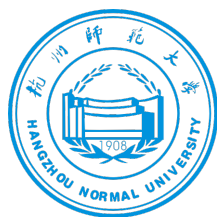


五月多校联合训练 (HZNU站)

China, Hangzhou

2023.5.27

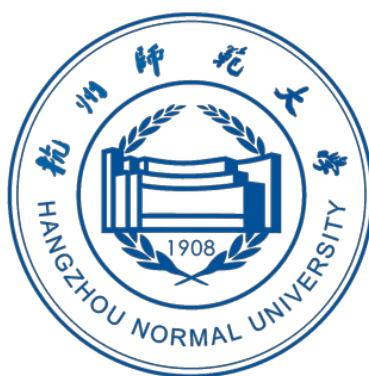


杭州师范大学
HANGZHOU NORMAL UNIVERSITY

Problems

A	括号
B	操作
C	买火柴的小男孩
D	tententen选月饼
E	企鹅取暖
F	我想要原石
G	我想玩光遇
H	排列
I	炸弹拆除
J	无限列车
K	这是一场豪赌，朋友！
L	饿了吧！

Do not open before the contest has started.



Problem A. 括号

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

learner 特别喜欢合法括号序列，可是他现在手上只有一个长度为 n 下标从 1 开始的数组 a ，于是他想要将这个数组变成一个合法括号序列。他选择一个整数 k ，执行以下程序：

```
1 s ← ""
2 for i ← 1 to n do
3   if a[i] < k then
4     s ← s + '('
5   else if a[i] > k then
6     s ← s + ')''
```

现在请你帮他计算有多少个整数 k 能使得 s 是一个合法括号序列。
合法括号序列定义是：

- 1. 空序列是合法括号序列。
- 2. 如果S是合法括号序列，那么(S)是合法括号序列。
- 3. 如果A和B都是合法括号序列，那么AB是合法括号序列。

Input

第一行包含一个正整数 n ($1 \leq n \leq 2 \cdot 10^5$) — 表示数组的长度。
第二行包含 n 个非负整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 2 \cdot 10^5$) — 题目给出的数组。

Output

输出一个整数，代表满足条件的 k 的个数。

Examples

standard input	standard output
4 4 10 5 10	4
4 1 4 4 10	1

Note

样例一中， k 可以为 6, 7, 8, 9，此时合法括号序列均为 "()
样例二中， k 只能为 4，此时合法括号序列为 "()

Problem B. 操作

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

有两个长度都为 n 的数组 a, b ，你可以进行如下操作：

- 选择一个数组，将该数组内的所有元素正负号反转。此操作的代价为 1。
- 选择一个数组，将该数组内的所有元素的值 $+1$ 。此操作的代价为 1。
- 选择一个数组，将该数组内的所有元素的值 -1 。此操作的代价为 1。
- 选择一个数组，将该数组重排。此操作的代价为 0。

若数组 a 能变成数组 b 则输出最小代价，否则输出 -1 。

Input

第一行包含一个正整数 n ($1 \leq n \leq 2 \cdot 10^5$) — 表示数组的长度。
第二行包含 n 个整数 a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — 表示数组 a 的每个元素。
第三行包含 n 个整数 b_1, b_2, \dots, b_n ($-10^9 \leq b_i \leq 10^9$) — 表示数组 b 的每个元素。

Output

输出一个整数，如果能将数组 a 变为数组 b ，则输出最小代价，否则输出 -1 。

Example

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

Problem C. 买火柴的小男孩

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

jjj 是一个特别有同情心的小男孩。在一个寒冬，他在街头遇上了一个卖火柴的小女孩。

小女孩十分可怜，抱着 jjj 的大腿说："哥哥，哥哥，我这里有 n 根火柴，如果我卖不完，我妈妈就不让我回家吃饭"。jjj 看着小女孩，十分心疼，于是大手一挥，买下了 n 根火柴。

但是他买了火柴有什么用呢？jjj 突然想到了家里还有一个正在上小学二年级的弟弟 stf，他经常看到 stf 在他的奥数书上做一种题目：给你一个不成立的火柴棒等式，你需要移动一根火柴，使得等式成立。

jjj 想使用买的 n 根火柴棒来出一个题考考他的弟弟，你能帮帮 jjj 吗？

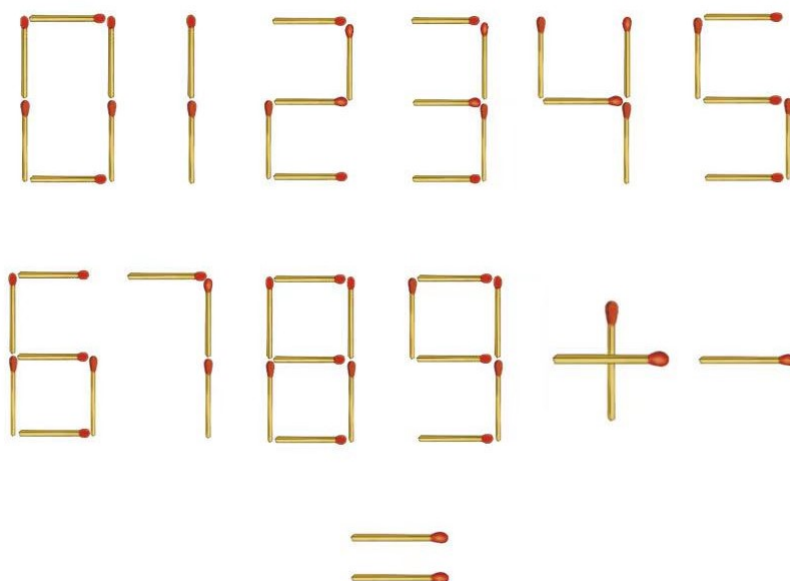
注意，你构造的火柴棒等式要满足如下几个条件：

- 参与运算的数字都是自然数。
- 运算符只含 $+$ ， $-$ ， $=$ 。有且仅有一个 $=$ ， $+$ 与 $-$ 数量不限。
- $=$ 的任意一边要么是单个自然数，要么是一个合法的运算表达式。
- 初始的等式是不成立的，移动一根火柴棒后成立。

在移动过程中，要遵循如下规则：

- 我们把一个符号、一个数字称为一个单位。在移动前和移动后，等式中单位个数不能改变。
- 组成 $=$ 的火柴棒不能移动。 $+$ 可以移动成 $-$ ， $-$ 可以放一根变为 $+$ ，但 $-$ 不能被移走。
- 不能把一个数字移动后拆成两个数字。例如：不能移动一根火柴棒，将 4 变为 11。
- 数字只能改变为数字。例如，不能移动一根火柴棒，将 1 变为 $+$ 。

所有数字和运算符号的表示方法见下图：



各数字和运算符号的表示方法

Input

本题有多组测试数据。

输入的第一行包含一个正整数 T ($1 \leq T \leq 10^3$) — 表示 T 组输入。

接下来的 T 行，每行包含一个正整数 n ($1 \leq n \leq 10^4$) — 表示 n 根火柴棒。

题目保证 T 组样例的 火柴棒个数 n 的和不超过 10^4 。

Output

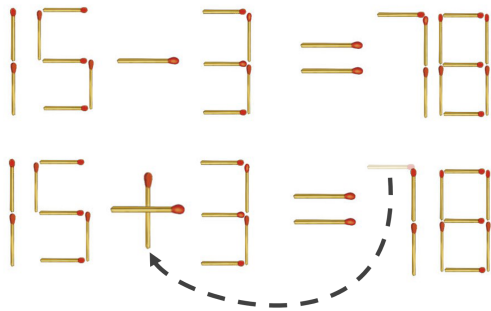
对于每组输入，若不存在这样的火柴棒等式，则在一行上输出 **No Solution**。否则输出两行，第一行表示初始时不成立的火柴棒等式，第二行表示移动一根火柴棒以后成立的火柴棒等式。

如果存在多种符合题意的答案，输出任意一种

Example

standard input	standard output
1	15-3=78
25	15+3=18

Note



样例 1 的解释

Problem D. tententen选月饼

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 512 megabytes

厨艺精湛的 tententen 做了 n 个月饼，这些月饼被排成一条直线摆放在桌面上，第 i 个月饼有一个美味值 a_i 。

现在她想要选择一些月饼拿去包装，然后送给实验室的小伙伴们！tententen 在选择月饼时会遵循下面的规则：

- 只选择连续的几个月饼。
- 在选择月饼中，美味值最大的月饼与美味值最小的月饼的差小于等于 k 。

现在，tententen 给你 q 次询问，第 i 次询问告诉你 l_i, r_i ，代表她只想选择这个区间内的月饼。

对于每次询问，你能告诉她有多少种选法吗？

我们认为两种选法不同，当且仅当存在一个月饼不同时出现在两种方案中。

Input

第一行包含三个正整数 n, k, q ($1 \leq n, q \leq 2 \cdot 10^5, 0 \leq k \leq 10^9$) — n 表示月饼的个数， k 表示美味值最大的月饼与美味值最小的月饼的差的最大值， q 表示询问的个数。

第二行包含 n 个非负整数 a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$) — 表示每个月饼的美味值。

接下来的 q 行中的第 i 行包含两个正整数 l_i 和 r_i ($1 \leq l_i \leq r_i \leq n$)，表示第 i 次询问在区间 $[l_i, r_i]$ 上进行。

Output

输出 q 行，第 i 行代表在第 i 次询问所限定的区间内，tententen 有多少种选法。

Example

standard input	standard output
5 2 4	13
1 3 3 5 3	9
1 5	3
1 4	1
2 3	
3 3	

Problem E. 企鹅取暖

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 512 megabytes

在极寒之地，有 n 只企鹅，第 i 只企鹅有一个体温值 a_i 。在第 0 天，每只企鹅单独为一个团队，每只企鹅的幸福值为 0。

在接下来的 q 天早晨，每天都会发生以下三种事件之一：

- 1 $u\ v$: 企鹅 u 所在的团队和企鹅 v 所在的团队合并为一个团队，一起抱团取暖。
- 2 $u\ v$: 企鹅 u 所在的团队集体运动，每只企鹅的体温值上升 v 。
- 3 u : 询问企鹅 u 的幸福值。

在每天晚上（包括第 0 天），每只企鹅都会获得一定的幸福值，大小等于其所在团队所有企鹅的体温值之和。

对于每个事件三，在一行上输出企鹅 u 的幸福值。

由于答案可能会很大，请输出对 $10^9 + 7$ 取模的结果。

Input

第一行包含两个正整数 n, q ($1 \leq n, q \leq 2 \cdot 10^5$) — n 表示企鹅的数量， q 表示发生事件的天数。

第二行包含 n 个正整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 2 \cdot 10^5$) — 每只企鹅的初始体温值。

接下来的 q 行，描述了每天发生了什么。每行是下面三种形式之一：

- 1 $u\ v$ ($1 \leq u, v \leq n$)
- 2 $u\ v$ ($1 \leq u \leq n, 1 \leq v \leq 2 \cdot 10^5$)
- 3 u ($1 \leq u \leq n$)

对于事件一，保证 u 和 v 之前不在同一团队。

Output

对于每个事件三，在一行上输出企鹅 u 的幸福值。由于答案可能会很大，请输出对 $10^9 + 7$ 取模的结果。

Examples

standard input	standard output
5 6 1 2 3 4 5 1 1 2 3 1 1 2 5 3 2 2 1 10 3 1	4 16 61
4 4 1 2 3 4 3 1 3 1 2 1 10 3 1	1 2 14

Problem F. 我想要原石

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

《原神》是由上海米哈游网络科技股份有限公司制作发行的一款开放世界冒险游戏。游戏发生在一个被称作“提瓦特”的幻想世界，游戏中的大陆也被称为提瓦特大陆。

然而，由于提瓦特大陆实在是太大了，游戏中设置了许多传送锚点。众所周知，每个传送锚点附近都有若干个原石（其实并没有），曾经有一位丰富经验的旅行者开辟了 $n - 1$ 条路和 n 个由路连通的传送锚点。为了便于后续的旅行者知道地图上原石的分布情况，他决定给旅行者一些提示，但是他没有直接将每个传送锚点附近的原石标注，而是标注了他所走过的路径的权值来考验后续的旅行者。对于一条路径连接的两个点 u, v ，其权值为：点 u 的原石数量异或点 v 的原石数量。

现在你来到了这片提瓦特大陆，但是你现在只有一次传送机会——即你可以选择一个传送锚点并到达，并且你将知道这个点的原石数量，请聪明的你回答这 n 个传送锚点附近原石数量的异或和。

你需要回答 q 次询问，每次询问告诉你传送的点及该点所有的原石数量，请你根据已有信息推断出这 n 个传送锚点附近原石数量的异或和。

Input

第一行包含一个正整数 n ($1 \leq n \leq 2 \cdot 10^5$) — 表示传送锚点的数量。

接下来的 $n - 1$ 行中的第 i 行包含三个正整数 u_i, v_i, w_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, 0 \leq w_i \leq 10^9$) — 表示传送锚点 u_i 和传送锚点 v_i 之间有一条权值为 w_i 的路径。

接下来的一行包含一个正整数 q ($1 \leq q \leq 5 \cdot 10^5$) — 表示询问的数量。

接下来的 q 行中的第 i 行包含两个正整数 u, x ($1 \leq u \leq n, 0 \leq x \leq 10^9$) — 表示初始传送位置和该位置的原石数量。

Output

输出 q 行，每行包含 1 个整数 — 表示每次询问的答案。

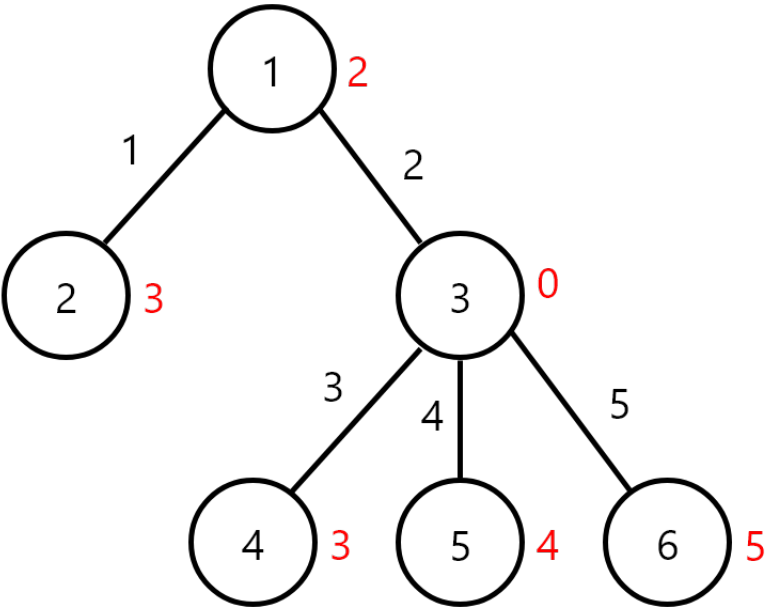
Example

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

Note

下面是样例一第一次询问的解释：

根据题意所述，初始点为 1，形成的路径如下，其中，红色为点权：



故答案为： $2 \otimes 3 \otimes 0 \otimes 3 \otimes 4 \otimes 5 = 3$

Problem G. 我想玩光遇

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

《光·遇》游戏世界是一个温暖人心的社交冒险游戏，这是一场拥抱自由和温暖的云端之旅。与玩家心爱的人们携手，在这座旷世的天空王国中翱翔，爱的力量将支持玩家一路前行。对于这样一款高质量的游戏，它的服务器经常爆满，想要进入游戏必须要排队，而现在 Satellite 是这个服务器的管理员。

今天，《光·遇》的服务器又开始排队了，在第 i 秒内有 a_i 个人进入队列。作为管理员的 Satellite 可以掌控队列的放行权，为了确保游戏服务器的稳定性，在每两次放行之间应该有至少 m 秒的间隔，每次释放最多允许 b 人。请注意，进入队列的人都是《光·遇》的忠实粉丝，在 n 秒内，他们不会离开队列。

请帮助 Satellite 制定一种放行策略，使得 n 秒内能玩上游戏的人最多。

Input

第一行包含三个正整数 n, m ($1 \leq m \leq n \leq 2 \cdot 10^5$), b ($1 \leq b \leq 10^9$) — n 表示时间长度, m 表示两次放行的间隔时间, b 表示每次最多允许放行的人数。

第二行包含 n 个正整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — 每秒进入服务器队列的人数。

Output

输出一个正整数，表示 n 秒内可以放行的最多的人数。

Examples

standard input	standard output
5 2 10 5 9 3 1 8	25
6 2 10 20 1 1 1 1 5	29

Note

样例一，他可以在 1, 3 和 5 秒时放行，答案为 25。

样例二，他可以在 1, 3 和 6 秒时放行，答案为 29。

Problem H. 排列

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

给你一个长度为 n 的数组 a ，这个数组是一个长度为 n 的排列，也就是说 1 到 n 的所有整数在这个数组中出现确定的一次。

在一次操作中，你可以选择数组中的两个元素，交换他们的位置。

问最少需要进行几次操作，使得数组中恰好有 k 个位置满足 $a_i = i$ ，或者说明这是不可能的。

Input

第一行包含两个整数 n, k ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq k \leq n$) — n 表示数组的长度， k 的意义见题目描述。

第二行包含 n 个非负整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — 题目给出的数组。

保证 a 是一个长度为 n 的排列。

Output

若不能通过操作使得恰好有 k 个位置满足 $a_i = i$ ，则输出 -1 。

否则，输出一个整数，表示最少的操作次数。

Examples

standard input	standard output
3 2 1 2 3	-1
3 0 1 2 3	2
3 1 1 2 3	1

Problem I. 炸弹拆除

Input file: **standard input**
Output file: **standard output**
Time limit: 5 seconds
Memory limit: 256 megabytes

这是一道交互题。

521 是黛蹟曜王国的国王，他拥有 n 个城市。城市间由 $n - 1$ 条双向道路连接，任意两座城市之间相互可达。

由于不满 521 的统治，Pancake 在某一条道路上埋下了一枚炸弹。521 为了避免自己的臣民因为爆炸而陷入恐慌，决心找到这颗炸弹并且排除它。

由于技术有限，他无法精确定位这颗炸弹的位置。但 521 有一颗预言水晶球，每一次使用水晶球，它可以知道从城市 u 出发到城市 v 的路上是否会遇到炸弹，由于预言水晶球的能力十分强大，它只能最多被使用 **521** 次。

521 只懂得如何用兵打仗，却不懂得如何合理使用水晶球来找到这个炸弹。你可以帮帮 521 国王吗？

Input

本题有多组测试数据。

输入的第一行包含一个正整数 T ($1 \leq T \leq 300$) — 表示有 T 组输入。对于每组测试数据：

第一行一个正整数 n ($2 \leq n \leq 1000$) — 表示黛蹟曜王国城市的数量。

接下来的 $n - 1$ 行，每行有两个正整数 $1 \leq u, v \leq n$ ，第 i 行代表编号为 i 的道路连接着城市 u 和城市 v 。

Interaction Protocol

对于每组数据，你可以使用最多 **521** 次水晶球的能力。每次使用水晶球的方式为输出：

- `"? u v"` ($1 \leq u, v \leq n$)。

代表使用水晶球来得知从城市 u 出发到城市 v 的路上是否会遇到炸弹。每次使用水晶球后，你需要从标准输入中读入一个整数。如果是 `"1"`，则表示会遇到炸弹，如果是 `"0"`，则表示不会遇到炸弹。

在任何时刻，如果你已经知道了炸弹藏在了哪条道路上，用如下格式输出单独的一行：

- `"! x"` ($1 \leq x \leq n - 1$)。

代表你告诉 521 国王，炸弹被藏在了编号为 `"x"` 的道路上。由于 521 国王的脾气十分火爆，你只有一次机会。

如果你在组数据中使用了水晶球超过 **521** 次，或者询问的方式不合法，交互程序将会立刻结束，你的程序运行结果将会返回 `"Wrong Answer"`。

每当你的程序使用了一次水晶球或是告诉 521 国王炸弹的位置后，你必须在末尾输出换行符并且刷新标准输出的缓冲区，否则你的程序运行结果将会返回 `"Idleness limit exceeded"`。你可以使用如下方式刷新缓冲区：

- 在 C++ 中： `fflush(stdout)` 或者是 `cout.flush()`；
- 在 Java 中： `System.out.flush()`；
- 在 Python 中： `stdout.flush()`；
- 在 Pascal 中： `flush(output)`；

- 其他语言请参阅相关文档。

Example

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

Problem J. 无限列车

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 512 megabytes

有一天, Alice 向你提出了一个问题:

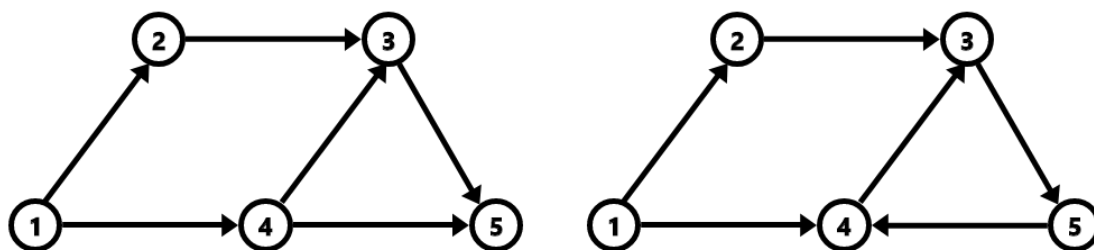
给定一张 n 个点 m 条边, 带点权的有向无环图 G_1 , 点 i ($1 \leq i \leq n$) 的点权为 w_i (初始值为 0), 保证图联通, 可能会出现重边。要求你在 G_1 的复制品 G_2 上依次加入 k 条有向边, 然后分别在 G_1 和 G_2 上做 10^{114514} 次操作。Alice 希望你告诉她有多少种加边方案, 满足最终 $G_1 = G_2$ 。

一次操作定义为: 等概率地选择图上的一条有向边 (u, v) , 然后执行 $w_v = \max(w_v, w_u + 1)$ 。

对于方案 S_1 和 S_2 , 若存在一个 i ($1 \leq i \leq k$), 满足第 i 次加入的边不同, 则 S_1 和 S_2 视作不同。

对于图 G_1 和 G_2 , 当 G_1 和 G_2 节点数相同, 且对于任意的 i ($1 \leq i \leq n$), 都满足在 G_1 和 G_2 中的 w_i 相同, 则 G_1 与 G_2 视作相同。

由于答案可能很大, 请输出对 $10^9 + 7$ 取模后的结果。



左图是有向无环图, 右图不是。

Input

第一行输入三个整数 n, m, k ($3 \leq n \leq 2 \times 10^5, n-1 \leq m \leq 8 \times 10^5, 0 \leq k \leq 10^9$) - n 表示节点数, m 表示边数, k 表示 Alice 要求加的边数。

接下来 m 行, 第 i 行包含两个整数 u_i, v_i ($1 \leq u_i, v_i \leq n, u \neq v$) - 代表一条由 u_i 指向 v_i 的有向边。

数据保证会形成一张联通的有向无环图, 可能会存在重边。

Output

输出一个整数, 代表有多少种加边方案。

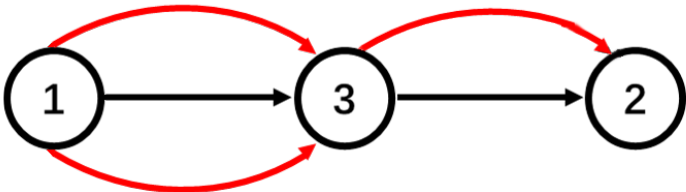
由于答案可能很大, 请输出对 $10^9 + 7$ 取模后的结果。

Examples

standard input	standard output
3 3 3 1 2 2 3 1 2	27
4 7 2 1 2 1 4 2 3 1 4 1 3 1 2 4 3	25
3 8 0 2 3 2 1 2 3 3 1 2 1 2 1 2 3 3 1	1
3 3 3 1 3 3 2 1 2	27

Note

下图是样例 4 的其中一种加边方案:



第一次加入 (1,3) , 第二次加入 (1,3) , 第三次加入(3,2)

Problem K. 这是一场豪赌，朋友！

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

你收到了一个三缺一的邀请，这一次，青雀打算带你玩仙舟上历史悠久的棋牌戏「帝垣琼玉」。游戏的规则如下：

- 每幅牌由 3 种不同花色的「琼玉牌」组成。初始时每位玩家手里有 n 张牌。
- 若某一时刻，一位玩家手中所有的琼玉牌的花色都相同，则称该玩家进入了「暗杠」状态。
- 轮到玩家的回合时，他可以摸 2 张牌（每张牌的花色是等概率的），并重复以下操作 2 次：
 任选一种牌数最少的花色，丢弃该花色的牌 1 张。

给你初始手牌数 n 以及 3 种花色的牌的数量，请你求出进入「暗杠」状态的期望回合数。

Input

第一行一个整数 n ($4 \leq n \leq 300$)，表示牌的总数。

第二行三个整数 a, b, c ($0 \leq a, b, c \leq n$, $a + b + c = n$)，分别表示初始时你的三种花色的牌的数量。

Output

一个整数，表示期望摸牌的次数，对 998244353 取模。

形式上讲，令 $M = 998244353$ 。可以证明，答案可以表示为一个不可约分的分数 $\frac{P}{Q}$ ，其中 P 和 Q 是整数且 $Q \not\equiv 0 \pmod{M}$ 。输出一个整数 x ($0 \leq x < M$) 满足 $x \cdot Q \equiv P \pmod{M}$ 。

Examples

standard input	standard output
4 2 1 1	249561091
5 5 0 0	0
100 33 33 34	100151198

Note

样例一的期望回合次数是 $\frac{11}{4}$ 。

Problem L. 饿了吧！

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

又到了点外卖的时间了，etilletas 将在 learner 开发的饿了吧 APP 中进行点单。这是一款很纯粹的外卖平台，目前只有一家月饼店，而且不能使用任何优惠券，唯一的优惠是饿了吧平台特有的一项满减活动。

月饼店目前有且仅有 5 款月饼，价格分别是 a_1, a_2, a_3, a_4, a_5 ，作为老饕的我们内心也会对这 5 款月饼有着一套评价价值 b_1, b_2, b_3, b_4, b_5 。

现在需要你帮 etilletas 选择最具性价比的搭配方案（不会不买），因为月饼店刚刚开业，所以对于每种月饼一位顾客至多购买一份，其中，性价比的计算方式为：若干个月饼的评价价值和/若干个月饼满减后的价格和。

该店铺的满减活动为：满 m 元减 k 元，即当顾客订单达到 m 时，实际付款金额可以减去 k 。

Input

第一行包含两个整数 m, k ($1 \leq k < m \leq 200$)，表示满 m 元减 k 元。

第二行五个整数 a_i ($1 \leq a_i \leq 200$)，表示月饼价格。

第三行五个整数 b_i ($1 \leq b_i \leq 200$)，表示 etilletas 对月饼的评价价值。

Output

输出包含一行，代表最高的性价比，结果保留两位小数。

Example

standard input	standard output
5 3 1 2 4 4 2 2 2 1 2 4	4.00