

Problem A. 购买饮料

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

在偶像大师OFA中，存在一类剧情，在live结束后，我们(producer)要给小偶像买饮料，买的饮料会影响到小偶像的心情，但是作为新手，你并不知道哪些饮料会提高小偶像的心情，不过好在这类剧情可能多次触发，你可以通过试错法来确定某些小偶像喜欢哪种饮料，每位小偶像只喜欢一种饮料。

假设有 a 位不同的小偶像和 b 种饮料，存在 n 个剧情 s_n ， s_i 表示是第 s_i 位小偶像的剧情，现在给定 q 次查询，每次查询一个区间 $[l, r]$ 的剧情，按照顺序进行剧情，最坏情况下最多可以提升多少次小偶像的心情？每次查询相互独立，你在第一次查询所得到的的小偶像喜欢喝的饮料在第二次查询中是不知道的。

Input

第一行包含四个整数 $a(1 \leq a \leq 2 \times 10^5)$, $b(1 \leq b \leq 2 \times 10^5)$, $n(1 \leq n \leq 2 \times 10^5)$, $q(1 \leq q \leq 2 \times 10^5)$ ，分别表示小偶像的数量、饮料的数量、剧情的数量和查询的数量。接下来一行包含 n 个整数，表示每个剧情中小偶像的编号 $s_i(1 \leq s_i \leq a)$ 。接下来 q 行，每行包含两个整数 l 和 r ，表示查询的区间

Output

对于每次查询，输出一行，每一行包含一个整数，表示该次查询下最坏情况下最多可以提升多少次小偶像的心情。

Example

standard input	standard output
3 3 7 2	2
1 2 3 1 1 2 1	0
1 7	
1 4	

Note

对于第一次查询，我们最坏情况下可以在第5个剧情知道小偶像1喜欢喝的饮料提升两次心情，对于第二次查询我们最坏情况我们一次也无法提升心情

Problem B. 仿生曜灵不会梦到电子鱼

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

正月为曜灵购买了一个超大的鱼缸，曜灵非常开心，决定往里面添加很多的鱼类。但由于黑潮之下的影响，曜灵想要买的鱼全部被核废水杀死了，她找到了璃寒，想要替代方案，而璃寒则给出了一个非常恰当的方式——使用电子鱼来替代真正的鱼。

拴不住的 AI 会乱跑，电子鱼也是一样，璃寒推销了与电子鱼同款的监视器，可以监测电子鱼的活动情况。为方便控制且美观，共有 N 个监视器被部署在被抽象为二维平面的鱼缸中，第 i 个监视器的坐标可以表示为 (X_i, Y_i) ，大小合计为 M 的电子鱼群在这个鱼缸中游动，第 j 条鱼的坐标也可以表示为 (x_j, y_j) ，为了保持整体的美观，这个鱼群的移动方式非常特殊，他们会收到共 Q 个向量指令，第 k 条指令可以视为一个 (vx_k, vy_k) 的向量，鱼群中的所有鱼都会向该向量方向移动，并在移动之后保持移动后的位置，随后接受下一个指令直到所有指令处理完成。

根据璃寒的推断，监视器能够在其他监视器的帮助下监视一个给定区域下的电子鱼，具体地，任意三个监视器组成的三角形（包括边界）可以用于监测这个三角形内的所有电子鱼，鱼群在每次移动过后，监视器都会判断一次鱼群是否能够被全部监测到。在每次移动后，如果能够被全部监测，输出 "Catch"，否则输出 "Escape"（均不含双引号）。

Input

第一行两个整数 N 和 M ，分别表示监视器和电子鱼的点集大小。接下来 N 行，每行两个整数 (X_i, Y_i) ，表示第 i 个监视器的坐标。接下来 M 行，每行两个整数 (x_i, y_i) ，表示第 i 个电子鱼的坐标。接下来一行一个整数 Q ，表示移动的指令数。接下来 Q 行，每行两个整数 vx_i, vy_i ，表示第 i 个指令代表的向量。

Output

共 Q 行，每行一个字符串，如题意所示。

Example

standard input	standard output
4 3	Catch
-2 -2	Escape
2 -2	Escape
2 2	Catch
-2 2	Catch
0 0	
1 0	
0 1	
5	
0 0	
4 0	
-4 4	
0 -4	
-2 -2	

Note

对于全部数据， $3 \leq N, M, Q \leq 10^5$ ， $-10^5 \leq X_i, Y_i, x_j, y_j \leq 10^5$ ， $-2 \times 10^5 \leq vx_k, vy_k \leq 2 \times 10^5$ ，保证初始状态下所有电子鱼都能够分别被至少一组监视器检测到。

Problem C. "袋鼠号"上的排班表

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

Yaimsea 买了一艘用来运输散装谷物的新船，起名叫做"袋鼠号"！

"袋鼠号"即将穿越一段繁忙海域。在这艘船上有一名叫做 lfw 的实习生，他并不承担独立航行值班任务，而是主要负责甲板巡查、整理工具、协助靠离泊准备、清洁维护等辅助工作。根据相关国际公约，lfw 的作息安排不能过度疲劳，需要符合船员休息时间的基本要求。

根据 STCW 国际公约，船员的休息安排需要满足：

- 1、任意连续 24 小时内，休息时间不少于 10 小时；
- 2、任意连续 168 小时内，休息时间不少于 77 小时；
- 3、任意连续 24 小时内，休息时间最多只能分成 2 个时间段，并且其中至少有 1 个休息时间段不少于 6 小时。

现在，船上的记录系统导出了一段 lfw 的工作休息记录，在这份记录里，时间按小时离散化。

每个小时的状态只有两种：

T：该小时正在休息；

F：该小时正在进行工作。

给定一个只包含 T、F 的字符串 S ，字符串 S 的第 i 个字符表示记录开始后的第 i 个小时状态，不考虑开始记录前的状态。

你需要从第 1 个小时开始，依次检查每个小时结束时，当前已经完整出现的 24 小时窗口和 168 小时窗口是否满足要求。

如果整份记录都符合要求，输出 **Yes**。

如果某个小时结束时首次发现不符合要求，输出 **No**，并输出首次不合规发生在记录开始后的第几个小时。

Input

输入第一行一个整数 n ，表示字符串 S 的长度。 $(1 \leq n \leq 10^5)$

输入第二行一个只包含 T 和 F 的字符串 S ，表示记录开始后的工作休息状态。

Output

如果整份记录符合要求，输出一行一个字符串 **Yes**，如果不符合要求，先输出一行一个字符串 **No**，然后在输出一个整数 x ，表示从记录开始后的第 x 个小时结束时，首次检测到不合规。注意：你输出的 **Yes/No** 的大小写必须严格匹配。

Examples

standard input	standard output
24 TTTTTTFFFFFFFFTTTTFFFFFFFF	Yes
24 TTTTTTFFFFFFFFFFFFFFFFTTTT	No 24
见下发的3.in	见下发的3.ans
24 TTTTTTFTTFTTFFFFFFFFFFFFFF	No 24
24 TTTTTTFFFFTTTTTTFFFFFFFF	No 24
见下发的6.in	见下发的6.ans

Note

注意：样例 3 与样例 6 输入输出较长，因此内容放到了下发文件 3.in、3.ans、6.in、6.ans 中。

样例 1 中，唯一完整的 24 小时窗口包含 10 小时休息，休息被分成 2 段，且存在一段连续 6 小时的休息，因此符合要求。

样例 2 中，第 24 小时结束时，唯一完整的 24 小时窗口只有 9 小时休息，首次违反"任意连续 24 小时内，休息时间不少于 10 小时"的要求。

样例 3 中，所有完整的 24 小时窗口均满足要求，但第 168 小时结束时，唯一完整的 168 小时窗口只有 70 小时休息，首次违反"任意连续 168 小时内，休息时间不少于 77 小时"的要求。

样例 4 中，第 24 小时结束时，唯一完整的 24 小时窗口虽然包含 10 小时休息，且存在连续 6 小时的休息，但休息被分成了 3 段，首次违反"任意连续 24 小时内，休息时间最多只能分成 2 个时间段"的要求。

样例 5 中，第 24 小时结束时，唯一完整的 24 小时窗口虽然包含 10 小时休息，且休息被分成 2 段，但每段连续休息都只有 5 小时，首次违反"其中至少有 1 个休息时间段不少于 6 小时"的要求。

样例 6 中，虽然整段记录长度为 168 小时，但第 24 小时结束时就已经首次发现不合规，因此应输出 24，而不是继续输出 168。第一个完整的 24 小时窗口包含 10 小时休息，且存在连续 8 小时的休息，但休息被分成了 3 段。

Problem D. 括号问题

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

给定一个长度为 n 的括号字符串 s 。字符串中每个位置 i 都有一个整数值 a_i 。

你需要选择 s 的一个子序列，使得这个子序列是一个合法括号序列，并最大化所选位置的价值和。

本文中的子序列指从原字符串中删除任意若干个字符后，保持剩余字符相对顺序不变得到的字符串。它不要求连续。形式化地，你可以选择若干个下标 $p_1, p_2, \dots, p_k$ ，满足 $1 \leq p_1 < p_2 < \dots < p_k \leq n$ ，得到字符串 $s_{p_1} s_{p_2} \dots s_{p_k}$ 。

一个括号序列是合法的，当且仅当从左到右扫描时，任意前缀中左括号数量都不少于右括号数量，并且整个序列中左括号数量等于右括号数量。

可以选择空子序列，空子序列的价值和为 0。

Input

第一行包含一个整数 n 。

第二行包含一个长度为 n 的字符串 s ，仅由字符 (和) 组成。

第三行包含 n 个整数 $a_1, a_2, \dots, a_n$ 。

约束如下：

- $1 \leq n \leq 200000$;
- s_i 为 (或);
- $-10^9 \leq a_i \leq 10^9$ 。

Output

输出一个整数，表示合法括号子序列的最大价值和。

Examples

standard input	standard output
6 () (()) 1 2 3 4 5 6	21
4 ()) (5 -100 7 8	12
2 (-5 -7	0

Problem E. 成语接龙

Input file: **standard input**
Output file: **standard output**
Time limit: 8 seconds
Memory limit: 1024 megabytes

坎格鲁斯普雷最近正在研究一种特殊的"成语接龙"规则。

在袋鼠语言中，所有的字符都可以用 1 到 m 之间的整数来表示。一个"成语"本质上是一个整数序列。现在一共有 n 个成语，第 i 个成语为序列 s_i 。

坎格鲁斯普雷想利用这些成语构造一个新的序列 t 。

初始时，序列 t 为空。之后，他会不断重复以下操作：

- 选择一个成语 s_i ；
- 将这个成语接到当前序列 t 的末尾。

但加入成语时需要满足特殊规则：

- 若当前序列 t 为空，则可以直接将整个成语加入；
- 否则，必须存在一个长度大于 0 的序列 p ，满足：
 - p 是 t 的一个后缀²；
 - p 是 s_i 的一个前缀¹。

此时，会将这个前缀 p 视为"重叠部分"，只保留一次。

也就是说：

- 不会重复加入这个前缀；
- 只会将 s_i 去掉前缀 p 后的剩余部分接到 t 的末尾。

形式化地，若：

$$s_i = p + q$$

则新的序列会变为：

$$t + q$$

如果存在多个合法成语或多个合法的重叠前缀，坎格鲁斯普雷可以任意选择。

坎格鲁斯普雷想知道，最终能够得到的序列 t 的最大长度是多少。如果长度可以无限增长，请输出 **Infinity**。

¹ 前缀：指一个序列从第一个元素开始，连续包含若干个元素所形成的子序列。例如，序列 $[1, 2, 3]$ 的非空前缀有 $[1]$ ， $[1, 2]$ ， $[1, 2, 3]$ 。

² 后缀：指一个序列从某一个元素开始，连续包含到最后一个元素所形成的子序列。例如，序列 $[1, 2, 3]$ 的非空后缀有 $[3]$ ， $[2, 3]$ ， $[1, 2, 3]$ 。

Input

本题包含多组测试数据。

输入第一行包含一个整数 T ，表示数据组数。 $(1 \leq T \leq 2 \times 10^5)$

对于每组测试数据：

第一行包含两个整数 n 和 m ，分别表示成语的数量和字符集的大小。 $(1 \leq n \leq 2 \times 10^5, 1 \leq m \leq 10^6)$

接下来 n 行，第 i 行首先包含一个整数 l_i ，表示第 i 个成语的长度；接下来包含 l_i 个由空格隔开的整数 $s_{i,j}$ ，表示该成语的字符序列。 $(1 \leq l_i \leq 10^6, 1 \leq s_{i,j} \leq m)$

保证所有测试数据的 n 之和不超过 2×10^5 ，所有测试数据的 $\sum l_i$ 之和不超过 10^6 。

Output

对于每组测试数据：

- 若最大长度有限，输出一个整数表示最大长度；
- 否则输出 **Infinity**。

Example

standard input	standard output
2	7
2 7	Infinity
4 1 2 3 4	
5 3 4 5 6 7	
2 2	
2 1 2	
2 2 1	

Note

对于第一组数据：

初始时选择第一个成语：

$$t = [1, 2, 3, 4]$$

随后选择第二个成语：

$$[3, 4, 5, 6, 7]$$

由于序列：

$$[3, 4]$$

既是当前序列 t 的后缀，也是第二个成语的前缀，因此可以进行拼接。

拼接后得到：

$$t = [1, 2, 3, 4, 5, 6, 7]$$

最终长度为 7。

对于第二组数据：

由于存在 $[1, 2]$ 和 $[2, 1]$ ，可以不断利用公共的 1 或 2 作为重叠部分进行如下拼接：

$$[1, 2] \rightarrow [1, 2, 1] \rightarrow [1, 2, 1, 2] \rightarrow [1, 2, 1, 2, 1] \rightarrow \dots$$

因此长度可以无限增长，答案为 **Infinity**。

Problem F. 海克斯宝石

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

猫头鹰手头上有 $2n$ 个蕴含能量的海克斯宝石，它们的初始能量值记为一个长度为 $2n$ 的数组 a 。

为了启动海克斯飞门，猫头鹰需要将这 $2n$ 个魔法石两两配对，形成 n 个能量对。我们将配对后的结果记为数组 b ，其中第 i 个能量对表示为一个二元组 (x_i, y_i) ($1 \leq i \leq n$)。

法阵的共振规则非常严格：对于数组 b 中的第 i 个能量对 (x_i, y_i) ，它们能量值的差的绝对值必须能够被 $(2i - 1)$ 整除。用数学语言表述即为：

$$(2i - 1) \mid |x_i - y_i|$$

请你帮猫头鹰判断，是否存在至少一种合法的配对方案。如果存在，请输出 **YES** 并给出任意一种合法的配对方案；如果无论如何都无法完成配对，请输出 **NO**。

Input

第一行包含一个整数 n ($1 \leq n \leq 1000$)，表示需要的能量对数量（魔法石总数为 $2n$ ）。第二行包含 $2n$ 个整数 $a_1, a_2, \dots, a_{2n}$ ($0 \leq a_i \leq 10^9$)，表示每个魔法石的能量值。

Output

如果存在合法的配对方案，第一行输出 **YES**。随后输出 n 行，第 i 行 ($1 \leq i \leq n$) 包含两个用空格隔开的整数 x_i 和 y_i ，表示你为第 i 个位置安排的能量对。（如果有多组解，输出任意一组即可；同一对内的两个数字输出顺序不限）。

如果不存在合法的配对方案，仅输出一行 **NO**。

Examples

standard input	standard output
2 10 20 30 40	YES 20 30 10 40
3 1 2 4 5 9 14	YES 1 14 2 5 4 9

Note

对于样例 1，共有 $n = 2$ 个能量对。

- 第 1 个能量对 ($i = 1$)，我们需要满足 $(2 \times 1 - 1) = 1$ 整除 $|x_1 - y_1|$ 。选择 $(20, 30)$ ，满足 $1 \mid |20 - 30|$ 。
- 第 2 个能量对 ($i = 2$)，我们需要满足 $(2 \times 2 - 1) = 3$ 整除 $|x_2 - y_2|$ 。选择 $(10, 40)$ ，满足 $3 \mid |10 - 40|$ 。

对于样例 2，共有 $n = 3$ 个能量对。

- 第 1 个能量对，选择 $(1, 14)$ ，满足 $1 \mid |1 - 14|$ 。
- 第 2 个能量对，选择 $(2, 5)$ ，满足 $3 \mid |2 - 5|$ 。
- 第 3 个能量对，选择 $(4, 9)$ ，满足 $5 \mid |4 - 9|$ 。

Problem G. 计算机系统能力实训

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

这是一道交互题。

猫头鹰这学期有一门叫做"计算机系统能力实训"的课，他很不幸地抽到了一个 FPGA 的题目。面对开发板上密密麻麻的排线，复杂的接线逻辑对他来说实在太难了！于是他许愿能够得到一个能自动理清线路的魔法道具。

第二天，他的桌面上真的出现了一个神秘的"黑盒测试仪"！测试仪上有 n 个输入管脚（编号 $1 \sim n$ ）和 m 盏 LED 指示灯（ $1 \leq m < n$ ）。已知每个输入管脚都在黑盒内部隐秘地连接着恰好一盏 LED 灯。由于管脚的数量大于 LED 灯的数量，黑盒中必定至少有两个管脚连接到了同一盏 LED 灯，猫头鹰的任务就是找出任意一对这样的管脚。（注意：如果有多于两个的管脚连接到了同一盏灯，你只需找出并输出其中任意两个管脚即可。）

为了方便测试，黑盒内部的电路经过了特殊设计：

1. 初始状态下，所有 m 盏 LED 灯均为亮起状态。
2. 所有的输入管脚在内部都连接着一个反相器（逻辑非门）。

由于无法拆开黑盒，猫头鹰只能通过发送测试信号来进行排查。每次测试，你可以指定一个连续的管脚区间 $[L, R]$ ，黑盒会向该区间内的所有管脚发送一个脉冲信号。收到信号后，管脚连接的反相器会被激活，其连接的 LED 灯的状态就会发生一次反转（亮变暗，暗变亮）。并且黑盒测试仪的内部状态是持久化的，这意味着每次发送测试信号后，LED 灯的状态会被保留，你每一次指定的测试区间，都是在当前灯的最新状态基础上进行反转的。

注意：如果区间 $[L, R]$ 内有 k 个管脚连接到了同一盏 LED 灯，那么这盏灯就会连续反转 k 次（例如，若有 2 个管脚连向同一盏灯，它会经历 亮 $\rightarrow$ 暗 $\rightarrow$ 亮，最终状态保持不变）。

在每次测试信号发送完毕后，黑盒面板上的计数器会向你显示当前所有处于亮起状态的 LED 灯的总数（不仅仅是你操作的区间，而是全部 m 盏灯中亮着的总数）。

由于魔法道具的能量有限，你最多只能进行 40 次测试。请帮猫头鹰找出一对连接到同一盏 LED 灯的管脚，顺利通过实训考核吧！

Interaction Protocol

第一行包含两个整数 n 和 m （ $1 \leq m < n \leq 10^5$ ）— 分别表示输入管脚的数量和 LED 指示灯的数量。

接下来每要进行一次测试查询，请输出 `? L R`（ $1 \leq L \leq R \leq n$ ），其中 $[L, R]$ 是你希望发送脉冲信号的连续管脚区间。

在每次 `? 查询` 之后，请读取一个整数 `ans`（ $0 \leq ans \leq m$ ）— 表示当前所有处于亮起状态的 LED 灯的总数。

注意，你的查询次数不能超过 40 次。

如果 `ans = -1`，这意味着你的查询次数超过了允许的限制，或者你的查询格式无效。此时你的程序应当立即终止（例如，通过调用 `exit(0)`），你将获得 "Wrong Answer" 的判定。如果你忽略它并继续执行，你的程序可能会因为从关闭的流中读取数据而获得其他不可预期的判定结果。

如果你什么都不输出，或者忘记刷新输出缓冲区（包括最终答案的输出），你的解决方案将会获得 "Idleness Limit Exceeded" 的判定。

要刷新缓冲区，你可以使用以下语句（紧跟在打印查询和换行符之后）：

- C++: `fflush(stdout)` 或 `cout << endl;`
- Java: `System.out.flush();`

- Python: `stdout.flush()`
- Pascal: `flush(output);`
- 对于其他语言，请参阅相应的官方文档。

当你的程序推断出答案时，请按格式 `! x y` 输出，其中 x 和 y ($1 \leq x < y \leq n$) 表示两个连接到同一盏 LED 灯的管脚编号。

如果有多个合法的管脚对，输出任意一对即可。输出答案不计入测试次数限制，并且输出最终答案后，你的程序应当立即结束。

Example

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

Note

在样例中， $n = 4, m = 3$ 。黑盒内部的真实连线情况为：管脚 1 连灯 1，管脚 2 连灯 2，管脚 3 连灯 1，管脚 4 连灯 3。（由此可知，管脚 1 和管脚 3 连接着同一盏灯）。

用 1 表示亮，0 表示暗，则 3 盏灯的初始状态为 111。

第一轮询问：

查询 ? 1 3。

黑盒触发管脚 1（灯 1 翻转为暗），触发管脚 2（灯 2 翻转为暗），触发管脚 3（灯 1 再次翻转为亮）。当前灯的状态变为 101，共 2 盏灯亮起，黑盒返回 2。

第二轮询问：

查询 ? 2 2。

黑盒触发管脚 2（灯 2 由暗翻转为亮）。

当前灯的状态恢复为 111，共 3 盏灯亮起，黑盒返回 3。

输出：

输出 ! 1 3，指出管脚 1 和管脚 3 同时连接着灯 1。回答正确并结束程序。

Problem H. 不！我讨厌武汉！

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

武汉区域赛之后，神秘人给了Natsuya一串神秘的数字，并告知如果想要成为构造大神就必须还原出背后的排列。

给定一个整数 n 。Natsuya手上有一个长度为 $n - 1$ 的整数序列

$$b_1, b_2, \dots, b_{n-1}.$$

神秘人希望他构造一个长度为 n 的排列

$$p_1, p_2, \dots, p_n,$$

保证 $p_1, p_2, \dots, p_n$ 是一个 1 到 n 的排列，使其满足下面的要求。

对于每个位置 i ，定义该位置的前缀最大值与后缀最小值分别为

$$\text{premax}_i = \max(p_1, p_2, \dots, p_i),$$

$$\text{sufmin}_i = \min(p_i, p_{i+1}, \dots, p_n).$$

神秘人要求：对于所有 $1 \leq i \leq n - 1$ ，都满足

$$\text{premax}_i - \text{sufmin}_{i+1} = b_i.$$

请你帮Natsuya求出任意一个满足条件的排列 p 。若存在多个满足条件的排列，输出任意一个即可；若不存在这样的排列，则输出 -1 。

Input

本题包含多组测试数据。

第一行包含一个整数 T ，表示测试数据组数。($1 \leq T \leq 10^5$)

对于每组测试数据：

第一行包含一个整数 n ，表示排列的长度。($2 \leq n \leq 2 \cdot 10^5$)

第二行包含 $n - 1$ 个整数 $b_1, b_2, \dots, b_{n-1}$ ，表示给定的序列。($-10^9 \leq b_i \leq 10^9$)

保证所有测试数据的 n 之和不超过 $2 \cdot 10^5$ 。

Output

对于每组测试数据：

若存在合法的排列，输出一行 n 个整数 $p_1, p_2, \dots, p_n$ ，相邻整数用一个空格分隔。

若不存在合法的排列，输出一行一个整数 -1 。

Example

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

Note

对于第一组测试数据:

一种合法的排列是 $p = [3, 1, 2]$ 。

在第 1 个位置, 前缀最大值为 $\text{premax}_1 = 3$, 第 2 个位置往后的后缀最小值为 $\text{sufmin}_2 = \min(1, 2) = 1$, 二者之差为:

$$3 - 1 = 2 = b_1$$

在第 2 个位置, 前缀最大值为 $\text{premax}_2 = \max(3, 1) = 3$, 第 3 个位置往后的后缀最小值为 $\text{sufmin}_3 = 2$, 二者之差为:

$$3 - 2 = 1 = b_2$$

两个条件都满足, 因此 $p = [3, 1, 2]$ 是一个合法的答案。

对于第三组测试数据:

可以证明, 不存在任何满足条件的排列 p , 因此输出 -1 。

对于第四组测试数据, 序列 b 含有负数。一种合法的排列是 $p = [1, 2, 3, 4]$, 此时对每个 $1 \leq i \leq 3$ 都有 $\text{premax}_i = i$ 、 $\text{sufmin}_{i+1} = i + 1$, 故 $b_i = i - (i + 1) = -1$ 。

Problem I. 寻找whut

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

猫头鹰找不到whut了，请你帮他寻找一下！

给定一个仅由小写英文字母组成的字符串 S 。你可以对该字符串执行以下两种操作，次数不限（也可以不执行）：

- 操作 1：选择当前字符串中的任意一个字母并将其删除。
- 操作 2：选择当前字符串的一个连续子串，将该子串按字典序升序重新排列。（例如：对于字符串 "badc"，若选择连续子串 "adc" 进行操作，因为 'a' < 'c' < 'd'，操作后该子串变为 "acd"，整个字符串变为 "bacd"）。

我们想知道，在 S 的所有连续子串中，有多少个子串可以通过上述操作，最终变成目标字符串 "whut"。

注意：两个连续子串即使内容完全相同，只要它们在原串 S 中的起始或结束位置不同，就视为两个不同的子串。

Input

输入仅包含一行一个字符串 S ($1 \leq |S| \leq 5000$)。保证 S 中仅包含小写英文字母。

Output

输出一行一个整数，表示能够通过操作变成 "whut" 的连续子串数量。

Examples

standard input	standard output
awuhtb	4
whut	1

Problem J. 万物一心 2

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

在一款神奇爬塔卡牌游戏中，玩家每场战斗拥有一定数量的能量，并需要通过打出卡牌造成伤害。

现在，你手中只有两种卡牌：爪击2 和 万物一心 2。为了方便计算，本题对原游戏机制进行了简化，没有抽牌、弃牌、力量、易伤等额外机制。

战斗开始时，你拥有 m 点能量。

你可以按任意顺序打出以下两种卡牌：

- 爪击2：消耗 1 点能量。支付费用后，造成等于当前剩余能量的伤害，然后放入弃牌堆。
- 万物一心 2：消耗 3 点能量。支付费用后，将弃牌堆里的所有 爪击2 各自再次造成一次伤害。这些伤害的数值都等于当前剩余能量。（注意此卡牌只使用弃牌堆里的牌，并不消耗，使用后仍在弃牌堆）。

爪击2 可以打出任意多次，万物一心 2 最多只能打出 5 次。

你可以在任意时刻停止出牌，不要求用完所有能量。你不能在当前能量不足以支付费用时打出对应卡牌。

请你计算，在最优出牌顺序下，最多可以造成多少总伤害。

Input

第一行包含一个整数 T ，表示测试数据组数。

接下来 T 行，每行包含一个整数 m ，表示初始能量。

满足：

$$1 \leq T \leq 10^5。$$

$$1 \leq m \leq 2 \cdot 10^5。$$

所有测试数据的 m 之和不超过 $2 \cdot 10^5$ 。

Output

对于每组测试数据，输出一行一个整数，表示最多可以造成的总伤害。

Example

standard input	standard output
5	0
1	3
3	46
10	271
20	14615
100	

Note

对于 $m = 3$ ，一种最优方案是连续打出 3 张 爪击。

第 1 张 爪击 消耗 1 点能量后，剩余 2 点能量，造成 2 点伤害。

第 2 张 爪击 消耗 1 点能量后，剩余 1 点能量，造成 1 点伤害。

第 3 张 爪击 消耗 1 点能量后，剩余 0 点能量，造成 0 点伤害。
总伤害为 3。

Problem K. 商业街

Input file: **standard input**
Output file: **standard output**
Time limit: 4 seconds
Memory limit: 1024 megabytes

坎格鲁斯普雷决定创建一条商业街。

这条商业街一共有 n 个店面，编号为 $1, 2, \dots, n$ 。商业街将会连续营业 n 天。

在第 i 天的开始，会有一个商家来到坎格鲁斯普雷面前申请入驻。

坎格鲁斯普雷可以选择投资 a_i 元，让这个商家来到第 p_i 个店面，建立一家店铺。若该店铺被建立，则从第 i 天开始直到第 n 天，每一天它都可以被选择营业（包括第 i 天）。该店铺每营业一天，可以获得 b_i 元收益。

坎格鲁斯普雷也可以选择不投资这个商家。若不投资，则该店铺不会被建立，以后也不会出现。

保证 $p_1, p_2, \dots, p_n$ 是一个 1 到 n 的排列，也就是说，每个店面恰好会被一个商家申请入驻一次。

每天，在当天已经存在的所有店铺中，坎格鲁斯普雷可以选择任意数量的店铺营业，但需要满足：

- 相邻的两个店铺不能在一天同时营业。

没有营业的店铺在当天不会产生收益。

请你求出，在 n 天结束后，坎格鲁斯普雷能够获得的最大利润。利润在这里定义为总营业收益减去总投资金额。

Input

本题包含多组测试数据。

第一行包含一个整数 T ，表示测试数据组数。（ $1 \leq T \leq 5000$ ）

对于每组测试数据：

第一行包含一个整数 n ，表示店面数量和营业天数。（ $1 \leq n \leq 5000$ ）

接下来 n 行，每行包含三个整数 p_i, a_i, b_i ，表示第 i 天出现的商家的申请入驻的店面编号、投资费用以及每天营业可获得的收益。（ $1 \leq p_i \leq n, 1 \leq a_i, b_i \leq 10^9$ ）

保证 $p_1, p_2, \dots, p_n$ 是一个 1 到 n 的排列。

保证所有测试数据的 n 之和不超过 5000。

Output

对于每组测试数据，输出一行一个整数，表示坎格鲁斯普雷在 n 天后能够获得的最大利润。

Example

standard input	standard output
2	9
3	0
2 3 4	
1 2 3	
3 2 5	
1	
1 2 1	

Note

对于第一组测试数据：

一种最优方案是投资所有三个商家。

第 1 天，只有第 2 个店铺存在，选择它营业，收益为 4。

第 2 天，第 1,2 个店铺存在，但它们相邻，最多只能选择一个营业。选择第 2 个店铺营业，收益为 4。

第 3 天，第 1,2,3 个店铺都存在。可以选择第 1 个和第 3 个店铺同时营业，收益为：

$$3 + 5 = 8$$

总营业收益为：

$$4 + 4 + 8 = 16$$

总投资金额为：

$$3 + 2 + 2 = 7$$

因此最大利润为：

$$16 - 7 = 9$$

对于第二组测试数据：

如果投资唯一的商家，需要花费 2 元，最多只能获得 1 元收益，因此最优方案是不投资，最大利润为 0。

Problem L. 环绕轨道

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

储君挂了 M 张环绕轨道的能力，环绕轨道：每消耗 K 点能量可以获得 1 点能量（可以触发任意多次，每张环绕轨道独立），此时每张环绕轨道还需要 a_i 点能量即可触发回复（即当前计数为 a_i ）

即每次消耗1能量后，会给所有环绕轨道能力-1计数，若某张环绕轨道的计数达到 0，则会触发回复，回复后该轨道的计数重置为 K ，每张环绕轨道独立。

拥有 N 点初始能量，不断地每次消耗 1 点能量，最多能消耗多少能量？

输出最多能消耗多少能量，无限则输出 -1

Input

输入包含两行，第一行包含三个整数 $M(1 \leq M \leq 2 \times 10^5)$, $N(1 \leq N \leq 10^9)$, $K(1 \leq K \leq 10^9)$ ，第二行包含 M 个整数 $a_i(1 \leq a_i \leq K)$ ，分别表示每张环绕轨道还需要多少点能量即可触发回复。

Output

输出一个整数，表示最多能消耗多少能量，无限则输出 -1 。

Example

standard input	standard output
3 5 5 1 2 3	14

Note

- 第 1-3 点能量：分别在消耗时触发了轨道 1, 2, 3 的回复，因此剩余能量经历了 $5 \rightarrow 4 \rightarrow 5$ 的循环，前 3 点能量消耗完后，剩余能量仍为 5。
- 第 4-5 点能量：正常消耗，无轨道回复，剩余能量变为 3。
- 第 6 点能量：消耗时总累计消耗达到 6，再次触发轨道 1，能量回复 1，剩余能量变为 3。
- 第 7 点能量：消耗时总累计消耗达到 7，再次触发轨道 2，能量回复 1，剩余能量变为 3。
- 第 8 点能量：消耗时总累计消耗达到 8，再次触发轨道 3，能量回复 1，剩余能量变为 3。
- 第 9-10 点能量：正常消耗，剩余能量变为 1。
- 第 11-13 点能量：分别再次触发轨道 1, 2, 3 的回复，剩余能量保持为 1。
- 第 14 点能量：正常消耗，剩余能量变为 0，系统停机。

Problem M. 轮椅侦探古明地觉

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

agKc发现还没有在题目里放过觉恋于是有了这道题--

作为幻想乡有名的轮椅侦探，古明地觉总是能收到各种奇奇怪怪的委托，但这次，她被一张奇怪的纸条给难住了，好在，在她妹妹古明地恋无意识的帮助下，问题得到了解决。



古明地觉收到的纸条上记录着 n 次区间操作 l, r, k ，表示将区间 $[l, r]$ 内每个数加上 k 。

而古明地恋对姐姐的纸条做出了 m 次修改 L, R, K ，表示将纸条上第 $[L, R]$ 次操作的 k 值加上 K 。

假设数组初始为长度为 N 的全 0 数组，请你给出分别给出经过古明地恋修改前和修改后的纸条表示的数组。

Input

第一行三个正整数 N, n, m ($1 \leq N, n, m \leq 1 \times 10^5$)，分别表示数组长度，纸条上的操作数量和修改次数。

接下来 n 行，每行三个整数 l, r, k ($1 \leq l \leq r \leq N, -1 \times 10^8 \leq k \leq 1 \times 10^8$)，其中第 i 行表示纸条上的第 i 次区间操作。

再接下来 m 行，每行三个整数 L, R, K ($1 \leq L \leq R \leq n, -1 \times 10^8 \leq K \leq 1 \times 10^8$)，其中第 i 行表示古明地恋做出的第 i 次修改操作。

Output

共两行。

第一行 N 个整数，表示被修改前纸条上的操作所得到的数组。

第二行 N 个整数，表示被修改后纸条上的操作所得到的数组。

Example

standard input	standard output
6 3 3	-8 1 1 -8 -8 -10
1 5 2	7 25 25 7 7 0
2 3 9	
1 6 -10	
1 3 5	
2 3 4	
3 3 1	