, p (1 \leq n \leq 10^6, 1 \leq p \leq 10^9$ ，不保证 p 是质数)，表示询问 n 个点图中符合条件的图的数量除以 p 的余数。

数据保证 $\sum n \leq 5 \times 10^6$ 。

Output

对于每组数据，输出一行一个整数，表示符合条件的图的数量。

Example

standard input	standard output
3	0
2 998244353	2
3 998244853	0
1000000 1	

Problem D. 博弈

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

给定一张 n 个点的无向图，其中包含 $n - 1$ 条边，第 i 条边连接点 i 和点 $i + 1$ 。

有红、绿、蓝三种颜色的棋子各一个，其中红色棋子位于点 r ，绿色棋子位于点 g ，蓝色棋子位于点 b 。（不同棋子的位置可以相同）

Alice 和 Bob 轮流进行以下操作，Alice 先操作：

- 选择一颗棋子，将其移动到相邻的位置上（不同棋子的位置可以相同）
- 若移动后，三个棋子位置构成的有序三元组 (r, g, b) 曾经在过去的某个时刻出现过了，则进行该操作的人判负，另一人获胜。（初始状态视作已经出现过的状态）

二人都采取最优策略的情况下，谁将取得胜利？

Input

输入包含多组数据。

第一行一个整数 $T(1 \leq T \leq 10^4)$ ，表示数据的组数。

接下来 T 行，每行四个正整数 $n, r, g, b(2 \leq n \leq 10^5, 1 \leq r, g, b \leq n)$ ，表示一个询问。

数据保证 $\sum n \leq 10^6$ 。

Output

对于每组数据，若 Alice 取胜，请输出 Alice；否则请输出 Bob。

Example

standard input	standard output
2	Alice
2 1 1 1	Bob
3 1 1 1	

Problem E. Before the deadline

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 1024 megabytes

Link 现在想从 LIT(Liangxiang Institute of Technology) 去 BIT(Beijing Institute of Technology)。

Link 计划坐地铁去 BIT。共有 n 座地铁站， m 条地铁线，其中 LIT 附近的地铁站编号为 1，BIT 附近的地铁站编号为 n 。假设第 i 条地铁线有自己固定的发车间隔 d_i ，且该线路的两个起点站会在时刻 kd_i ($k \in \mathbb{Z}$) 各发一班车。进出站时间、换乘时间均忽略不计。

已知现在的时刻是 t_1 ，他需要在 t_2 时刻（含）前到达 BIT。从 Link 的宿舍到 LIT 附近的地铁站需要 t_3 的时间，从 BIT 附近的地铁站到 BIT 需要 t_4 的时间。

Link 最近睡眠不足，他希望在宿舍多睡一会。请问 Link 最多还可以睡多久，才能按时到达 BIT。

Input

第一行四个整数 t_1, t_2, t_3, t_4 ($1 \leq t_1, t_2, t_3, t_4 \leq 10^9, t_1 < t_2$)，含义如题目所述。

第二行两个整数 n, m ($2 \leq n \leq 2 \times 10^5, 1 \leq m \leq 10^5$)，含义如题目所述。

接下来 $3m$ 行，每三行描述一条地铁线，对于第 i 条地铁线：

第一行两个整数 k_i, d_i ($2 \leq k_i \leq n, 1 \leq d_i \leq 10^9$)，表示该地铁线经过车站的数量和该地铁线起点站的发车间隔。

第二行 k 个整数，第 j 个整数 $a_{i,j}$ ($1 \leq a_{i,j} \leq n$ ，单条地铁线的 $a_{i,j}$ 互不相同) 表示该地铁线上行方向经过的第 j 个车站。（地铁是双向的，下行方向经过的车站自然为 $a_{i,k}, a_{i,k-1}, \dots, a_{i,1}$ ）

第三行 $k-1$ 个整数，第 j 个整数 $b_{i,j}$ ($1 \leq b_{i,j} \leq 10^9$ ，单条地铁线的 $\sum b_{i,j} \leq 10^9$) 表示该地铁线从车站 $a_{i,j}$ 运行到车站 $a_{i,j+1}$ 所需的时长。

数据保证 $\sum k_i \leq 5 \times 10^5$ 。

Output

输出一行一个整数，表示 Link 还可以睡的时长。

无论出于何种原因，如果 Link 在 t_1 时刻出发也不可能通过坐地铁的方式在 t_2 时刻前到达 BIT，请输出 -1。

Examples

standard input	standard output
1 10 1 1 2 1 2 2 1 2 2	4
1 10 1 1 3 1 2 2 1 2 2	-1

Note

在样例 1 中，Link 最晚可以睡到 5 时刻，然后在 6 时刻到达 LIT 地铁站，8 时刻到达 BIT 地铁站，9 时刻到达 BIT。如果他睡到了 6 时刻，则他在 7 时刻到达 LIT 地铁站时并没有地铁，需要等到 8 时刻，这将导致他在 11 时刻才能到达 BIT 并迟到，所以不行。

在样例 2 中，LIT 和 BIT 的地铁站并不能互相到达，因此 Link 不能乘坐地铁前往 BIT。

Problem F. 随机数破解程序

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

Alice 写了一个随机数生成器，但不幸的是，由于算法存在缺陷，这个生成器生成的数字会陷入一个长度为 n 的循环。不妨设循环的第 i 个数字为 a_i ($1 \leq i \leq n$)。

Bob 拥有一个长度为 m 的数组 b ，他使用 Alice 的随机数生成器得到了长度为 m 的数组 c ，满足 $c_i = a_{((i-1) \bmod n)+1} \bmod b_i (1 \leq i \leq m)$ ，其中 $x \bmod y$ 表示 x 除以 y 的余数。

Bob 试图破解 Alice 的随机数生成器，也就是想获得数组 a 。作为破解的第一步，给定数组 b 和 c ，你需要求出最小的可能的 n 。

Input

输入包含多组数据。第一行一个整数 $T(1 \leq T \leq 10^3)$ ，表示数据的组数。

对于每组数据：

第一行一个整数 $m(1 \leq m \leq 10^3)$ ，表示数组 b 和数组 c 的长度。

第二行 m 个整数，第 i 个整数为 $b_i(1 \leq b_i \leq 10^6)$ 。

第三行 m 个整数，第 i 个整数为 $c_i(0 \leq c_i < b_i)$ 。

数据保证 $\sum m \leq 10^3$ 。

Output

对于每组数据，输出一行一个整数，表示最小的可能的 n 。

Example

standard input	standard output
2	2
4	3
2 3 3 3	
1 1 0 1	
4	
1 14 5 14	
0 5 3 10	

Note

对于样例 1，一种可能的方案是 $a = [3, 1]$ ，可以证明不存在 $n = 1$ 的方案。

对于样例 2，一种可能的方案是 $a = [19, 19, 8, 10]$ ，但这种方案不是最小的。可以构造 $a = [10, 5, 3]$ ，并证明不存在 $n = 1$ 或 $n = 2$ 的方案。

Problem G. 鲭鱼圣者

Input file: standard input

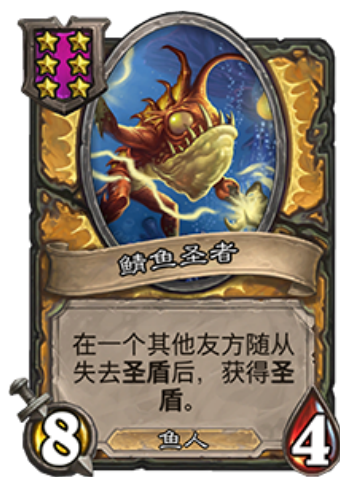
Output file: standard output

Time limit: 10 seconds

Memory limit: 1024 megabytes

鲭鱼圣者是游戏炉石传说中的一张卡牌。其具有 8 点攻击力，4 点生命值。在一个其他友方随从失去圣盾后，其会获得圣盾。

圣盾：该随从免疫其受到的第一次伤害。



敌方场上有且仅有 n 个**拥有圣盾**的鲭鱼圣者，你手上有 m 张卡牌，第 i 张卡牌的效果为：重复 a_i 次，随机对一名敌方随从造成 1 点伤害。

现在你想知道，你手上每张卡牌杀死所有敌方随从的概率。

请注意：生命值小于等于 0 的随从视为死亡，其不会再成为随机攻击的目标。

Input

第一行两个整数 n, m ($1 \leq n \leq 7, 1 \leq m \leq 400$)

第二行 m 个整数，第 i 个整数为 a_i ($1 \leq a_i \leq 400$)，描述了第 i 张卡牌。

Output

输出 m 行，每行一个浮点数，第 i 行的浮点数表示第 i 张卡牌杀死所有敌方随从的概率。

设你的答案为 a ，标准答案为 b ，你的答案正确当且仅当 $|a - b| \leq 10^{-6}$ 。

Examples

standard input	standard output
1 5	0.00000000114514
1 2 3 4 5	0.00000001919810
	0.00000001234567
	0.00000009999999
	0.99999999999999
2 2	0
9 10	0.0625

Note

对于样例 1，每次攻击的都是同一个目标。其在第 1 次受到攻击后会失去圣盾，而后每次攻击减少 1 点血量，在第 5 次攻击后，目标死亡。前 4 组数据的标准答案为 0，第 5 组数据的标准答案为 1，但是由于样例输出与标准答案的误差都没有超过 10^{-6} ，它们都被视为正确答案。

对于样例 2，前 9 次攻击显然不能杀死所有随从。第 10 次攻击能杀死随从，当且仅当其攻击一个随从 5 次，再攻击另一个随从 5 次，概率为 $C_2^1 \times (\frac{1}{2})^5 = \frac{1}{16}$ ，因此输出 0.0625。

Problem H. 回文串分割

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

给定一个字符串，请判断其是否能被表示为若干个回文串拼接的结果。

形式化的说：

一个字符串 S 是回文的，当且仅当其从前往后和从后往前读相同。

一个字符串 S 是好的，当且仅当存在某个 $n \geq 1$ 使得 $S = T_1T_2 \cdots T_n$ ，其中 $T_i(1 \leq i \leq n)$ 是回文串。

给定字符串 S ，请判断其是否是好的字符串。

Input

输入包含多组数据。

第一行一个整数 $T(1 \leq T \leq 10^6)$ ，表示数据的组数。

接下来 T 行，每行一个字符串 $S(1 \leq |S| \leq 10^6, S$ 仅包含小写英文字母)，表示被询问的串。

数据保证 $\sum |S| \leq 5 \times 10^6$ ，其中 $|S|$ 表示字符串 S 的长度。

Output

对于每组数据，若其是好的字符串，输出 **Yes**，否则输出 **No**。

Example

standard input	standard output
2	Yes
sosos	Yes
hahaha	

Note

sosos 本身就是回文的，因此输出 **Yes**。

hahaha 可以被划分为 hah 和 aha，二者都是回文串，因此输出 **Yes**。

Problem I. 猜数字

Input file: standard input
Output file: standard output
Time limit: 10 seconds
Memory limit: 1024 megabytes

这是一道交互题。

有两个隐藏的整数 $x, y (0 \leq x, y < 2^{60})$ ，你需要通过询问猜测 x, y 的数值。

每次询问中，你需要输出一行 `? a b`，需要满足 $0 \leq a, b < 2^{61}$ ，系统会返回给你 $\gcd(x+a, y+b)$ 。

其中， $\gcd(u, v)$ 指 u, v 的最大公因数。当 $u = 0$ 时， $\gcd(u, v) = v$ 。当 $v = 0$ 时， $\gcd(u, v) = u$ 。

对于每对隐藏整数 x, y ，你最多可以询问 200 次。

Interaction Protocol

首先读入一个整数 $T (1 \leq T \leq 100)$ ，代表猜测轮数。保证每轮隐藏的整数 x, y 都满足 $0 \leq x, y < 2^{60}$ 。

在每轮猜测中，你可以进行不超过 200 次询问。

每次询问中，若你询问的数是 a, b ，则需要输出一行 `? a b`，然后读入一个整数代表 $\gcd(x+a, y+b)$ 的结果。询问应满足 $0 \leq a, b < 2^{61}$ 。

当你确定结果后，输出一行 `! x y`，代表你猜测的结果分别为 x, y ，之后立刻开始进行下一轮猜测。

由于本题为交互题，为了让系统接收到你的输出，你在每次输出后都需要刷新缓冲区，否则你将收到 Time Limit 的结果。

你可以用如下方式刷新缓冲区（在每次输出一行内容之后）

- 在 C++ 中，执行 `fflush(stdout)`
- 在 Java 中，执行 `System.out.flush()`
- 在 Python 中，执行 `stdout.flush()`

本题是非自适应的，即隐藏的整数在询问开始前就已经确定且不会改变了。

喜报！为了让大伙更加专注于题目本身而不是交互方法，出题人决定给大家喂到嘴里！

以 C++ 为例，询问和猜测答案的函数实现如下：

```
long long ask(long long a, long long b){
    printf("? %lld %lld\n", a, b);
    fflush(stdout);
    long long res;
    scanf("%lld", &res);
    return res;
}
```

```
void guess(long long x,long long y){  
    printf("! %lld %lld\n",x,y);  
    fflush(stdout);  
}
```

Example

standard input	standard output
2	? 2 0
3	? 1 1
2	! 1 3 ? 3 1
3	? 10 5
10	? 7 2
7	! 0 5

Note

注意，样例中的询问并不能真的唯一确定结果，仅作为格式示例。

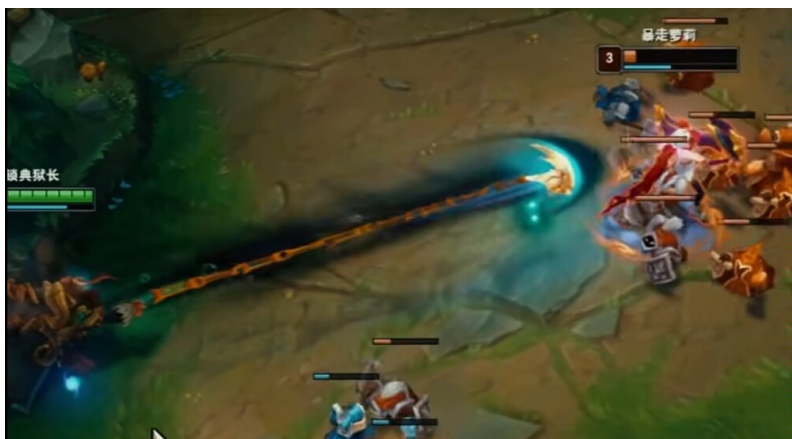
第一轮的结果是 1 3，第一次询问 2 0，结果为 $\gcd(1 + 2, 3 + 0) = 3$ ；第二次询问 1 1，结果为 $\gcd(1 + 1, 3 + 1) = 2$ 。

第二轮的结果是 0 5，第一次询问 3 1，结果为 $\gcd(0 + 3, 5 + 1) = 3$ ；第二次询问 10 5，结果为 $\gcd(0 + 10, 5 + 5) = 10$ ；第三次询问 7 2，结果为 $\gcd(0 + 7, 5 + 2) = 7$ 。

Problem J. 圆神

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

锤石 (Tresh) 是英雄联盟 (league of legend) 中一名辅助角色, 它的 Q 技能死亡判决是在短暂前摇后甩出钩子, 命中的第一个非友方单位会被眩晕, 并将其拉拽向锤石 1.5 秒。今天, DLee 在玩锤石, 但他总是勾不中想勾的目标, 在经过机制的研究后, 他发现可以抽象成这样一个问题:



锤石位于二维平面上的原点 $(0,0)$, 在二维平面上有 n 个敌人, 每个敌人的碰撞体积可以看作是位于 (x_i, y_i) , 半径为 r_i 的一个圆, 钩子可以看成从原点出发的一条射线, 这条射线上碰到的第一个圆 (相交或相切) 就是他能勾中的目标。现在 DLee 想知道, 他能够勾到这 n 个敌人中的多少个。

战场的情况瞬息万变, 因此 DLee 需要对多个情况进行分析。

Input

第一行一个正整数 T , 表示数据组数。

对于每组数据, 第一行一个正整数 $n (1 \leq n \leq 10^5)$ 。接下来 n 行, 每行三个整数 $x_i, y_i, r_i (-10^7 \leq x_i, y_i \leq 10^7, 1 \leq r_i \leq 10^7)$, 表示第 i 敌人的碰撞体积是以 (x_i, y_i) 为圆心, 半径 r_i 的圆。

数据保证:

- $\sum n \leq 10^6$
- 任意两个圆相距距离大于 1, 即 $(x_i - x_j)^2 + (y_i - y_j)^2 > (r_i + r_j + 1)^2, \forall i \neq j$
- 圆与玩家不会重合且距离大于 1, 即 $x_i^2 + y_i^2 > (r_i + 1)^2$

Output

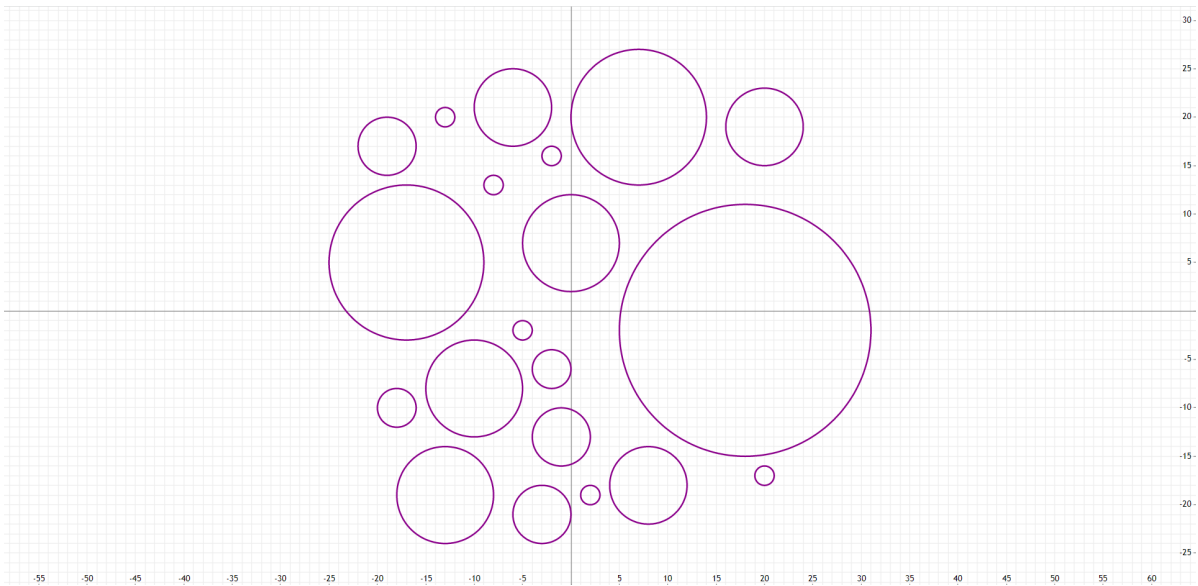
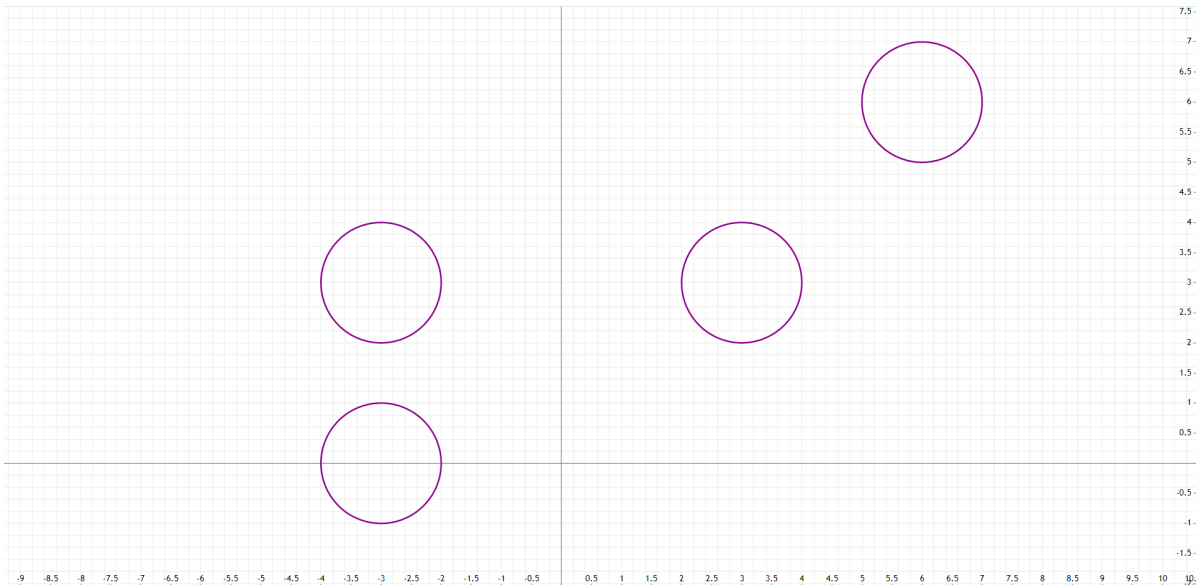
对于每组数据, 输出一行仅一个整数, 表示当前情况下 DLee 能勾到的敌人数量。

Example

standard input	standard output
2	3
4	11
3 3 1	
6 6 1	
-3 0 1	
-3 3 1	
20	
-17 5 8	
18 -2 13	
0 7 5	
8 -18 4	
-19 17 3	
-3 -21 3	
7 20 7	
-13 -19 5	
-6 21 4	
-10 -8 5	
-8 13 1	
-1 -13 3	
-2 -6 2	
20 -17 1	
-18 -10 2	
-2 16 1	
-13 20 1	
2 -19 1	
-5 -2 1	
20 19 4	

Note

样例数据的图如下：

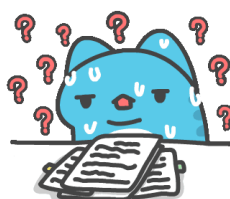


Problem K. 数据结构基本功

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 1024 megabytes

“为了打好数据结构基本功，深入理解数据结构精神，贯彻落实代码实现工作，我们需要努力学习数据结构的深刻内涵……”

ACM 训练基地在开会，可是只听了两句之后猫猫虫就睡了过去，醒来的他面对留下的课后作业感到不知所措，幸运的是，他认识一个数据结构大神，那就是你！为了不让可怜的猫猫虫被学长责罚，你决定帮助他完成数据结构作业。



作业题目如下：

给定一棵以 1 号点为根节点的树，树上每个点有点权 $a_i \in \{0, 1\}$ ，接下来你需要进行 q 次操作，操作共分为两种：

- 1 $u\ v\ x$ ，代表将树上的一条以 u, v 为端点的简单路径上所有点的点权设为 x 。
- 2 u ，代表查询以 u 为根节点的子树内满足 $x < y, a_x \oplus a_y \oplus a_{lca(x,y)} = 0$ 的点对 (x, y) 的数目。

其中 $\oplus$ 代表按位异或，即 C 语言中的位运算符 $\wedge$ 。

$lca(x, y)$ 指 x 和 y 两个点的最近公共祖先，如样例 3 中， $lca(4, 5) = 3$ 。

Input

第一行两个整数 $n, q (2 \leq n \leq 3 \times 10^5, 1 \leq q \leq 3 \times 10^5)$ ，代表树上点数和操作数。

第二行 n 个整数 $a_i (a_i \in \{0, 1\})$ ，代表初始点权。

第三行 $n - 1$ 个整数，第 i 个整数为 $f_i (1 \leq f_i \leq i)$ ，代表第 $i + 1$ 个点和第 f_i 个点之间有一条边。

接下来 q 行，每行为一个操作：

对于每个操作，每行第一个整数为 $op (op \in \{1, 2\})$ ，代表操作类型。

- 若 $op = 1$ ，则随后三个整数 $u, v, x (1 \leq u, v \leq n, x \in \{0, 1\})$ ，代表覆盖的链的两个端点，以及这条链被覆盖为的权值。
- 若 $op = 2$ ，则随后一个整数 $u (1 \leq u \leq n)$ 代表需要查询的子树根节点。

Output

对于每个询问，输出一行一个整数代表这次询问的答案。

Examples

standard input	standard output
2 1 0 0 1 2 1	1
3 3 0 0 1 1 1 2 1 1 1 2 1 2 1	1 0
5 5 1 0 0 1 1 1 1 3 3 1 5 4 1 1 2 4 0 2 3 1 2 1 1 2 1	1 5

Note

对于样例 3，第一次询问时，经过前两次覆盖之后的树点权为 $\{0, 0, 0, 0, 1\}$ ，子树 3 中合法的点对为 $(3, 4)$ 。

第二次询问时，经过前三次覆盖之后的树点权为 $\{1, 1, 0, 0, 1\}$ ，子树 1 中合法的点对为 $(1, 3), (1, 4), (2, 3), (2, 4), (3, 4)$ 。

Problem L. 数路径

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

给定一个根节点为 1 的二叉树，点 i 的权值为 val_i 。求满足以下条件的路径 $a_1, a_2, \dots, a_m$ 的数目：

- 路径经过至少 2 个点，即 $m \geq 2$
- 路径上所有点的权值相等，即 $val_{a_1} = val_{a_2} = \dots = val_{a_m}$
- 对于 $1 \leq i < m$ ， a_{i+1} 为 a_i 的一个孩子

Input

第一行 1 个正整数 $n(1 \leq n \leq 500)$ ，表示树的点数。

接下来一行 n 个整数 $a_1, a_2, \dots, a_n(1 \leq a_i \leq n)$ ，表示每个点的点权。

接下来 n 行，第 $i + 2$ 行两个整数，表示点 i 的左、右孩子编号（0 表示不存在）。

Output

一行一个整数，表示满足条件的路径个数。

Example

standard input	standard output
8 2 2 1 1 2 2 3 3 0 5 4 3 0 0 0 0 8 6 2 0 0 0 7 0	7

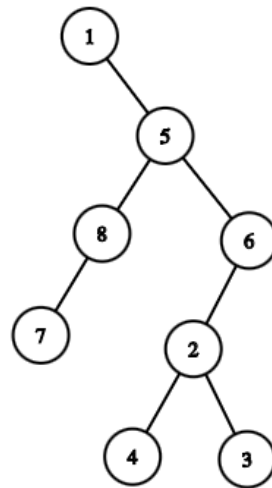
Note

满足条件的路径如下：

- 1, 5：权值都为 2
- 1, 5, 6：权值都为 2

- 1,5,6,2: 权值都为 2
- 5,6: 权值都为 2
- 5,6,2: 权值都为 2
- 6,2: 权值都为 2
- 8,7: 权值都为 3

树的结构如下所示:

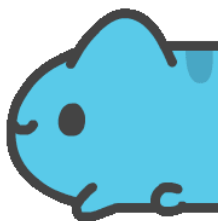


Problem M. 猫猫虫虫虫

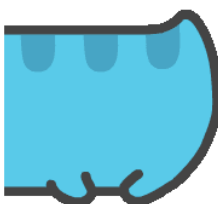
Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

众所周知，猫猫虫 capoo 有六条腿，分为前中后三对。其中前腿与中腿之间的距离和中腿与后腿之间的距离相等。

这天，猫猫虫在走路，迎面而来的兔兔看到了猫猫虫此时的前中两对腿的坐标位置落在了区间 $[a, b]$ 中。



而跟着猫猫虫走路的狗子则看到了猫猫虫此时中后两对腿的坐标位置落在了区间 $[c, d]$ 中。



现在，兔兔和狗子把它们看到的位置的范围告诉了你，希望你计算出猫猫虫腿之间的距离的最大值和最小值。

具体的，设猫猫虫的腿间距为 len ，此时前腿所在的位置为 x ，则它们需要满足：

- $a \leq x \leq x + \text{len} \leq b$
- $c \leq x + \text{len} \leq x + 2\text{len} \leq d$

请根据以上信息计算出 len 的最大值 $\text{len}_{\max}$ 。当 len 的最大值为 0 时，猫猫虫的腿长在了一起，但由于这是猫猫虫的世界，我们认为这是正常的。

Input

一行四个整数， a, b, c, d ($0 \leq a, b, c, d \leq 10000$)， a, b 代表前中两对腿的坐标范围为 $[a, b]$ ， c, d 代表中后两对腿的坐标范围为 $[c, d]$ 。

数据保证不等式存在至少一组合解。

Output

一行一个浮点数 $\text{len}_{\max}$ ，表示计算出的猫猫虫腿间距的最大值，输出四舍五入保留两位小数。

Examples

standard input	standard output
1 3 3 5	2.00
0 0 0 0	0.00
0 10000 0 10000	5000.00
0 1 0 10000	1.00
0 10000 9999 10000	1.00