

第三届环球杯



第 15 赛段：成都

2024 年 11 月 2-3 日

该问题集应包含 13 个问题，共 17 页编号。

根据



国际大学生程序设计竞赛（ICPC）

准备和主持





问题 A.箭头一行

时间限制 1 秒 内存限制 1024 兆字节

将 "箭头字符串" 定义为满足以下条件的字符串：

- 字符串的长度至少为 5。
- 字符串以 > 开始，以 >>> 结束。
- 字符串的其余部分仅由 "-" 组成。

例如，>-->>> 和 >--->>> 是有效的箭头字符串，而 >->> 和 >->->> 则不是。

Sauden 给你一个长度为 n 的字符串 s ，由 > 和 - 组成。你需要在一个长度 n 相同、完全由 * 组成的字符串上进行一系列涂画操作，从而创建 s 。在一次绘制操作中，你可以选择一个长度至少为 5 的子串，并将其转换为箭头字符串。您执行的操作总数不能超过 n 。

否则，输出 "是"，并提供绘制操作的详细信息。如果有多个解决方案，则输出任意一个。

输入

第一行包含一个整数 T ($1 \leq T \leq 10^4$)，表示测试用例的数量。

每个测试用例都包含一个长度为 n ($5 \leq n \leq 10^5$) 的字符串 s ，仅由 > 和 - 两行组成。保证所有测试用例的 n 之和不超过 $5 \cdot 10^5$ 。

输出

对于每个测试用例，如果通过不超过 n 次涂画操作无法获得给定字符串，则单行输出 "否"。否则，输出 "是" 和一个正整数 m ($1 \leq m \leq n$)，表示要执行的绘制操作的次数。然后输出 m 行，每行包含两个整数 p ($1 \leq p \leq n-4$) 和 l ($5 \leq l \leq n-p+1$)，表示所选子串的起始位置及其长度。

示例

标准输入	标准输出
4	是 2
>>->>>	1 5
>>>->	2 5
>>>>>	没有
>->>>>>>	没有
	是 2
	2 7
	1 5



备注

对于示例中的第四个测试用例，喷漆过程如下所示：

***** → *>--->>> → >->>>>>>



问题 B.运动员欢迎仪式

时间限制 2.5 秒 内存限
制 1024 兆字节

成都即将举办 2025 年世界运动会。在开幕式的运动员欢迎仪式上，将有 n 名身着三种不同传统民族服装的志愿者列队欢迎运动员。志愿者的位置已经确定，现在我们需要为志愿者分配服装。为了达到特定的视觉效果，相邻的志愿者不能穿同一种服装。

在 n 名志愿者中，有些人已经拥有三种民族服装中的一种，有些人则没有，需要由组织者提供定制服装。有 Q 个定制计划，每个计划都规定：制作 x 套服装 a、 y 套服装 b 和 z 套服装 c。

针对每个定制计划，确定在将定制服装分发给没有任何服装的志愿者后，可以做出多少种不同的有效服装安排。具体来说，在不超出给定计划限制的条件下，确定分配服装 a、b 和 c 的方法的数量。如果分配给同一志愿者的服装类型不同，则认为这两种安排是不同的。请注意，相同类型的服装不会相互区分。由于数字可能很大，请输出答案的模数 $10^9 + 7$ 。

输入

第一行包含两个整数 n ($1 \leq n \leq 300$) 和 Q ($1 \leq Q \leq 10^5$)，分别代表志愿者人数和定制计划数。

第二行是长度为 n 的字符串 s 。如果第 i 个字符是 a、b 和 c 中的一个，则表示第 i 位志愿者已经有了相应的服装；否则，如果是 ?，则表示第 i 位志愿者没有任何服装。

接下来的 Q 行中，每一行都包含三个整数 x, y, z ($0 \leq x, y, z \leq 300$)，代表一个定制计划。保证 $x + y + z$ 的总和不少于不穿服装的志愿者人数。

输出

输出 Q 行，其中第 i 行包含一个整数，表示满足第 i 个定制计划要求的有效服装安排的数量。请输出答案的模数 $10^9 + 7$ 。

实例

标准输入	标准输出
6 3 a? b?? c 2 2 2 1 1 1 1 0 2	3 1 1
6 3 ?????? 2 2 2 2 3 3 3 3 3	30 72 96



备注

在第一个样本中，第一个定制计划的有效服装安排是 acbabc、acbcac 和 acbcbc。



问题 c. 中国象棋

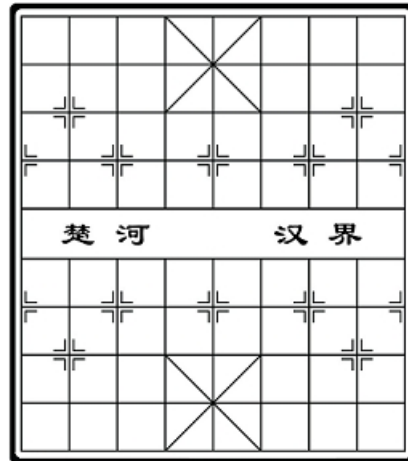
时间限制

1 秒 内存限制

1024 兆字节

这是一个互动问题。

中国象棋是一种双人策略棋类游戏，在中国非常流行。中国象棋棋盘上有 10 行 9 列。我们用 (r, c) 表示从下到上第 r 行和从左到右第 c 列上的点 ($0 \leq r \leq 9, 0 \leq c \leq 8$)。



本题共有六种棋子：王、吏、车、马、象和卒。注意，炮不在考虑之列。此外，所有棋子都可以在棋盘上任意移动，这与原始规则略有不同。每个棋子的移动规则如下（假设当前位置为 (r, c) ）：

国际象棋	符号	一次移动可能出现的位置
国王	J	$(r+1, c) (r-1, c) (r, c+1) (r, c-1)$
普通话	S	$(R+1, C+1) (R+1, C-1) (R-1, C+1) (R-1, C-1)$
路克	C	所有 (r', c') ，使得 $r' = r$ 或 $c' = c$ ，但 (r, c) 本身除外
骑士	M	$(R+2, C+1) (R+2, C-1) (R-2, C+1) (R-2, C-1)$ $(R+1, C+2) (R+1, C-2) (R-1, C+2) (R-1, C-2)$
主教	X	$(R+2, C+2) (R+2, C-2) (R-2, C+2) (R-2, C-2)$
典当	B	$(r+1, c)$ 如果 $r \leq 4$ ，否则 $(r+1, c) (r, c+1) (r, c-1)$

当然，棋子不能移动到棋盘边界之外。既然了解了规则，那我们就来玩一局吧。

我在棋盘上放了一个棋子，你看不到它的位置或类型。不过，我会提供一组 A ，其中包含该棋子可能所在的 n 个位置。你的目标是用最少的查询次数确定棋子的类型。对于每个查询，你都可以选择棋盘上的一个位置，然后我会告诉你棋子移动到该位置所需的最少步数，或者我会告诉你棋子



是否不可能移动到该位置。

有趣吧？更有趣的是，我其实是在暗中与你对弈。说实话，虽然集合 A 是固定的，但我从一开始就没有固定这枚棋子的类型和位置。我的策略是在不违背已经给出的信息的前提下，调整对你每次询问的回答，从而最大限度地增加你的询问次数。在我们的策略都是最优的情况下，你的查询次数 m 其实是可以确定的。我相信聪明的你一定能想出办法。游戏开始

互动协议

首先，你应该从标准输入中读取一个整数 n ($1 \leq n \leq 90$)，表示集合 A 的大小。



然后，读取 n 行。每行包含两个整数 r 和 c ，表示集合 A 中的一个位置。

读完这 $n + 1$ 行后，您应该向标准输出端输出一个整数 m ，表示查询的次数。如果输出的 m 不正确，则会得到 "错误答案 "的提示。然后，您必须准确地进行 m 次查询。

要进行查询，请在一行中输出 `? r c 9, 0 c 8)`。然后从标准输入中读取一个整数，表示棋子到达所选位置所需的最少步数，如果棋子无法到达该位置，则读取 1。如果您的程序进行了无效查询或超出了查询次数，交互程序将立即终止，您将收到 "错误答案 "的提示。

要输出您对棋子类型的猜测，您需要输出 `! ch`，其中 `ch` 表示语句中显示的代表棋子的符号。

在进行查询或输出答案（包括 m 和棋子类型）后，不要忘记输出行结束符（End of Line `\n`）并刷新输出。为此，请使用

- `fflush(stdout)` 或 C++ 中的 `cout.flush()`；
- Java 中的 `System.out.flush()`；
- 在 Python 中使用 `stdout.flush()`；

实例

标准输入	标准输出
1 9 0 6	1 ?3 6 !S
4 2 1 2 3 2 5 2 7 3 1	 2 ?0 0 ?2 0 !J

备注

在第一个例子中，六种棋子从位置（9，0）移动到（3，6）所需的步数如下：

国王 (J)	柑橘 (S)	鲁克 (C)	骑士 (M)	主教 (X)	棋子 (B)
12	6	2	4	3	-1



因此，您只需进行一次查询即可确定答案。



问题 D. 最接近的冲突

时间限制 3 秒 内存限制 1024 兆字节

黑鸟有一个长度为 n 的排列组合 p 。
对于每个 $i = 1, 2, \dots$ 满足 $q_i \neq p_i$ 的另一个长度为 n 的排列 q ， $\sum_{i=1}^n |p_i - q_i|$ 应最小化。满足上述条件的置换 q 称为 p 的最接近错乱。

p 可能有多个最邻近的导差，你的任务是按词典顺序输出 k 个最小的最邻近导差。如果 p 的最接近导差少于 k 个，则输出-1。

长度为 n 的排列指长度为 n 的序列，其中所有元素都是不同的，并且是 1 到 n 之间的正整数。
设 a 和 b 是长度为 n 的两个不同的排列。那么，当且仅当在 $a_i = b_i$ 的最小索引 i 中， $a_i < b_i$ 时， $a < b$ 。

输入

第一行包含一个整数 T ($1 \leq T \leq 10^4$)，表示测试用例的数量。
对于每个测试用例，第一行包含两个正整数 n ($2 \leq n \leq 10^5$) 和 k ($1 \leq k \leq 10^9$)。第二行包含 n 个正整数 p_1, p_2, \dots, p_n ，代表排列组合 p 。
保证所有测试用例的 n 之和不超过 10^6 。

输出

对于每个测试用例，如果至少有 k 个最接近的偏差，则输出 n 个正整数 q_1, q_2, \dots, q_n ，用空格分隔成一行，代表按词典顺序排列的 p 的第 k 个最小最接近的偏差。否则，输出-1。

示例

标准输入	标准输出
2	-1
2 2	3 1 2
2 1	
3 2	
1 2 3	

备注

对于第一个测试用例， $[1, 2]$ 是唯一最接近的错位，因此输出-1。
对于第二个测试用例， $[2, 3, 1]$ 和 $[3, 1, 2]$ 是 p 的最邻近出差，而 $[3, 1, 2]$ 大于按词典顺序排列为 $[2, 3, 1]$ 。



问题 E.中断通信

时间限制 2 秒 内存限制 1024 兆字节

敌人在多个地点建立了通信前哨，这些前哨可以用网络中的节点和边来表示。这个网络形成了一棵树--一种相连且没有循环的图形。作为通信工程专家，您的任务是破坏他们的通信。

每次通信都是沿着树中两个节点之间的简单路径进行的。您可以选择这棵树的一个子图，并中断该子图中的每个节点。如果通信路径包含一个被中断的节点，则通信被成功中断。您选择的子图必须由原始树中的节点和边的子集组成，并且必须是连接的，也就是说它也是一棵树。

通信网络由 n 个节点组成，标号从 1 到 n 。你的任务是回答 q 个不同的查询。对于每个查询，你都会得到两个节点 u_i 和 v_i ，你必须确定可以选择多少个不同的子图来破坏这两个节点之间的通信。由于这个数字可能非常大，因此您应该提供以 998244353 为模数的答案。有可能 $u_i = v_i$ ，这表示节点内部通信，您也可以通过选择包含该节点的子图来中断它。

输入

第一行包含一个整数 T ($1 \leq T \leq 10^4$)，表示测试用例的数量。

每个测试用例的第一行包含两个整数 n ($2 \leq n \leq 10^5$) 和 q ($1 \leq q \leq 10^5$)，分别表示节点数和查询次数。

第二行包含 $n - 1$ 个整数 p_2, p_3, \dots, p_n ($1 \leq p_i < i$)，表示节点 i 和 p_i 通过一条边相连。

接下来的 q 行中，每一行都包含两个整数 u_i, v_i ($1 \leq u_i, v_i \leq n$)，代表第 i 个查询。

保证所有测试用例的 n 和 q 之和分别不超过 $3 \cdot 10^5$ 。

输出

对于每个测试用例，输出 q 行，每行包含 q 个查询之一的结果 modulo 998244353。

示例

标准输入	标准输出
2	6
3 2	5
1 1	14
2 3	13
1 2	15
5 3	
1 1 2 2	
4 5	
2 4	
2 3	



备注

在示例的第一种情况下，总共可以选择 6 个相连的子图。对于第一个查询，所有这些子图都会破坏通信。对于第二个查询，其中 5 个子图可以中断通信，只有节点 3 组成的子图例外。



问题 F. 双 11

时间限制 6 秒 内存限制
1024 兆字节

双十一购物节即将来临，一家商店正在对其畅销产品进行管理。

有 n 种畅销产品，每种产品的日销售量为 s_i 。产品需要多次补货才能满足销售需求，而由于仓库空间有限，总库存不能超过存储容量。

如果每次只补充一种产品，工作量就会过大。为了优化补货，商店决定将产品类型分为 m 组，并为每组分配一个补货参数。对于第 j 组，补货参数 k_j 是一个正实数，这意味着对于该组中的每种产品 i ，每次补货都要准备 $k_j s_i$ ，平均每天补货 $\frac{1}{k_j}$ 次。请注意， k_j 可以大于或小于或等于 1。

假设 c_i 是第 i 类产品的组指数。仓库中的最大库存量可以表示为 $\sum_{i=1}^n k_{c_i} \cdot s_i$ 。由于仓库容量限制，该值不能超过 1。

你的任务是找到一种方案，将 n 种产品分成 m 组，并为每组分配一个补货参数，使每天的补货总数最小，同时满足以下条件

仓库容量限制，即最小化 $\sum_{i=1}^n \frac{1}{k_{c_i}}$ ，条件是 $\sum_{i=1}^n k_{c_i} \cdot s_i \leq 1$ 。为了方便起见，你应该输出最小答案的平方根，即 $\sqrt{\sum_{i=1}^n \frac{1}{k_{c_i}}}$ 。请注意，您可以将相同的补货参数分配给不同的组。

输入

第一行包含两个整数 n 和 m ($1 \leq n \leq 2 \cdot 10^5$)，表示产品类型数和组数。

第二行包含 n 个整数 s_i ($1 \leq s_i \leq 10^5$)，表示第 i 种产品的销售量。

输出

输出一个实数，表示答案。如果相对误差或绝对误差不超过 10^{-9} ，您的输出将被视为正确。

实例

标准输入	标准输出
4 2 1 2 3 4	6.1911471295571
10 3 1 2 3 4 5 6 7 8 9 10	22.5916253665141

备注

在第一个例子中，让 $k_1 = \frac{1}{2}$, $k_2 = \frac{1}{2}$, 和 $c = c = 1, c = c = 2$ 。即， $c_1 = c_2 = 1, c_3 = c_4 = 2$ 。我们将得到最大值



库存为

$(1 + 2)k_1 + (3 + 4)k_2 = 1$

因此，答案是 $\sqrt{20 + 4\sqrt{21}} \approx 6.1911471295571$ ，这是最小值。

$3 + 21$

$7 + 21$

$\frac{2}{\cdot}$

$\frac{2}{\cdot}$

$\sqrt{\quad}$



问题 G. 扩展阵列

时间限制 2 秒 内存限制
1024 兆字节

给定一个整数数组 a_1, a_2, \dots, a_n 的长度为 n ，可以对这个数组执行任意数量的操作。在每次操作中，你可以选择两个相邻的元素 a_i 和 a_{i+1} ($1 \leq i < n$)，并在它们之间插入以下三个值中的一个： a_i 和 a_{i+1} ， a_i 或 a_{i+1} ，或 $a_i a_{i+1}$ 。你的任务是确定在执行任意次数的操作后，数组中可能存在的最大不同值的个数。

注： x 和 y 表示 x 和 y 的位和。 x 或 y 表示 x 和 y 的位 OR。
表示 x 和 y 的比特 XOR（排他 OR）。

输入

第一行包含一个整数 n ($2 \leq n \leq 10^5$)，代表数组的长度。
第二行包含 n 个整数 a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$)，表示数组的元素。

输出

输出一个整数，表示数组在执行任意次数的运算后可获得的最大不同值个数。

实例

标准输入	标准输出
2 2 3	4
2 3 4	4



问题 H. 友谊是神奇的

时间限制 1 秒 内存限制
1024 兆字节

洛克杜是一匹住在小马镇的小马。他最好的朋友马卡龙林也住在那里。Rockdu 非常重视友谊，他和 Macaronlin 分享一切，甚至整数。

问题来了：如何与他人共享一个整数？对于整数 x ，Rockdu 会将其拆分成两部分。具体来说，他将十进制形式的 x 视为一个不含前导零的字符串，并在某个位置将其拆分成两个非空的子串。然后，他将这两个子串视为两个独立的十进制整数，分别表示为 x_1 （前者）和 x_2 （后者）。

Rockdu 希望 x_1 和 x_2 的值越接近越好。因此，在所有可能的拆分方法中，他会选择能使 x_1 和 x_2 之间的绝对差值最小的方法。例如，如果 $x = 1003$ ，有三种可能的拆分方法：1|003、10|03 和 100|3：1|003、10|03 和 100|3。Rockdu 选择第一种方式，因为它产生的绝对差值最小： $|1 - 3| = 2$ ，而其他方法的结果分别是 $|10 - 3| = 7$ 和 $|100 - 3| = 97$ 。

让 $f(x)$ 被定义为两个整数分割后的最小绝对差值。
例如， $f(1003) = 2$ 。给定两个整数 l 和 r ，Rockdu 希望计算在 $[l, r]$ 范围内所有 i 的 $f(i)$ 之和。由于答案可能非常大，请输出它的模数 $10^9 + 7$ 。

输入

第一行包含一个整数 T ($1 \leq T \leq 1000$)，表示测试用例的数量。每个测试用例在一行中包含两个整数 l, r ($10 \leq l \leq r \leq 10^{18}$)。

输出

对于每个测试用例，请在一行中输出答案 modulo $10^9 + 7$ 。

示例

标准输入	标准输出
2 108 112 114514 1919810	31 86328270

备注

对于样本中的第一个测试用例：

- $f(108) = \min(|1 - 8|, |10 - 8|) = \min(7, 2) = 2$
- $f(109) = \min(|1 - 9|, |10 - 9|) = \min(8, 1) = 1$



- $f(110) = \min(|1 - 10|, |11 - 0|) = \min(9, 11) = 9$
- $f(111) = \min(|1 - 11|, |11 - 1|) = \min(10, 10) = 10$
- $f(112) = \min(|1 - 12|, |11 - 2|) = \min(11, 9) = 9$

因此， $\sum_{i=108}^{112} f(i) = 2 + 1 + 9 + 10 + 9 = 31$ 。



问题 I. 良好分区

时间限制 1 秒 内存限制 1024 兆字节

Lawliet 有一串长度为 n 的数字，分别表示为 a_1, a_2, \dots, a_n ，他想知道有多少个好的分区存在。如果分区大小 k 满足 $1 \leq k \leq n$ ，且按分区大小将序列 a 分成若干部分后，得到的每个子序列都不递减，则该分区大小 k 被认为是一个好的分区大小。分割方法如下：

- 序列 a 被分为 $\lceil \frac{n}{k} \rceil$ 部分。
- 对于第 i 部分 ($1 \leq i \leq \lceil \frac{n}{k} \rceil - 1$)，元素为 $a_{(i-1) \times k + 1}, a_{(i-1) \times k + 2}, \dots, a_{i \times k}$
- 对于 $\lceil \frac{n}{k} \rceil$ -th 部分，元素为 $a_{(\lceil \frac{n}{k} \rceil - 1) \times k + 1}, \dots, a_n$ 。请注意，最后一部分的长度为可能小于 k 。

Lawliet 发现这个问题太简单了，所以他要做 q 项修改。每次修改都提供两个正整数 p 和 v ，表示 a_p 的值将改为 v 。

您的任务是帮助 Lawliet 计算出修改前和修改后良好分区大小的数量。

输入

第一行包含一个整数 T ($1 \leq T \leq 10$)，表示测试用例的数量。

对于每个测试用例，第一行包含两个整数 n ($1 \leq n \leq 2 \cdot 10^5$) 和 q ($1 \leq q \leq 2 \cdot 10^5$)，分别代表序列长度和修改次数。

第二行包含 n 个整数，代表序列 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 2 \cdot 10^9$)。

下面的 q 行分别包含两个整数 p ($1 \leq p \leq n$) 和 v ($1 \leq v \leq 2 \cdot 10^9$)，表示序列中位于第 p 个位置的元素将被修改为 v 。

保证所有测试用例的 n 和 q 之和分别不超过 $2 \cdot 10^5$ 。

输出

对于每个测试用例，输出 $q + 1$ 行，代表修改前和修改后良好分区大小的数量。

示例

标准输入	标准输出
1	1
5 2	2
4 3 2 6 1	3
2 5	



3 5	
-----	--

备注

最初，唯一合适的分区大小是 $k = 1$ 。

第一次修改后，序列变为 $[4, 5, 2, 6, 1]$ 。 $k = 1$ 和 $k = 2$ 都是很好的分区大小。

第二次修改后，序列变为 $[4, 5, 5, 6, 1]$ 。好的分区大小为 $k = 1$ 、
 $k = 2$ 和 $k = 4$ 。



问题 J. 巴朗斯大奖赛

时间限制

2 秒 内存限制

1024 兆字节

Ballance 是一款经典游戏，玩家通过键盘控制小球在高空复杂的结构中穿行，在解开谜题的同时避免坠落，最终到达每个关卡的终点。最近，玩家社区开发了在线 MOD，并定期举办在线比赛，如 Ballance 大奖赛。

大奖赛由 n 个关卡（从 1 到 n ）和 m 名参赛者（从 1 到 m ）组成。比赛采用积分制：参赛者根据在每个关卡中的排名获得积分，所有关卡的积分总和决定最终排名。每个关卡都有指定的开始时间，参赛者必须尽快完成关卡。作为工作人员，您会在比赛期间收到服务器日志，其中包含三种信息（保证 $1 \leq x \leq n$ 且 $1 \leq id \leq m$ ）：

- 1 x - 类型 1：第 x 层的比赛已经开始。
- 2 id x - 类型 2：参与者索引 id 已完成第 x 级。
- 3 id x - 类型 3：参与者索引 id 自愿放弃完成第 x 级。

类型 1 报文表示第 x 级开始，直到出现不同级别的新类型 1 报文。只有与当前活动级别相对应的信息才有效，其他级别的所有信息都应忽略。第一条 1 类报文之前的报文也将被忽略。每个级别在 1 类报文中最多出现一次。

每个玩家在每个关卡都可以发出多条 2 类和 3 类信息，但只有第一条有效信息才算数。具体来说

- 如果报文与类型 1 报文指示的活动级别不匹配，则会被忽略。
- 如果玩家在某个关卡中的第一条有效信息是类型 2，那么他在那一刻就会被视为成功完成了该关卡，玩家在该关卡中的任何后续信息都会被忽略。
- 如果玩家在某个关卡中的第一条有效信息是第 3 类，则会被视为放弃完成该关卡，玩家在该关卡中的任何后续信息都会被忽略。
- 如果玩家在某一关卡中没有收到任何信息，则视为未完成该关卡。

完成关卡的参赛者可获得如下积分：第一个完成关卡的参赛者可获得 m 分，第二个完成关卡的参赛者可获得 $m-1$ 分，以此类推。未完成关卡的参赛者，包括放弃的参赛者，不得分。

您的任务是根据日志，按降序输出每位参与者当前的总积分。如果多个参赛者的积分相同，则应按指数升序排列。



输入

第一行包含一个整数 T ($1 \leq T \leq 10^4$)，表示测试用例的数量。

对于每个测试用例，第一行包含三个整数 n 、 m 和 q ($1 \leq n \leq 10^5$ ， $2 \leq m \leq 10^5$ ， $1 \leq q \leq 2 \cdot 10$)，分别表示级别数、参与者数和日志信息数。

以下 q 行包含上述指定的日志信息。当然，这些信息是按时间顺序排列的。日志可能不会包含所有级别，因为您可能会在比赛中途收到日志。您只需处理当前的结果。

保证所有测试用例的 n 、 m 和 q 之和分别不超过 $5 \cdot 10^5$ 。



输出

对于每个测试用例，输出 m 行，每行包含两个整数： id 和 x ，其中 id 是参与者的索引， x 是他们的总积分，按积分降序排列。如果多个参与者的分数相同，则按指数升序排列。

示例

标准输入	标准输出
3	2 4
3 4 6	1 3
1 2	3 0
2 1 1	4 0
2 2 2	1 7
3 3 2	2 4
2 3 2	3 0
2 1 2	4 0
3 4 8	2 4
1 2	1 3
2 1 1	3 0
2 2 2	4 0
3 3 2	
2 3 2	
2 1 2	
1 1	
2 1 1	
3 4 7	
1 2	
2 1 1	
2 2 2	
3 3 2	
2 3 2	
2 1 2	
1 1	



问题 K. 神奇套装

时间限制 1 秒 内存限制
1024 兆字节

你有一个神奇的集合，最初包含 n 个不同的整数。你发现这些数字可以通过除以它们的因数来产生能量。在每一步中，你都可以从集合中选择任何一个大于 1 的数字，移除它，然后插入它的一个因数。你插入的因数必须不等于原来的数字。此外，由于魔法集合的不稳定性，您的操作必须确保集合中的数字保持不同。

每次操作都会产生一个单位的能量，你的目标是通过执行尽可能多的操作，使产生的总能量最大化。给定集合中的初始数字，确定可产生的最大能量，即可执行的最大操作次数。

输入

第一行包含一个整数 n ($1 \leq n \leq 300$)，表示初始集合中的整数个数。第二行包含 n 个不同的整数 a_i ($1 \leq a_i \leq 10^9$)，表示初始集合中的数字。

输出

输出一个整数，表示可产生的最大能量，即可执行的最大操作次数。

实例

标准输入	标准输出
3 2 4 6	3
6 2 3 5 6 10 12	3



问题 L.恢复统计数据

时间限制 1 秒 内存限制
1024 兆字节

你们最近对大学生从宿舍到教学楼的通勤时间进行了调查。您认为这项调查可以极大地改善校园规划，使师生的通勤更加方便。作为分析的一部分，您计算了 P50、P95 和 P99 通勤时间，以支持您的结论。例如，集合 1, 1, 4, 5, 1, 4 的 P50 可以是 1、2 或 3，因为正好有 6 50% = 3 个值小于或等于 y 。 { } ×
为 1、2 或 3。但是，这些值没有有效的 P95 或 P99，因为 6 95% 和 6 99% 不是整数。 ×
不幸的是，出错了--您不小心删除了整个数据集。您只剩下通勤时间的 P50、P95 和 P99 值。由于没有时间重做调查，您需要重建一组与所有 P50、P95 和 P99 值相匹配的数据。

输入

输入由三行组成。第一行包含一个整数 a ，代表 P50 值。第二行包含一个整数 b ，代表 P95 值。第三行包含一个整数 c ，代表 P99 值。 $(1 \leq a < b < c < 10)^9$

输出

输出两行。第一行应包含一个整数 n ($100 \leq n \leq 10^5$)，代表数据集的长度。
第二行应包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$)，表示重建的数据集。整数的顺序不限。任何符合要求的输出都将被认为是正确的。

示例

标准输入	标准输出
50 95 99	100 1 2 ...100

备注

为便于显示，示例输出省略了 3 到 99 之间的值。在您的输出中，必须包含所有这些值。
请注意，本问题中 P_x 值的定义不同于 x 百分位数。



问题 M. 两个凸孔

时间限制

5 秒 内存限制

1024 兆字节

考虑三维直角坐标系中的两个不透明平面 $z = z_1$ 和 $z = z_2$ ($z_1 < z_2$)。每个平面都有一个凸多边形孔，分别记为 P_1 和 P_2 ，它允许光线通过。

有一个点光源 L 在平面 $z = z_0$ ($z_2 < z_0$) 上沿平行于 xOy 平面的方向移动。光源从 $t = 0$ 时的初始位置 (x_0, y_0, z_0) 开始，以恒定速度 $\mathbf{v} = (v_x, v_y, 0)$ 移动。因此，在任意时间 t ，光源的位置由 $(x_0 + v_x t, y_0 + v_y t, z_0)$ 给出。

对于平面 $z = 0$ 中的一个点 A ，当且仅当线段 LA 与多边形 P_1 和 P_2 的内部（包括边界）相交时，将其定义为在 t 时刻**被照亮**。 t 时刻的**照亮区域**（记为 $f(t)$ ）是平面 $z = 0$ 中所有被照亮的点所形成的区域。

假定 t 是 $[t_1, t_2]$ 上均匀分布的随机变量，将 $[t_1, t_2]$ 时间段内的**平均照明面积**定义为 $E[f(t) \mid t \in [t_1, t_2]]$ ，即 $f(t)$ 在 $[t_1, t_2]$ 时间段内的期望值。

给定多个时间段 $[t_1, t_2]$ ，你的任务是找出每个时间段的平均照明面积。

输入

第一行包含一个整数 T ($1 \leq T \leq 10^4$)，表示测试用例的数量。

对于每个测试用例，第一行包含五个整数 x_0, y_0, z_0, v_x, v_y ($-10^5 \leq x_0, y_0 \leq 10^5, 1 \leq z_0 \leq 10, ^5$

$-10^3 \leq v_x, v_y \leq 10^3$)，表示光源的初始位置 (x_0, y_0, z_0) 及其速度矢量

$\mathbf{v} = (v_x, v_y, 0)$ 。保证 $\mathbf{v} \neq (0, 0, 0)$ 。

第二行包含两个整数 n_1 和 z_1 ($3 \leq n_1 \leq 10^5, 1 \leq z_1 \leq 10^5$)，表示多边形 P_1 的顶点数和 z_1 的值。接下来的 n_1 每行包含两个整数 x_i 和 y_i ($-10^5 \leq x_i, y_i \leq 10^5$)，按逆时针顺序描述 P_1 的顶点。

下面一行包含两个整数 n_2 和 z_2 ($3 \leq n_2 \leq 10^5, 1 \leq z_2 \leq 10^5$)，表示多边形 P_2 的顶点数和 z_2 的值。下面 n_2 每行包含两个整数 x_j 和 y_j ($-10^5 \leq x_j, y_j \leq 10^5$)，按逆时针顺序描述多边形 P_2 的顶点。

保证 P_1 和 P_2 没有三个或三个以上的顶点相交。

下面一行包含一个整数 q ($1 \leq q \leq 10^5$)，表示查询次数。下面 q 行中的每一行都包含两个整数 t_1 和 t_2 ($0 \leq t_1 \leq t_2 \leq 10^3$)，表示一个时间段。

保证所有测试用例的 $n_1 + n_2$ 和 q 的总和分别不超过 10^5 ，且 $z_1 < z_2 < z_0$ 。



输出

对于每个查询，输出一个代表平均照明面积的实数。只有当您的答案与正确答案之间的相对误差或绝对误差不超过 10^{-4} 时，您的答案才会被认为是正确的。



示例

标准输入	标准输出
1	0.450000000
0 0 3 0 -1	1.125000000
4 1	2.250000000
1 0	
3 0	
3 2	
1 2	
4 2	
0 0	
1 0	
1 1	
0 1	
3	
0 10	
1 2	
1 1	

备注

在本例中，凸多边形 P_1 和 P_2 在 $t=0$ 时在 xOy 平面上的投影以及这些投影的移动如下图所示。多边形 $A B C_{1111} D$ 是多边形 P_1 的投影，多边形 $A B C_{2222} D$ 是多边形 P_2 的投影。

