Problem A. 插入 1

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

给定一个没有前导零的十进制整数 x，请你在其中插入一个数字 1，使得修改后还是一个没有前导零的十进制整数，且 x 最大。

这里的前导零是指位于数字表示序列中首个非零数位左侧的所有零，如果数字本身是 0，则它不包含前导零。

Input

第一行包含一个整数 T $(1 \le T \le 10^5)$，表示测试用例的数量。

接下来是 T 个测试用例。对于每个测试用例：

唯一的一行包含一个整数 x $(-2^{63} \le x < 2^{63})$。

Output

对于每个测试用例，在一行中输出一个整数，表示修改后最大可能的 x 值。

Example

standard input	standard output
6 2 1 0 -1 -2 -1000000000000000000	21 11 10 -11 -12 -10000000000000000001

Problem B. 逆序对奇偶性

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	32 megabytes

注意：本题的内存限制不同寻常

序列中的逆序对是指失去自然顺序的元素对，而对于一个 0 到 $(n-1)$ 的排列 $P=[p_0,p_1,\ldots,p_{n-1}]$，其逆序对数是指满足 $i<j$ 且 $p_i>p_j$ 的元素对 (p_i,p_j) 数量。

给定一个排列 $P=[p_0,p_1,\ldots,p_{n-1}]$ 和 $(n-1)$ 次修改 $[(l_1,r_1,d_1),(l_2,r_2,d_2),\ldots,(l_{n-1},r_{n-1},d_{n-1})]$，你的任务是确定每次修改前后 P 的逆序对数的奇偶性。

具体来说，排列 P 将有 n 个版本，其中第一个版本如上所述给出，对于 $i=1,2,\ldots,n-1$，第 $(i+1)$ 个版本是在第 i 个版本的基础上将下标在 l_i 和 r_i 之间的元素区间循环左移 d_i 次后得到的（更多细节说明请查看样例解释），而你需要为 P 的每个版本计算逆序对数的奇偶性。

此外，由于给定的 n 有点大，我们将通过下面的随机生成器从给定的 n、A、B 和 C 来生成排列 P 和修改。

随机生成的序列 $[f_0,f_1,\ldots]$ 基于以下定义：

- $\wedge$ 表示按位与，$\oplus$ 表示按位异或，$\lfloor x\rfloor$ 表示将 x 向下取整到最近的整数；
- 令 $U=2^{30}-1$，下面的 f_x、g_x、h_x 都是介于 0 和 U 之间的整数（闭区间）；
- 令 $f_{-3}=A\wedge U$，$f_{-2}=B\wedge U$ 和 $f_{-1}=C\wedge U$；
- 对于 $i=0,1,\ldots,$ 令 $g_i=f_{i-3}\oplus((2^{16}f_{i-3})\wedge U)$；
- 对于 $i=0,1,\ldots,$ 令 $h_i=g_i\oplus\left\lfloor\frac{g_i}{2^5}\right\rfloor$；
- 对于 $i=0,1,\ldots,$ 令 $f_i=h_i\oplus((2h_i)\wedge U)\oplus f_{i-2}\oplus f_{i-1}$。

结合上述生成器，

- $x \bmod y$ $(y>0)$ 表示 x 除以 y 的余数；
- 排列 P 从初始状态 $[0,1,\ldots,n-1]$ 开始，依次交换 p_i 和 $p_{i+(f_i \bmod (n-i))}$ $(i=0,1,\ldots,n-1)$ 后，得到它的第一个版本；
- 对于 $i=1,\ldots,n-1$，令 $l_i=\min(f_{n+3i-3} \bmod n, f_{n+3i-2} \bmod n)$，$r_i=\max(f_{n+3i-3} \bmod n, f_{n+3i-2} \bmod n)$，$d_i=(f_{n+3i-1} \bmod n)+1$。

Input

第一行包含一个整数 T $(1\le T\le 10^5)$，表示测试用例的数量。

接下来是 T 个测试用例。对于每个测试用例：

唯一一行包含四个整数 n、A、B 和 C $(1\le n\le 10^7, 0\le A,B,C\le 10^9)$。

保证 T 个测试用例中 n 之和不超过 10^7。

Output

对于每个测试用例，在一行中输出一个长度为 n 的字符串，其中第 i 个字符为 0 则表示 P 的第 i 个版本的逆序对数为偶数，为 1 则表示为奇数。

Example

standard input	standard output
5 3 0 0 0 3 0 0 1 3 0 1 0 3 6 7 8 10 123 456 789	000 111 111 110 1111101111

Note

示例中前两个测试用例生成的序列 $[f_0, f_1, \ldots]$ 为：

- $[0, 0, 0, 0, 0, 0, 0, 0, 0, \ldots]$；
- $[1, 0, 202754, 1, 202755, 692070724, 692070724, 692070725, 792595, \ldots]$。

示例中所有测试用例生成的排列和修改为：

- $P = [0, 1, 2]$，$\text{modifications} = [(0, 0, 1), (0, 0, 1)]$；
- $P = [1, 0, 2]$，$\text{modifications} = [(0, 1, 2), (1, 2, 2)]$；
- $P = [1, 0, 2]$，$\text{modifications} = [(0, 2, 1), (0, 1, 2)]$；
- $P = [2, 1, 0]$，$\text{modifications} = [(1, 1, 3), (1, 2, 3)]$；
- $P = [3, 1, 2, 8, 5, 0, 4, 7, 6, 9]$，$\text{modifications} = [(1, 8, 2), (0, 6, 10), (3, 5, 10), (2, 4, 2), (7, 8, 9), (1, 2, 7), (0, 9, 2), (8, 9, 2), (5, 7, 5)]$。

对于最后一个测试用例，

- P 的第二个版本为 $[3, 8, 5, 0, 4, 7, 6, 1, 2, 9]$，是通过将索引区间 $[1, 8]$ 循环左移两次得到的（即 $[\ldots, 1, 2, 8, 5, 0, 4, 7, 6, \ldots] \to [\ldots, 2, 8, 5, 0, 4, 7, 6, 1, \ldots] \to [\ldots, 8, 5, 0, 4, 7, 6, 1, 2, \ldots]$）；
- P 的第三个版本为 $[0, 4, 7, 6, 3, 8, 5, 1, 2, 9]$，是通过将索引区间 $[0, 6]$ 循环左移 10 次得到的（即 $[3, 8, 5, 0, 4, 7, 6, \ldots] \to [8, 5, 0, 4, 7, 6, 3, \ldots] \to \cdots \to [0, 4, 7, 6, 3, 8, 5, \ldots]$）；
- 所有版本的逆序数分别为 13、21、19、19、21、22、23、25、25 和 27。

Problem C. 伯努利原理

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

伯努利原理是流体力学中的一个定义，它声称流体的动能、势能与内能的总和保持不变。托里切利定律是伯努利原理在特定场景的一种表现，即容器内液体从容器表面小孔流出时，其初始速度为 $\sqrt{2gh}$，其中 h 为小孔中心到液面最高点的距离，g 为重力加速度。

现在有一竖直放置在水平平面上的圆柱体容器内盛满了液体，并持续缓慢注入液体使其液面最高点高度保持不变。其柱侧面开有 n 个小孔（编号 1 到 n）正在向外流水，水流做竖直方向初速度为 0、加速度为 g 的自由落体运动（水平方向初速度如上所述）。已知液面最高点高度、各小孔中心分别到水平平面的高度差，请你分析各孔流出液体在到达平面前的水平方向移动距离，根据该距离的大小关系对这 n 个小孔进行排序。

Input

第一行包含一个整数 T $(1 \le T \le 10^5)$，表示测试用例的数量。

接下来是 T 个测试用例。对于每个测试用例:

第一行包含两个整数 n 和 H $(1 \le n \le 10^5, 2 \le H \le 10^9)$，表示小孔数量和液面最高点高度。

第二行包含 n 个整数 $h_1, h_2, \ldots, h_n$ $(1 \le h_i < H)$，其中第 i 个数字表示编号 i 的小孔到水平平面的高度差。

保证 T 个测试用例中 n 之和不超过 10^6。

Output

对于每个测试用例，在一行中输出 n 个数字，表示 n 个小孔的编号关于水平移动距离排非降序的结果。如果有多种可能，输出任意一种。

Example

standard input	standard output
2 5 10 1 3 5 7 9 5 20 1 3 5 7 9	1 5 2 4 3 1 2 3 4 5

Problem D. 圣诞树

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

圣诞树上有 n 个灯泡，通过 $(n-1)$ 根电线连接。每根电线连接两个不同的灯泡，使得所有灯泡形成一棵以第 1 个灯泡为根的有根树。一开始圣诞树的美观度是 0，你可以点亮一些灯泡来提升美观度。

点亮第 i 个灯泡需要消耗 a_i 单位的电力，并会提升 b_i 的美观度。由于电线会发热，对于第 j 根电线，如果其连接的第 u_j 个灯泡和第 v_j 个灯泡都被点亮，则还会额外消耗 c_j 单位的电力。由于点亮的灯泡也会发热，为了安全起见，每个点亮的灯泡都必须通过电线与至少一个没有点亮的灯泡直接连接。

有一个量程为 m 的电表，会显示总消耗电力对 m 取模的结果。有 q 次询问，每次询问给定 r 和 k，你需要计算在只点亮以第 r 个灯泡为根的子树里的灯泡，保持子树外的灯泡不被点亮，且电表显示 k 时圣诞树的最大美观度，以及得到最大美观度的点亮灯泡的方案数。

两种点亮灯泡的方案不同，当且仅当存在至少一个灯泡在一种方案中被点亮而在另一种方案中未被点亮。由于方案数可能很大，你只需要输出方案数模 (10^9+7) 的值。如果不存在可行的方案，则认为最大美观度是 -1，对应的方案数是 0。

Input

第一行包含三个整数 n，m 和 q $(1 \le n \le 100, 1 \le m \le 3 \times 10^4, 1 \le q \le 10^5)$，表示圣诞树上的灯泡个数、电表的量程以及询问的个数。

接下来 n 行，第 i 行包含两个整数 a_i 和 b_i $(1 \le a_i, b_i \le 10^6)$，表示点亮第 i 个灯泡需要消耗 a_i 单位的电力，并会提升 b_i 的美观度。

接下来 $(n-1)$ 行，第 j 行包含三个整数 u_j，v_j 和 c_j $(1 \le u_i, v_j \le n, 1 \le c_j \le 10^6)$，表示第 j 根电线连接第 u_j 个灯泡和第 v_j 个灯泡，且如果这两个灯泡都被点亮，则还会额外消耗 c_j 单位的电力。保证输入的电线将所有灯泡连成一棵树。

接下来 q 行，每行包含两个整数 r 和 k $(1 \le r \le n, 0 \le k < m)$，表示一个询问。

Output

输出 q 行，每行包含两个整数，表示在只点亮以第 r 个灯泡为根的子树里的灯泡，保持子树外的灯泡不被点亮，且电表显示 k 时圣诞树的最大美观度，以及得到最大美观度的点亮灯泡的方案数模 (10^9+7) 的值。如果不存在可行的方案，则认为最大美观度是 -1，对应的方案数是 0。

Examples

standard input	standard output
4 4 4 1 1 2 2 3 3 4 4 1 2 1 1 3 2 2 4 3 1 0 1 1 1 2 1 3	4 1 5 2 2 1 7 1
1 2 2 1 1 1 0 1 1	0 1 -1 0

Note

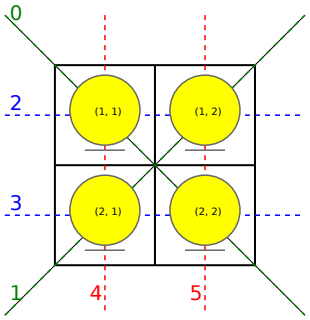
在第二个样例中，单个灯泡不能被点亮。

Problem E. 矩阵彩票

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

有一个彩票箱，抽奖行为对应一个 $n \times n$ 矩阵。矩阵里每个格子有一盏灯，初始每盏灯都为关闭状态。每次抽奖都会以固定概率尝试点亮某一盏灯（即使这盏灯已经被点亮），而每盏灯一旦被点亮就会持续保持开启状态。换句话说，每次抽奖有 $p_{i,j}$ 的概率选择第 i 行、第 j 列也即坐标 (i,j) 的灯去尝试点亮 $(1 \le i,j \le n)$，如果已经点亮则忽略此次操作。

每当某一行、某一列、某一对角线上的所有灯都被点亮时，主办方将为抽奖者送出这条线对应的特殊礼物。已知有 $(2n+2)$ 种礼物，编号为 0 到 $(2n+1)$，且每种礼物恰好对应上述的一条线，每条线也恰好对应一种礼物。



抽奖者对某些特殊礼物情有独钟，迫切地想知道为了拿到这些礼物需要的期望抽奖次数是多少，于是找到了你。你需要回答 q 次询问，其中第 i 次询问对应一个数字 t_i $(0 < t_i < 2^{2n+2})$。令 t_i 的长度为 $(2n+2)$ 的二进制表示为 $(b_{2n+1}b_{2n}\cdots b_0)_2$，也即 $t_i = \sum_{j=0}^{2n+1} b_j 2^j$ 且 $b_j \in \{0,1\}$。如果 $b_j = 1$ 则表示此次询问需要拿到编号为 j 的礼物，而你需要基于这些目标礼物给出期望的抽奖次数模 (10^9+7) 的值（具体定义见输出格式）。

Input

第一行包含一个整数 T $(1 \le T \le 200)$，表示测试用例的数量。

接下来是 T 个测试用例。对于每个测试用例：

第一行包含两个整数 n 和 q $(2 \le n \le 7,\ 1 \le q \le 10^5)$。

接下来的 n 行描述了选择每个单元格的概率，第 i 行包含 n 个整数 $w_{i,1}, w_{i,2}, \ldots, w_{i,n}$，表示 $p_{i,j} = \frac{w_{i,j}}{W}$，其中 $W = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j}$ $(1 \le w_{i,j} \le W \le 10^3)$。

接下来的 $(2n+2)$ 行描述了与礼物相关的线，第 i 行包含 $2n$ 个整数 $x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2}, \ldots, x_{i,n}, y_{i,n}$，表示与礼物 $(i-1)$ 相关的线上恰好有 n 个不同单元格，坐标依次为 $(x_{i,1}, y_{i,1}), (x_{i,2}, y_{i,2}), \ldots, (x_{i,n}, y_{i,n})$。保证所有可能的线（$n$ 行、n 列、2 对角线）都会给出。

下一行包含 q 个整数 $t_1, t_2, \ldots, t_q$ $(0 < t_i < 2^{2n+2})$。

保证对于 $n = 4,5,6,7$ 的测试用例数量分别不超过 $50, 10, 3, 1$。

同时保证 T 个测试用例中的 q 之和不超过 5×10^5。

Output

对于每个测试用例，输出 q 行，其中第 i 行包含一个整数，表示你对第 i 个询问的答案。

可以证明期望始终是一个有理数。此外，在本题的条件下，还可以证明当该值表示为最简分数 u/d 时，$d \not\equiv 0 \pmod{10^9+7}$，因此存在唯一整数 f $(0 \le f < 10^9+7)$，使得 $u \times f \equiv d \pmod{10^9+7}$，这个 f 即为所求。

Example

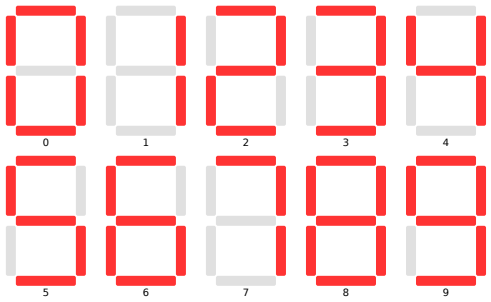
standard input	standard output
1 2 12 10 20 30 40 1 1 2 2 1 2 2 1 1 1 1 2 2 1 2 2 1 1 2 1 1 2 2 2 1 2 3 4 5 6 8 9 10 16 32 63	500000014 333333342 361111126 666666683 928571447 166666680 404761912 154761917 849206362 833333350 833333345 361111126

Problem F. 破旧的 LED 灯管

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

有一座古老高耸的建筑物在电梯中使用 LED 灯管展示楼层编号，然而出于维修经费的压力，管理者试图减少电梯可以停靠的楼层数量，并想尽量精简不必要的 LED 灯管线路。

目前，电梯可以停靠的楼层数量已经确定，而接下来缩减 LED 灯管线路的工作则被分配给了你。你知道这座建筑物有 10^m 层，每层楼的编号在 0 到 $(10^m - 1)$ 之间，且都是恰好 m 位数字（可能会在左侧补零）。对于楼层编号的每一位，原先配有 7 根灯管（如下图所示），它们会根据需要显示的数字灵活地开关。



现在，其中 n 个楼层已经被选定继续支持停靠，电梯在移动和停靠时也只会展示这 n 种楼层数值。你的任务是让电梯里尽量少的灯管还能继续正常工作，使得利用这些灯管的开关状态即可判断停靠楼层对应的 n 个可能编号中的哪一个。你的时间非常宝贵，请尽快计算出需要保持运行的最小灯管数量。

Input

第一行包含一个整数 T $(1 \le T \le 100)$，表示测试用例的数量。

接下来是 T 个测试用例。对于每个测试用例：

第一行包含两个整数 n 和 m $(1 \le n \le 100, 1 \le m \le 3)$。

第二行包含 n 个两两不同的整数 $x_1, x_2, \ldots, x_n$ $(0 \le x_i < 10^m)$，表示被选定的楼层编号，保证每个楼层编号恰好有 m 位数字。

另外还保证 T 个测试用例中 n 之和不超过 10^3。

Output

对于每个测试用例，在一行中输出一个整数，表示需要保持正常工作的灯管的最小数量。

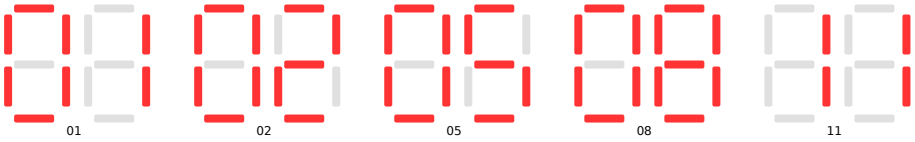
Example

standard input	standard output
3 10 1 0 1 2 3 4 5 6 7 8 9 5 2 01 02 05 08 11 1 3 012	5 3 0

Note

对于样例中第二个测试用例，下面列出了一个可能的解决方案。

原始状态：



缩减后：

Problem G. 修改最小生成树

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

无向图是任意一条边可以双向移动的图，简单图是不存在重复边和自环的图，连通图是图中任意两点间存在至少一条路径的图。

一个无向图的生成树是包含该图所有顶点和部分边的连通子图，且任意两点间恰好有一条路径。无向图可以带有边权，此时边权之和最小的生成树称为最小生成树。

给定一张含有 n 个点与 m 条边的带权无向简单图，其中每条边权要求是 1 到 k 之间的整数，现在我们希望再加一条满足边权限制的边进去，使得新图是一张存在**至少一个**生成树的简单图，且新加的边一定在它的**任何一个**最小生成树里，请你帮忙计算有多少种方案，输出方案数模 (10^9+7) 的值。

Input

第一行包含一个整数 T $(1 \le T \le 10^5)$，表示测试用例的数量。

接下来是 T 个测试用例。对于每个测试用例:

第一行包含三个整数 n、m 和 k $(1 \le n \le 10^5, 0 \le m \le \min\left(2\times10^5, \frac{n(n-1)}{2}\right), 1 \le k \le 10^9)$。

接下来的 m 行描述给定的简单图，每行包含三个整数 u、v 和 w $(1 \le u,v \le n,\ u \ne v,\ 1 \le w \le k)$，表示一条连接 u 和 v 的权重为 w 的边。

保证 T 个测试用例中 N 之和、M 之和分别不超过 5×10^5 和 10^6。

Output

对于每个测试用例，在一行中输出一个整数，表示满足要求的加边方案数模 (10^9+7)。

Example

standard input	standard output
5 3 2 2 1 2 1 2 3 2 3 3 5 1 2 3 2 3 4 1 3 5 2 1 3 1 2 3 6 6 5 1 2 3 1 3 1 2 4 2 3 5 2 2 6 4 5 6 5 2 0 25	1 0 0 20 25

Problem H. 区间 LRU

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

小 Q 正在学习操作系统内存管理中的页面置换算法 – LRU（最近最少使用）算法。

为了帮助小 Q 理解这个算法，你需要回答一些关于 LRU 的问题。首先，我们简要介绍一下这个算法的原理:

- 设置一个容量为 k 的缓存容器，用于存储加载的页面和相关信息。最初，容器中没有页面。
- 系统将顺序处理访问页面的请求，每次只处理一个请求。
- 对于访问已经在缓存中的页面（即缓存命中），直接使用缓存作为数据源，并更新页面的最近访问时间。
- 对于访问不在缓存中的页面（即缓存未命中），如果缓存中的页面数量达到了 k，首先从缓存中驱逐页面使得缓存中的页面数量少于 k，然后将访问的页面从其真实地址加载到缓存中，使用缓存作为数据源并更新页面的最近访问时间。
- 每当需要驱逐页面时，选择最近访问时间最早的页面从缓存中释放。

接下来，我们使用编号从 1 到 n 的页面，并提供一个长度为 n 的数组 $[a_1, a_2, \ldots, a_n]$，以及 q 个询问，你需要回答这些询问。每个询问是以下两种类型之一:

- 1 l r k: 按顺序访问编号为 $a_l, a_{l+1}, \ldots, a_r$ 的页面，使用最大容量为 k 的 LRU 缓存，请你回答缓存命中次数。
- 2 l r k: 按顺序访问编号为 $a_l, a_{l+1}, \ldots, a_r$ 的页面，使用 LRU 缓存，要求缓存命中次数至少为 k，请你回答 LRU 缓存所需的最小容量。如果无法达成命中次数目标，则回答 -1。

Input

第一行包含一个整数 T $(1 \le T \le 10^5)$，表示测试用例的数量。

接下来是 T 个测试用例。对于每个测试用例:

第一行包含两个整数 n 和 q $(1 \le n, q \le 10^5)$。

第二行包含 n 个整数 $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le n)$。

接下来的 q 行描述询问，每行包含四个整数 t, l, r, k $(t \in \{1, 2\}, 1 \le l \le r \le n, 1 \le k \le r - l + 1)$，表示一个第 t 种类型的询问，参数含义如题目所述。

保证 T 个测试用例中 n 之和、q 之和分别不超过 2×10^5 和 2×10^5。

Output

对于每个测试用例，输出 q 行，其中第 i 行包含一个整数，表示第 i 个询问的答案。

Example

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

Problem I. 网球比赛

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

校集训队有 m 位候选队员，现在需要选出 $2n$ 位队员两两配对组合，参加 n 场网球双打比赛。

第 i 名队员有一个预估能力值 e_i，并按照参训时长分为初级和高级两类。

第 i 场比赛要求参赛选手的能力值均不超过 l_i，且出于稳定性考虑，教练要求每组配对的两名队员能力值相差不超过 d。

虽然没有限制球员必须基于相同类别进行配对，但是为了充分挖掘初级队员的潜力，教练们希望选出一定初级队员参赛。

教练想知道，对于 $t = 0, 1, \ldots, 2n$，是否存在恰好选出 t 名初级队员又满足上述约束的配对方案。如果不存在这样的方案，输出 -1，否则输出所选 $2n$ 名球员的能力值的最大可能总和。

Input

第一行包含一个整数 T ($1 \le T \le 10^5$)，表示测试用例的数量。

接下来是 T 个测试用例。对于每个测试用例：

第一行包含三个整数 n, m 和 d ($1 \le n \le 20$, $2n \le m \le 10^5$, $0 \le d < 10^9$)。

第二行包含 n 个整数 $l_1, l_2, \ldots, l_n$ ($1 \le l_i \le 10^9$)。

接下来 m 行描述候选队员信息，第 i 行包含两个整数 e_i 和 t_i ($1 \le e_i \le 10^9$, $t_i \in \{1, 2\}$)，表示第 i 名队员的预估能力值为 e_i，如果 $t_i = 1$ 则为初级类型，否则为高级类型。

保证 T 个测试用例中 m 之和不超过 2×10^5。

Output

对于每个测试用例，在一行中输出 $(2n + 1)$ 个整数，其中第 i 个整数表示恰好选出 $(i - 1)$ 名初级队员的答案。

Example

standard input	standard output
2 4 9 400 800 900 1050 1200 46 1 264 2 295 1 305 1 332 2 678 1 770 2 903 2 1291 2 4 9 400 800 900 1050 1200 46 1 264 2 295 1 305 2 332 2 678 2 770 2 903 2 1291 1	-1 -1 -1 -1 3593 -1 -1 -1 -1 -1 -1 3593 -1 -1 -1 -1 -1 -1

Problem J. 根号 -2 进制乘法

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

在这道题中，你需要解决一个简单的任务 – A **乘** B。

令 $U = \{a + b\sqrt{-2} | a, b \in \mathbb{Z}\}$ 为进行乘法运算的整个集合。显然，U 中任何两个元素的乘积也是 U 中的一个元素。

每个元素 $x \in U$ 在 $\sqrt{-2}$ 进制下可以表示为长度为 n 的 01 字符串，形式为 $(c_{n-1}c_{n-2}\dots c_0)_{\sqrt{-2}}$，其中 $x = \sum_{i=0}^{n-1} c_i\sqrt{-2}^i$，$c_i \in \{0, 1\}$ $(i = 0, 1, \dots, n-1)$ 且当 $n > 1$ 时有 $c_{n-1} = 1$。比如，$-1 + \sqrt{-2} = \sqrt{-2}^0 + \sqrt{-2} + \sqrt{-2}^2 = (111)_{\sqrt{-2}}$ 和 $2 - 4\sqrt{-2} = (4-2) + (-8+4)\sqrt{-2} = \sqrt{-2}^2 + \sqrt{-2}^4 + \sqrt{-2}^5 + \sqrt{-2}^7 = (10110100)_{\sqrt{-2}}$。

你的任务是计算 U 中两个元素 A 和 B 在 $\sqrt{-2}$ 进制下的乘积。

Input

第一行包含一个整数 T $(1 \le T \le 10^5)$，表示测试用例的数量。

接下来是 T 个测试用例。对于每个测试用例:

唯一的一行包含两个 01 字符串 A 和 B $(|A|, |B| \ge 1, |A| + |B| \le 2 \times 10^5)$。

保证 T 个测试用例中 $(|A| + |B|)$ 之和不超过 2×10^6。

Output

对于每个测试用例，在一行中输出一个 01 字符串，表示 A 和 B 在进制 $\sqrt{-2}$ 下的乘积。

Example

standard input	standard output
5 0 1 10 10 10 11 101 101 110 110	0 100 110 1 10110100

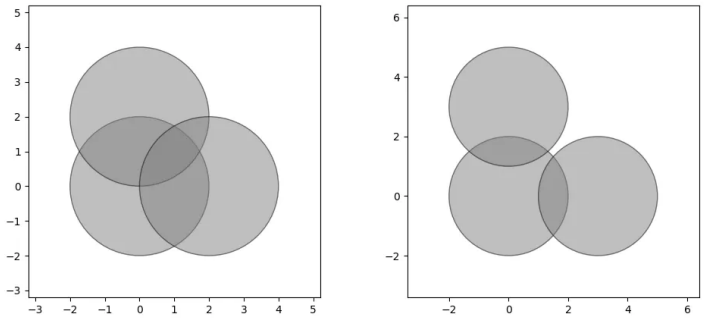
Note

对于样例的最后一个测试用例，$(-2 + \sqrt{-2})^2 = 2 - 4\sqrt{-2}$。

Problem K. 好多喷洒器！！！

Input file:	standard input
Output file:	standard output
Time limit:	6 seconds
Memory limit:	512 megabytes

在二维笛卡尔平面上有 n 个喷洒器，每个喷洒器可以浇灌一个圆形区域的内部（含边界）。



样例中的喷洒器。

你需要计算可以被这些喷洒器浇灌到的区域面积。

Input

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

接下来是 T 个测试用例。对于每个测试用例：

第一行包含一个整数 n ($1 \le n \le 10^5$)，表示喷洒器的数量。

接下来 n 行，第 i 行包含三个整数 x_i、y_i 和 r_i ($-10^4 \le x_i, y_i \le 10^4$, $1 \le r_i \le 10^4$)，表示第 i 个喷洒器浇灌一个以 (x_i, y_i) 为中心，半径为 r_i 的圆形区域内或边界上的面积。

保证 T 个测试用例中 n 之和不超过 10^5。

Output

对于每个测试用例，在一行中输出一个实数，表示可以被这些喷洒器浇灌到的区域面积。

如果你的答案的绝对误差或相对误差不超过 10^{-6}，则将被视为正确。更正式地，假设你的输出为 a，标准答案为 b，当且仅当 $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$ 时，你的输出会被接受。

Example

standard input	standard output
2 3 0 0 2 0 2 2 2 0 2 3 0 0 2 0 3 2 3 0 2	27.360855087873111763 34.072617811256640852