

Problem A. 新太阳睡觉中心

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

'Cause morning rolls around, and it's another day of sun!

但是作为一个睡眠患者，肖恩经常睡觉睡个半天，因此，当他醒来时，他甚至记不清今天是星期几。

所以从某一天开始，他开始进行记录：每次他醒来的时候，他就会写下一个数字，表示此时外面是否有阳光。如果有，他会写一个 1；否则，他会写一个 0。在完成记录后，还没等太阳下山，或是太阳升起，他又再次入睡。假设每次肖恩醒来时，他看到的要么是阳光，要么是月光，但不会同时看到两者。

这些记录下来的数字实际上形成了一个长度为 n 的数组： $[a_1, a_2, \dots, a_n]$ ($\forall 1 \leq i \leq n, 0 \leq a_i \leq 1$)，其中 a_i 表示肖恩写下的第 i 个数字。

然而，随着时间的推移，一些写下的数字变得模糊不清，你无法判断它是 1 还是 0。如果有 k 个数字无法识别，则可能有 2^k 种不同的数组。

对于每个可能的数组，你都可以依据这个数组计算肖恩看到阳光的最小天数。如果将可能的不同数组的最小天数结果相加，得到的结果如何呢？由于答案可能很大，请将结果关于 998 244 353 取模后输出。

Input

每组测试包含多个测试用例。第一行是测试用例的数量 T ($1 \leq T \leq 10^4$)。

每个测试用例由两行组成。

第一行包含一个整数 n ($2 \leq n \leq 5 \times 10^5$)，表示笔记的数量。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($-1 \leq a_i \leq 1$)，表示每个记录下来的数字。只有当 $a_i = -1$ 时，数字是未知的。

可以保证单组测试中， $\sum n$ 不会超过 5×10^5 。

Output

对于每个测试用例，输出 1 个整数 — 不同数组的最小天数求和后，对 998 244 353 取模的结果。

Example

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

Note

在第一个测试用例中，当肖恩记录第一个和第三个数字 1 时，他看到的阳光一定是两个不同的天的。

在第二个测试用例中，肖恩从未看到阳光，因此答案是 0。

在第三个测试用例中，数组可能是 $[1, 1, 1]$ 或 $[1, 0, 1]$ 。如果数组是 $[1, 1, 1]$ ，那么这些数字可能都是同一天被记录下来的，因此结果是 1。如果数组是 $[1, 0, 1]$ ，那么与第一个测试用例一致，结果为 2。将两种情况的答案求和，结果是 $1 + 2 = 3$ 。

Problem B. 位完美

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

军蚁正在为一场残酷的战争做准备，但事情变得比他们想象的更复杂。

队伍中有 n 只军蚁。最初，每只军蚁 i ($1 \leq i \leq n$) 的力量为 a_i 。

然而，在战争期间会发生一些奇怪的事情：一股神秘力量会随机选择一个二元组 (i, j) ($1 \leq i < j \leq n$)，而选择后，蚂蚁 i 和蚂蚁 j 会消失，并且一只新蚂蚁会替代它们神奇地出现，其力量为 $a_i \oplus a_j$ ，即 a_i 和 a_j 的按位异或 (XOR)。

这种情况非常罕见，在整个战争中最多只会发生一次。

蚂蚁们认为整个队伍的总力量为 $\sum_{i=1}^n 2^{a_i}$ ，如果那件"奇怪的事"永远不会减少军队的整体力量，则这个军队被认为是 **位完美** 的。

由于忙于练兵，蚂蚁们没有足够的时间来检查队伍是否位完美。你能帮帮他们吗？可能会有 T 支不同的军队。

Input

每组测试包含多个测试用例。第一行包含测试用例的数量 T ($1 \leq T \leq 10^5$)。

每个测试用例由两行组成。

第一行包含一个整数 n ($2 \leq n \leq 5 \times 10^5$)，表示队伍中蚂蚁的数量。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^{18}$)，表示每只蚂蚁的力量。

保证在单组测试中， $\sum n$ 不会超过 5×10^5 。

Output

对于每个测试用例，如果该军队是 **位完美** 的，则输出 YES，否则输出 NO。你可以以任意大小写打印答案。

Example

standard input	standard output
3	YES
2	YES
3 5	NO
4	
1 2 4 8	
3	
1 2 3	

Note

在第一个测试用例中，如果两只蚂蚁合并，由于 $3 \oplus 5 = 6$ ，整个队伍的总力量从 $2^3 + 2^5 = 40$ 变为 $2^6 = 64$ ，因此该事件不会减少总力量。所以该队伍是 **位运算完美**。

Problem C. 彩色的树

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 512 megabytes

在彩色镇的世界里, 色彩斑斓!

彩色镇因其著名的景点——"彩色的树"而闻名。"彩色的树"并不是真正的树。它实际上是地面上的一幅巨型画作, 从热气球上俯瞰时, 它看起来像一棵树。

镇上的家庭每年都努力共同绘制这幅画。镇上有 n 个家庭, 而整个画作由 n 个部分组成。家庭 i 负责绘制第 i 部分。

为什么这幅画看起来像一棵树呢? 实际上, 这幅画的结构几乎就跟图论中的有根树完全一致! 除了第 1 部分之外, 每个部分都有其"父节点", 第 u 部分的父节点是 p_u 。当且仅当 i 是 j 的父节点或 j 是 i 的父节点时, 第 i 部分和第 j 部分是相邻的。

每个家庭生产自己的油漆。但由于家庭的油漆生产技术可能不稳定, 其生产的油漆颜色可能会有很大不同。我们用整数来表示每种颜色, 家庭 i 生产的油漆颜色可以是范围 $[l_i, r_i]$ 内的任何一个整数, 且各个颜色出现的概率相同。

为了减轻工作负担, 每个家庭每一年都完全使用自己生产的油漆来绘制自己负责的那一部分。它们共同形成了一幅美丽的画作!

关于油漆, 你还应该知道另一件事: 随着时间的推移, 家庭可能会提升或丢失技术, 因此 $[l_i, r_i]$ 可能会随时间变化。这个信息也是我们已知的。

为了向世界展示这幅画, 镇长肖恩决定每年拍一张画作的照片。在第 i 年, 当他拍摄画作时, 照片中显示的是第 u_i 部分对应的子树, 其他部分不会出现在照片中。第 i 部分在第 j 部分对应的子树中, 如果存在一个序列 v_1, v_2, \dots, v_k 使得 $v_1 = i, p_{v_t} = v_{t+1} (1 \leq t < k), v_k = j$ 。特别地, 第 i 部分在它自己对应的子树中。

肖恩想告诉别人这张照片有多么多彩, 因此他将色彩度定义为照片中的极大连通块数量。一个连通块指的是一个区域, 区域内的所有点都可以通过该连通块中的点到达另一个点。一个极大连通块是一个不能被其他连通块完全覆盖的连通块。

设第 i 年的色彩度为 C_i 。由于油漆的随机性, C_i 每年都是一个随机变量。为了了解更多关于 C_i 的信息, 肖恩向你询问 C_i 的期望和方差。你能帮助他吗?

关于方差, 随机变量 X 的方差为 $\text{Var}(X) = E[(X - EX)^2]$, 其中 EX 是 X 的期望。

这里, 保证树的根是 1, 树 (除了示例外) 是以这种方式随机生成的:

— 令 $\text{rnd}(a)$ 是从 $1, 2, \dots, a$ 中等概率选择一个数字的结果。

— 我们用 $[p_2, p_3, \dots, p_n]$ 来描述根树, 这意味着第 i 部分的"父节点"是 p_i 。

— 数组 $[p_2, p_3, \dots, p_n]$ 是这样生成的: $p_2 = \text{rnd}(1), p_3 = \text{rnd}(2), \dots, p_{i+1} = \text{rnd}(i), \dots, p_n = \text{rnd}(n-1)$ 。

查询不一定是随机生成的。

Input

每组测试包含多个测试用例。第一行包含测试用例的数量 t ($1 \leq t \leq 10^4$)。

每个测试用例由多行组成。

第一行包含 2 个整数 n, q ($4 \leq n \leq 5 \times 10^5, 1 \leq q \leq 10^5$), 表示树的大小和查询的数量。

第二行包含 $n - 1$ 个整数 p_2, p_3, \dots, p_n ($1 \leq p_i < i$), 表示 p_i 是 i 的父节点。

第三行包含 n 个整数 cl_1, cl_2, \dots, cl_n ($1 \leq cl_i < 998\ 244\ 353$), 第四行包含 n 个整数 cr_1, cr_2, \dots, cr_n ($cl_i \leq cr_i < 998\ 244\ 353$)。这表示家庭 i 的初始颜色范围是 $[cl_i, cr_i]$ 。

从第 5 行到第 $(q + 4)$ 行都是查询的信息。查询可以是两种类型之一:

- 1 $u\ l\ r$: 第一种查询。表示将顶点 u ($1 \leq u \leq n$) 的颜色范围更改为 $[l, r]$ ($1 \leq l \leq r < 998\ 244\ 353$)。
- 2 u : 第二种查询。表示询问顶点 u ($1 \leq u \leq n$) 的子树中的期望色彩度及其方差。
- 保证在单组测试中, 所有测试用例的 $\sum n$ 不超过 5×10^5 , $\sum q$ 不超过 10^5 。

Output

对于每个第二种查询, 输出 2 个整数, 分别表示顶点 u 的子树中的期望色彩度及其方差。

可以证明答案总是有理数, 且可以表示为 $\frac{p}{q}$ ($\gcd(p, q) = 1, p, q \in \mathbb{Z}, 998\ 244\ 353 \nmid q$)。你只需输出一个整数 x , 使得 $q \times x \bmod 998\ 244\ 353 = p$ 且 x 在范围 $[0, 998\ 244\ 353)$ 内。

Example

standard input	standard output
2	3 0
4 5	332748120 887328314
1 2 2	3 0
1 1 2 3	2 0
1 3 2 3	7 998244339
2 1	7 998244345
2 2	7 998244343
1 2 2 3	7 998244345
2 1	
2 2	
4 7	
1 1 1	
1 998244352 1 998244352	
998244352 998244352 1 998244352	
2 1	
1 2 1 998244352	
2 1	
1 3 998244352 998244352	
2 1	
1 4 1 1	
2 1	

Problem D. 噢，薯片...

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

驴认为薯片是最好的食物！

所以今天，当他决定去长途旅行时，他希望他的背包里装满各种薯片。他在家里的零食区寻找，发现了很多薯片。

为了更好地决定带哪些袋薯片（可能是总袋数的一个子集，也可以是空集），他定义了袋薯片的属性如下：

- h_i . 这袋薯片能给驴带来的快乐。
- s_i . 这袋薯片占据的空间。
- d_i . 这袋薯片的易碎度。

为了简便，我们将 h_i, s_i, d_i 记作这袋薯片的快乐度、空间量和易碎度。因为背包有大小，所以所选袋子的总占用空间不能超过背包的容量 V 。

然而，未占用的空间可能在驴旅行时造成颠簸，进一步导致价值损失。如果选择的薯片是所有薯片的一个非空子集，包含了第 i_1, i_2, \dots, i_k ($k \geq 1$) 包，而未占用的空间是 U ，则由于颠簸造成的总价值损失为 $(d_{i_1} + d_{i_2} + \dots + d_{i_k}) \times U$ 。特别的，不选择任何一包薯片的情况不会产生任何价值损失。

考虑到带薯片的利弊，整个背包的价值是这些袋薯片带来的总快乐值减去总价值损失。驴想要最大化这个价值，但就是无法做出决定。需要帮助！

Input

每组测试包含多个测试用例。第一行包含测试用例的数量 T ($1 \leq T \leq 10^4$)。

每个测试用例由多行组成。

第一行包含 2 个整数 n, V ($1 \leq n \leq 10^5, 1 \leq V \leq 500$)，薯片袋的数量和背包的总容量。

从第 2 行到第 $(n+1)$ 行，每行包含 3 个整数 h_i, s_i, d_i ($1 \leq s_i \leq 500, 1 \leq h_i, d_i \leq 10^9$)，第 i 袋薯片的快乐度、空间量和易碎度。

保证在单组测试中，所有测试用例的 $\sum n$ 不超过 10^5 ，且所有测试用例的 $\sum V^2$ 不超过 2.5×10^5 。

Output

对于每个测试用例，输出一个整数 — 价值的最大值。

Example

standard input	standard output
2	7
2 5	12
10 2 1	
2 2 100	
2 5	
10 2 1	
2 3 100	

Note

在第一个测试用例中，驴只选择第一袋，导致价值为 $10 - (5 - 2) \times 1 = 7$ 。

在第二个测试用例中，驴选择第一袋和第二袋，导致价值为 $10 + 2 - (5 - 2 - 3) \times (1 + 101) = 12$ 。

可以证明没有更好的策略。

Problem E. 高效率数字

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

平方数总是令人喜爱的。平方数是可以由整数的平方表示的数字。

现在有一组数字 $l, l+1, \dots, r$ ，它们想要成为平方数。女巫给它们提供了一种特殊的药水。当一个数字喝下一瓶药水时，它可以通过增加其一个质因子来增大自己，并且它可以自己选择增加哪个质因子。由于它们的梦想总是成为平方数，如果它们能够增大到一个平方数，它们总会这样做。

这种药水有点危险，因此喝下 2 瓶或更多可能会导致数字消失，所以这些数字不会这样做。此外，由于药水可能相当昂贵，这组数字决定在使尽可能多的数字变成平方数的情况下尽可能少地购买药水。这是最高效的做法！

他们应该购买多少瓶药水？

Input

输入包含一行。该行由两个整数 l, r ($2 \leq l < r \leq 10^{18}$) 组成。

Output

输出要购买的药水瓶数。

Example

standard input	standard output
2 9	2

Note

对于区间 $[2, 9]$ ，有两个数字需要买药水：2，因为 $2 + 2 = 4 = 2^2$ ；以及 6，因为 $6 + 3 = 3^2$ 。

Problem F. 不，是火灾

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

警报！火灾正在吞噬森林！

这片森林的形状像一个环，可以分为 n 个部分。第 i 部分 ($1 \leq i \leq n$) 仅与第 $((i-2) \bmod n) + 1$ 部分和第 $(i \bmod n) + 1$ 部分相邻。如果不采取任何行动，并且某一时刻第 i 部分发生火灾，则其相邻部分将在这一时刻后的一分钟内被火焰吞噬。

作为一个负责任的居民，你已经拨打了消防员的电话，希望他们尽快赶来扑灭这场可怕的火灾。然而，他们到达这里需要 t_0 分钟。你能做些什么吗？

可以的！你决定在森林的一个部分上点燃可控的火，以便在火灾到达这个区域之前立即建立一个火灾隔离区，也就是建立一个隔离带。这样，火灾就无法在此地蔓延。

但是，做出明智的决定是困难的：你需要选择火灾隔离区的位置，以确保更多的森林部分不被烧毁。在消防员到达之前，你能得到的最大的不被烧毁部分数量是多少？

Input

每组测试包含多个测试用例。第一行包含测试用例的数量 T ($1 \leq T \leq 10^4$)。

每个测试用例由 2 行组成。

第一行包含两个整数 n, t_0 ($1 \leq n, t_0 \leq 10^5$)，表示森林中的部分数量和消防员到达森林所需的分钟数。

第二行包含一个字符串 s ($|s| = n$)，该字符串由 0 和 1 组成。第 i 部分正在着火，当且仅当 $s_i = 1$ 。保证存在一个 i ($1 \leq i \leq n$) 使得 $s_i = 1$ 。

保证在单组测试中， $\sum n$ 不会超过 5×10^5 。

Output

对于每个测试用例，输出一个整数 — 当消防员到达时，森林中未烧毁部分的最小数量。

Example

standard input	standard output
3	1
5 2	0
10000	4
5 3	
10000	
10 1	
1000000100	

Note

对于第一个测试用例，一旦你点燃第 2 部分，火灾只会有时间烧毁第 4 和 5 部分，留下第 3 部分未烧毁。

Problem G. 几何，朋友

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

一个凸多边形懒洋洋地躺在欧几里得平面 xOy 上。"拥有一个朋友是什么样的感觉呢？"它自言自语道。一个小小的点 P 出现在位置 (x, y) 上，它引起了多边形的注意。它们成为了朋友，而多边形喜欢绕这个小点 P 逆时针旋转（即以之为旋转中心），因为这样它就可以覆盖一些以前从未覆盖过的区域。如果某个点在多边形的边缘或内部，则称这个点被多边形覆盖。

但是 P 感到担忧：有一天，多边形可能会厌倦绕着它旋转。它觉得，当多边形覆盖过的区域不再增大（也就是不再能覆盖新的点）时，多边形就会抛弃它，他们将不再是朋友。

多边形的旋转角速度为每年 1 弧度，设多边形开始旋转的时间为时间 0。请告诉 P 什么时候多边形会不再是它的朋友，如果确实如它所想。

Input

每组测试包含多个测试用例。第一行包含测试用例的数量 T ($1 \leq T \leq 10^4$)。

每个测试用例由多行组成。

第一行包含 3 个整数 n, x, y ($3 \leq n \leq 5 \times 10^5, |x| \leq 10^9, |y| \leq 10^9$)，其中 n 是凸多边形的边数， (x, y) 是点 P 的位置。

从第二行到第 $(n + 1)$ 行的每一行包含两个整数 x_i, y_i ($|x_i| \leq 10^9, |y_i| \leq 10^9$)，表示多边形中一个顶点的位置。保证给出的顶点是按逆时针顺序排列的。

保证在单组测试中，所有测试用例的 $\sum n$ 不超过 5×10^5 。

Output

对于每个测试用例，输出一个数值 — P 的担忧可能成真的时刻。如果真实答案是 ans ，你的答案 ans' 只要满足 $\frac{|ans - ans'|}{\max(1, ans)} \leq 10^{-6}$ ，就会被认为是正确的。

Example

standard input	standard output
3	1.570796326794897
4 0 0	6.283185307179586
1 0	4.712388980384690
0 1	
-1 0	
0 -1	
3 0 0	
1 1	
2 -1	
2 0	
3 0 0	
0 0	
1 -1	
1 1	

Problem H. 高速公路升级

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 512 megabytes

NowLand 的高速公路系统已经使用了几十年, 需要进行升级。

在这个国家, 有 n 个城市和 m 条单向高速公路, 这些高速公路从一个城市通往另一个城市。一辆车可以使用第 i 条高速公路在 t_i 分钟内从城市 u_i 到达城市 v_i 。一次关于这条公路的高速公路升级可以将这条路的用时减少 w_i 分钟。每条高速公路都可以进行多次升级。

作为一个自私的人, NowLand 的总统只考虑自己的利益。升级高速公路系统是他最后的工作, 而此后他就退休了。退休后, 他将从 NowLand 的首都城市 1 出发前往城市 n , 在那里度过余生。在这次旅程中, 他只会使用高速公路, 因此他只关心从城市 1 到城市 n 的时间, 并希望尽可能缩短它。

但是由于政府没有足够的资金, 因此高速公路升级的预算是有限的。在预算尚未确定的情况下, 总统需要为不同的情况做好准备。于是, 他想知道, 如果他可以进行 k 次升级, 那么他从城市 1 到城市 n 所需的最短时间是多少呢?

可以保证, 使用高速公路是从城市 1 到城市 n 的。

Input

每组测试包含多个测试用例。第一行包含测试用例的数量 T ($1 \leq T \leq 10^4$)。

每个测试用例由多行组成。

第一行包含 2 个整数 n, m ($4 \leq n \leq 10^5, 1 \leq m \leq 3 \times 10^5$), 表示国家中的城市和高速公路的数量。

从第 2 行到第 $(m+1)$ 行的每一行包含 4 个整数 u_i, v_i, t_i, w_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, 2 \leq t_i \leq 10^{12}, 1 \leq w_i \leq \min(t_i - 1, 10^9)$), 表示第 i 条高速公路的起点、终点、原始旅行时间和升级参数。可以保证可以使用高速公路从城市 1 旅行到城市 n 。

第 $(m+2)$ 行包含一个整数 q ($1 \leq q \leq 3 \times 10^5$), 表示查询数量。

从第 $(m+3)$ 行到第 $(m+q+2)$ 行的每一行包含一个整数 k_i ($1 \leq k_i \leq 10^9$), 表示升级的次数。可以保证 $\forall 1 \leq j \leq m, t_j - w_j \times k_i > 0$ 。

可以保证在单组测试中, 所有测试用例的 $\sum n$ 不会超过 2×10^5 , $\sum m$ 和 $\sum q$ 不会超过 6×10^5 。

Output

对于每个测试用例, 输出 q 个整数 — 使用 k_i 次升级从城市 1 到城市 n 的最短旅行时间。

Example

standard input	standard output
1	12
4 4	24
1 2 15 1	
1 3 20 2	
2 4 10 1	
3 4 10 1	
2	
9	
1	

Problem I. 相同，一定意义上

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

哈希函数可以将数据转换为固定长度的输出序列；也可以说，它为每个数据创建一个"标签"。

在本题中，输入始终是整数。如果两个不同的整数 x, y ($x \neq y$) 被转换成了相同的字符串，这可能是个坏事。在哈希函数的视角来看，这两个数可以一定意义上被认为是相同的。这被称为哈希冲突。

我们考虑以下带参数 k 的哈希函数：

$$H(x) = (x \bmod k) + (k \bmod x)$$

对于每对输入 (x, y) ($x \neq y$)，是否存在一个参数，使得对应的哈希函数会导致哈希冲突？如果存在，输出任意这样的 k ；否则，输出 -1 。

Input

每组测试包含多个测试用例。第一行包含测试用例的数量 T ($1 \leq T \leq 10^4$)。

每个测试用例由一行组成。该行包含两个整数 x, y ($1 \leq x, y \leq 10^9, x \neq y$)，即元素对。

Output

对于每个测试用例，输出一个整数 — 导致哈希冲突的参数 k ($1 \leq k \leq 10^{18}$)。如果没有不大于 10^{18} 的正整数满足条件，则输出 -1 。

Example

standard input	standard output
2	4
5 9	6
9 15	

Note

对于第一个案例， $5 \bmod 4 + 4 \bmod 5 = 1 + 4 = 5, 9 \bmod 4 + 4 \bmod 9 = 1 + 4 = 5$ 。

对于第二个案例， $9 \bmod 6 + 6 \bmod 9 = 3 + 6 = 9, 15 \bmod 6 + 6 \bmod 15 = 3 + 6 = 9$ 。

Problem J. 调分!

Input file: **standard input**
 Output file: **standard output**
 Time limit: 6 seconds
 Memory limit: 1024 megabytes

肖恩很高兴他已经批完了所有学生的卷子。但当他看到最终分数时, 他开心不起来了。这分也太低了! 幸运的是, 院长办公室允许肖恩调分, 以使分数"看起来更好", 只要在调分前更高的分在调分后仍然更高。

这里, 原始分数的范围是 $[1, 100]$, 并且有两种经典的调分操作:

— 类型 1: $f_1(x) = kx + b$, 其中 $10^{-2} \leq k \leq 1, 0 \leq b \leq 10^4, k + b \geq 1$ 。

— 类型 2: $f_2(x) = \sqrt{x}$ 。

在经过一系列上述函数操作后, 肖恩会将把调分后的分数映射到目标分数范围, 但这一点我们目前不太关心。

肖恩有一个长度为 n 的操作数组: $[g_1, g_2, \dots, g_n]$, 每个操作要么是类型 1, 要么是类型 2。

为了测试调分结果, 肖恩尝试用初始值 x 对函数序列进行测试。

每次, 他选择操作数组的一个子数组 $[g_l, g_{l+1}, \dots, g_r]$, 并按如下方式使用操作:

— 用值 x 初始化 a_0 。

— 使用以下方程计算变量 $a_1, a_2, \dots, a_{r-l+1}$: $a_1 = g_l(a_0), a_2 = g_{l+1}(a_1), \dots, a_{r-l+1} = g_r(a_{r-l})$ 。

由于肖恩太忙, 同时也太懒, 因此他请你帮助他。为了了解调分结果并确保你完全遵循他的过程进行计算, 他会让你回答 a_{r-l+1} 和 $\sum_{i=1}^{r-l+1} a_i$ 到底是多少。

肖恩可能会在他询问问题的过程中也会修改一些操作。你能处理这样的麻烦事儿吗?

Input

第一行包含一个整数 n ($1 \leq n \leq 10^5$), 表示操作数组的长度。

从第 2 行到第 $(n+1)$ 行的每一行包含一个表示函数 g_i 的行。每一行可以有两种形式:

— 1 k b: 操作是类型 1 — $kx + b$ ($10^{-2} \leq k \leq 1, 0 \leq b \leq 10^4, k + b \geq 1$)。 k, b 是两个小数, k 和 b 的小数点后都有恰好 2 位。

— 2: 操作是类型 2 — \sqrt{x} 。

第 $(n+2)$ 行包含一个整数 q ($1 \leq q \leq 10^5$), 表示查询的数量。

从第 $(n+3)$ 行到第 $(n+q+2)$ 行的每一行包含一个查询。查询有以下两种给定形式之一:

— 1 l r x ($1 \leq l \leq r \leq n, 1 \leq x \leq 100$): 类型 1 查询。询问操作子数组 $[g_l, g_{l+1}, \dots, g_r]$ 在初始值 x 下的 a_{r-l+1} 和 $\sum_{i=1}^{r-l+1} a_i$ 。 l, r, x 都是整数。

— 2 idx op ($1 \leq idx \leq n$): 类型 2 查询。将第 idx 个操作更改为 op , 其中 op 以此前操作函数的形式给出。

Output

对于每个类型 1 查询, 输出两个值: a_{r-l+1} 和 $\sum_{i=1}^{r-l+1} a_i$ 。

由于答案不一定是整数, 如果真实答案是 ans , 则你的答案 ans' 如果满足 $\frac{|ans - ans'|}{\max(1, ans)} \leq 10^{-4}$, 就会被认为是正确的。

Example

standard input	standard output
5	27.02893543 188.65806431
1 1.00 10.00	21.66726225 125.98917842
2	85.07500000 163.57500000
1 0.50 50.00	
2	
1 0.95 20.00	
5	
1 1 5 80	
2 3 2	
1 1 5 80	
2 4 1 0.35 65.00	
1 3 5 100	

Problem K. 第 K 个特殊子字符串

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

字符串是字符的序列，而 01 字符串是由仅包含 0 和 1 的字符串。

在查看了一些 01 字符串后，肖恩发现其中一些字符串非常特殊：它们有一个前缀和一个后缀是完全一致的，且这个前缀和这个后缀没有重叠部分。同时，这个前缀和后缀之间间隔的字符数不超过 k 。

然后，他遇到了一个长字符串 s 。他想更深入地了解这个字符串，并了解 s 的所有特殊子字符串的相关信息。子字符串由字符串中连续出现的字符构成。

他还希望你也能了解这些特殊子字符串，这真是太不妙了！为了确保你真的充分了解了这些串，他会问你这样的问题：

— s 中第 q_i 个按字典序最小的特殊子字符串是什么？

什么叫字典序更小呢？如单词 s_1 被认为按字典序小于 s_2 ，当且仅当一下情况之一为真：

— s_1 是 s_2 的前缀。

— 存在一个 j ，使得 $\forall 1 \leq i < j, s_1[i] = s_2[i]$ 且 $s_1[j] < s_2[j]$ 。

此外，如果 $s = '011101'$ ，则子字符串 $s[1\dots 2]$ 和 $s[5\dots 6]$ 被认为是不同的子字符串，因为两个子字符串选择了不同的索引。

另外，肖恩想确保你一直听着他所说的话，因此问题会根据上一个问题的答案稍作调整。更多信息请查看输入描述。

Input

第一行包含两个整数 n, k ($2 \leq n \leq 10^4, 0 \leq k \leq n - 2$)，表示给定字符串 s 的长度和特殊子字符串的参数。

第二行包含字符串 s ，该字符串仅由 0 和 1 组成。

第三行包含一个整数 q ($1 \leq q \leq 5 \times 10^5$)，表示肖恩提出的问题数量。

从第 4 行到第 $(q + 3)$ 行的每一行包含一个整数 v_i ($1 \leq v_i \leq 10^9$)，表示你需要回答的问题。

设 $total$ 为 s 中不同特殊子字符串的总数，设 ans_i 为第 i 个查询的答案 ($1 \leq i \leq q$)。答案 ans_i 总是一个 01 字符串，它是整数 x_i 的二进制表示（忽略不必要的前导零）。设 $x_0 = 0$ ，那么在第 i 个查询中， $q_i = ((x_{i-1} \bmod 998\,244\,353 + v_i - 1) \bmod total + 1)$ ，你需要找到第 q_i 个按字典序最小的特殊子字符串。

保证输入的字符串 s 一定至少存在一个特殊子字符串。

Output

对于每个查询，输出一个字符串作为答案，即第 q_i 个按字典序最小特殊子字符串。

保证输出字符串的总长度不会超过 10^7 。

Example

standard input	standard output
6 1	11011
110111	11
2	
5	
6	

Note

原始字符串的特殊子字符串为 $s[2...4] = '101'$, $s[1...2] = '11'$, $s[4...5] = '11'$, $s[5...6] = '11'$, $s[1...5] = '11011'$, $s[4...6] = '111'$ (按字典序排列)。因此, $total = 6$ 。

第一个查询询问第 5 个元素, 因此结果是 11011。

11011 是 27 的二进制表示, 因此第二个查询询问 $(27 \bmod 998\,244\,353 + 6 - 1) \bmod 6 + 1 = 3$ -rd 元素, 因此结果是 11。

Problem L. 爱胜过一切

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

这是一个充满爱的社区！

社区中有 n 名居民，每位居民 i ($1 \leq i \leq n$) 在社区中都有一个他/她非常爱的居民 a_i ($1 \leq i \leq n$)。每两个居民爱着不同的居民。居民可以爱自己。保证 n 是偶数。

有一天，发生了一件坏事：他们需要选择 2 名居民，他们将永远被禁止结婚。

为了防止将来发生这样的事情，其余的 $n - 2$ 名居民会组成 $\frac{n}{2} - 1$ 组对象结婚，每组对象由 2 个人组成（当然）。如果一组对象由居民 x 和居民 y 组成，而 x 既不爱 y ， y 也不爱 x ，那这也太怪了！因此这样的情况永远不会发生。

因此，作为婚姻规划者，您需要弄清楚如何安排这一切。您想知道不同婚姻计划的数量。两个婚姻计划是不同的，当且仅以下条件至少其一为真：

- 在一个计划中，某人 i 已婚，而在另一个计划中，他/她未婚。
- 在一个计划中，某人 i 与 j 结婚，而在另一个计划中，他/她没有与 j 结婚。

由于计划的数量可能非常多，请输出数量关于 998 244 353 取模的结果。

Input

每组测试包含多个测试用例。第一行包含测试用例的数量 T ($1 \leq T \leq 10^4$)。

每个测试用例由两行组成。

第一行包含 1 个整数 n ($4 \leq n \leq 5 \times 10^5$)，表示社区中的居民数量。保证 n 是偶数。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$)，其中 a_i 表示居民 i 爱的人。保证如果 $i \neq j$ ($1 \leq i, j \leq n$)，则 $a_i \neq a_j$ 。

保证单组测试内， $\sum n$ 不会超过 5×10^5 。

Output

对于每个测试用例，输出 1 个整数 — 婚姻计划的数量关于 998 244 353 取模的结果。

Example

standard input	standard output
2	3
4	9
1 3 4 2	
6	
3 4 5 6 2 1	

Problem M. 又算错了吗?

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 512 megabytes

肖恩在自己的数学作业中遇到了这个问题:

— 在一个二维平面中, 边长在 $\{1, 2, \dots, s\}$ 的三角形有多少种不同的形状, 且周长不超过 l ? 如果两个三角形可以通过平移和旋转完全重合, 则认为它们是相同的形状。请注意, **不允许翻转**。因此, 对于 $\triangle ABC$ 和 $\triangle A'B'C'$ (A, B, C 和 A', B', C' 按照逆时针顺序排列), 如果 $AB = 2, BC = 3, CA = 4$ 且 $A'B' = 2, A'C' = 4, C'A' = 3$, 它们 **不** 被认为是相同的形状。

肖恩喜欢用二进制来考虑问题, 因此他使用数字的二进制表示上面提供的所有数字。他遍历了所有可能的 (a, b, c) , 试图找到答案, 因此他想计算的是满足以下条件的三元组 (a, b, c) 的数量:

- $1 \leq a, b, c \leq s$ 且都是整数。
- $a + b > c, a + c > b, b + c > a$
- $a + b + c \leq l$

然后, 他从这些三元组中找到对应的不同形状的三角形个数。然而, 肖恩数学太烂了, 因此当他计算 $a + b + c$ 时, 他完全忘记了进位, 因此实际上, 他得到的是 $a \oplus b \oplus c$ 的结果, 即 a, b, c 的按位异或 (XOR) 和。

除了犯了这个错误外, 肖恩想知道他是否还犯了其他错误, 因此他问你, 如果第三个条件是 $a \oplus b \oplus c \leq l$ 而不是 $a + b + c \leq l$, 那么答案是什么。你能帮他吗?

由于答案可能非常大, 请输出它关于 998 244 353 取模的结果。

Input

每组测试包含多个测试用例。第一行包含测试用例的数量 t ($1 \leq t \leq 10^4$)。

每个测试用例由一行组成。该行包含 2 个字符串 s_l, s_s ($|s_l| \leq 10^5, |s_s| \leq 10^5$), 表示整数 l 和 s 的二进制表示。保证 $l > 0$ 且 $s > 0$, 并且字符串 s_s 和 s_l 的第一个字符始终为 1。

保证单组测试, $\sum |s_l|$ 不超过 5×10^5 , $\sum |s_s|$ 不超过 5×10^5 。

Output

对于每个测试用例, 输出 1 个整数 — 不同形状的三角形数量关于 998 244 353 取模的结果。

Example

standard input	standard output
2	267
101 1111	25
1111 101	