第十一届中国大学生程序设计竞赛（哈尔滨）

正式赛

2025 年 11 月 9 日

HARBIN INSTITUTE OF TECHNOLOGY

试题列表

A	k-子集和最大公约数问题
B	液压机
C	Many Many Sequence Covering Problems
D	比赛
E	函数求和
F	1-2-按位或子序列问题
G	扫雪
H	匹配
I	六边形翻转
J	幻想乡的裁判长
K	01 背包
L	网格避障
M	连通的正三角形

本试题册共 13 题，28 页。

如果您的试题册不完整，请立即通知志愿者。

题目 A. k-子集和最大公约数问题

考虑一个无限大的可重集 S，它的元素包含 $a_1, a_2, \cdots, a_n$，每种元素都有无限个。在此基础上，我们定义函数 $f(k)$ 如下：考虑 S 的每个大小恰为 k 的子集 S'，计算 S' 中的元素之和，求出所有这些子集对应的和的**最大公约数**，作为 $f(k)$ 的值。形式化地，

$$f(k) = \gcd_{S' \subseteq S, |S'|=k} \left(\sum_{x \in S'} x \right)$$

- 例如，对于 $a = [3, 6]$，我们有 $f(2) = \gcd(3+3, 3+6, 6+6) = 3$。

现在请你求出 $f(k)$ 的最大值,并求出取得最大值的最小的 k。特别地,如果最大值不存在,报告 “`infinite`”。

输入

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 4 \times 10^5$）表示测试数据组数。对于每组测试数据：

第一行包含一个整数 n（$1 \le n \le 10^5$），表示 S 内数的种数。

第二行包含 n 个整数 $a_1, a_2, \ldots, a_n$（$1 \le a_i \le 10^{18}$），表示 S 中的元素。

保证所有数据 n 之和不超过 4×10^5。

输出

对于每组数据，如果 $f(k)$ 存在最大值，则输出一行两个整数，分别表示 $f(k)$ 的最大值以及取得最大值时最小的 k。

否则输出 “`infinite`”（不包含引号）。

样例

standard input	standard output
2 2 3 6 2 2 2	3 1 infinite
2 3 1 4 7 4 4 16 28 34	3 3 6 3

注释

对于样例一中的第一组数据，可以发现无论 k 取多少，都有 $f(k) = 3$。

对于样例一中的第二组数据，有 $f(k) = 2k$，故 f 会无限增长从而不存在最大值。

该页留空

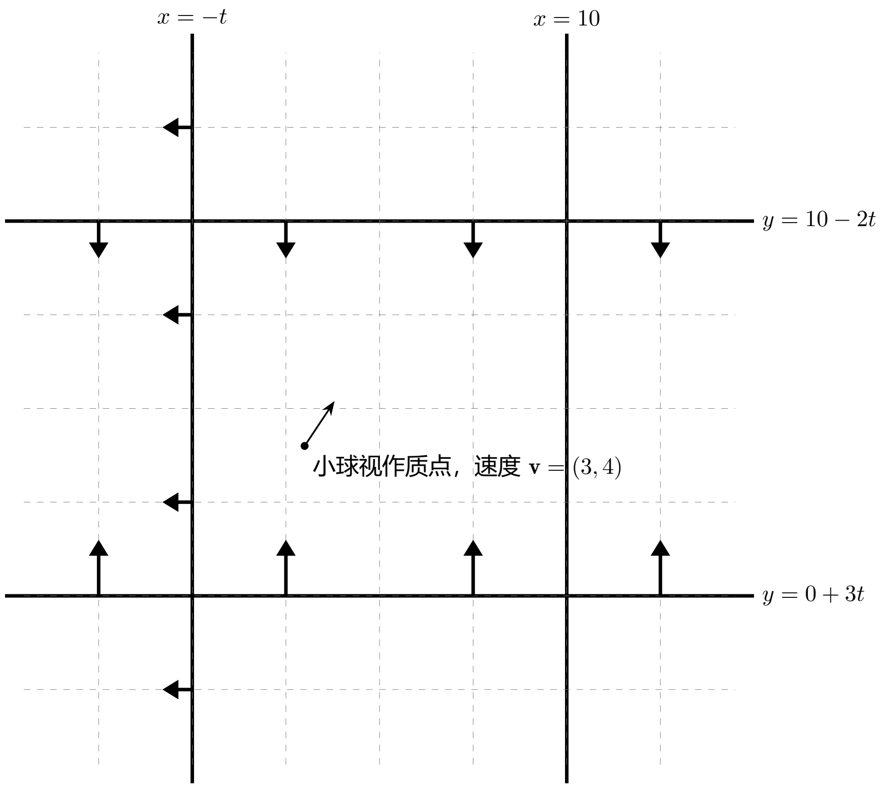
题目 B. 液压机

在一个二维试验场中，以四条无限长直线作为可移动的 “墙”：上下为水平线，左右为竖直线。放入一颗**大小可忽略不计**的小球后，小球做匀速直线运动；每当与某对平行墙中的一条发生接触，沿该法向方向进行完全弹性反弹（仅改变该方向的速度符号），另一方向保持不变。上下两条水平线**相互靠近**，左右两条竖直线**彼此远离**，最终会在某一时刻合拢并将小球夹住。

给定小球的初始位置与速度，以及四条直线的初始位置与速度。以上下两线相遇时刻为终止时刻，输出该时刻小球的坐标。

- 小球的初始位置记为 (x_{beg}, y_{beg})，初始速度记为 (v_x, v_y)。
- 下方的水平线和上方的水平线初始时分别位于 $y = y_1$ 和 $y = y_2$,在经过时间 t 后移动到 $y = y_1 + v_{y_1}t$ 和 $y = y_2 - v_{y_2}t$。
- 左边的竖直线和右边的竖直线初始时分别位于 $x = x_1$ 和 $x = x_2$，在经过时间 t 后移动到 $x = x_1 - v_{x_1}t$ 与 $x = x_2 + v_{x_2}t$。

保证初始时小球一定在四条直线围成的矩形内部，即 $x_1 < x_{beg} < x_2$，$y_1 < y_{beg} < y_2$。



样例初始情况

我们保证每次碰撞都是小球先撞上墙，即保证小球在竖直方向的速度一定大于两条水平线的运动速度，$v_{y_1}, v_{y_2} < |v_y|$。

为了保证上下两条线必定相遇，保证两条水平线的速度不同时为 0。

发生碰撞时具体的规则如下:

1. 小球在不接触任何直线时做匀速直线运动。
2. 与水平线接触时，仅将纵向速度取相反数（把 v_y 变为 $-v_y$），横向速度不变。

3. 与竖直线接触时，仅将横向速度取相反数（把 v_x 变为 $-v_x$），纵向速度不变。

4. **反弹判定与直线的平移速度无关，仅按“取相反数”处理对应速度分量。**

输入

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 100$）表示测试数据组数。对于每组测试数据：

第一行包含四个整数 $x_{beg}, y_{beg}, v_x, v_y$（$-20 \le v_x, v_y \le 20$，$v_x \neq 0$、$v_y \neq 0$），分别表示小球的初始坐标以及初始速度。

第二行包含四个整数 $y_1, y_2, v_{y_1}, v_{y_2}$（$0 \le y_1 < y_{beg} < y_2 \le 10^6$，$0 \le v_{y_1}, v_{y_2} < |v_y|$，$v_{y_1} + v_{y_2} \neq 0$），分别表示水平线的初始位置和运动速度。

第三行包含四个整数 $x_1, x_2, v_{x_1}, v_{x_2}$（$0 \le x_1 < x_{beg} < x_2 \le 10^6$，$0 \le v_{x_1}, v_{x_2} \le 20$），分别表示竖直线初始位置和运动速度。

输出

对于每组数据输出一行两个用空格分隔的浮点数 x, y，表示小球终止时刻的坐标。

如果你的输出和标准答案的绝对误差或相对误差小于 10^{-3}，那么你的答案将会被判为正确。

形式化地，设你的答案是 out，标准答案为 ans，你的结果被认为正确，当且仅当 $\frac{|out - ans|}{\max(1, |ans|)} \le 10^{-3}$。

样例

standard input	standard output
1 3 4 2 5 0 10 3 2 0 10 1 0	7.000000000000 6.000000000000
2 631043 768016 20 20 1 1000000 1 0 631040 631050 1 0 631044 768016 20 20 1 1000000 1 0 631041 631051 1 0	-0.456742460183 1000000.000000000000 0.543257539817 1000000.000000000000

题目 C. Many Many Sequence Covering Problems

考虑以下两个问题:

Sequence Covering Problems

给出三个长度为 n 的非负整数序列 a, b, c，你可以执行如下操作无数次:

- 花费 $b_l + c_r$ 的代价选择区间 $[l, r]$，满足 $a_l, a_{l+1}, \cdots, a_r$ 的最小值不为 0，将 $a_l, a_{l+1}, \cdots, a_r$ 全部减去 1。

你需要用最小的代价将 a 中的所有数变为 0。

Many Sequence Covering Problems

在 Sequence Covering Problems 的基础上，给出两个非负整数序列 d, e，现在可以做以下操作任意次:

- 选择 $i \in [1, n]$，花费 d_i 的代价让 b_i 增大 1，或花费 e_i 的代价让 c_i 增大 1。

在所有操作结束后，对操作后的 b, c 序列求解其 Sequence Covering Problems，设其答案为 P，花费的代价为 Q，你需要最大化 $P - Q$ 的值，如果这个值可以是无限大，请输出 ∞，否则请给出这个最大值。

现在你要解决 **Many Many Sequence Covering Problems**:

Many Many Sequence Covering Problems

给出五个长度为 n 的残缺序列 A, B, C, D, E，规定若 $A_i \geq 0$，则 A_i 的值已经给出，否则认为 A_i 的取值范围为 $[0, -A_i]$ 的**整数**，对于 B, C, D, E 序列同理。

你需要求出所有可能的情况下，其对应的 Many Sequence Covering Problems 的答案之和。由于答案可能是 ∞，所以你需要分别给出两个结果:

1 当答案不为 ∞ 时，所有的答案之和;

2 有多少种情况其对应的答案为 ∞。

由于答案可能很大，请将答案对 998244353 取模。

输入

第一行包含一个整数 n（$1 \leq n \leq 5000$），表示序列的长度。

第二行包含 n 个整数 $A_1, A_2, \ldots, A_n$（$0 \leq |A_i| \leq 5000$）。

第三行包含 n 个整数 $B_1, B_2, \ldots, B_n$（$0 \leq |B_i| \leq 5000$）。

第四行包含 n 个整数 $C_1, C_2, \ldots, C_n$（$0 \leq |C_i| \leq 5000$）。

第五行包含 n 个整数 $D_1, D_2, \ldots, D_n$（$0 \leq |D_i| \leq 5000$）。

第六行包含 n 个整数 $E_1, E_2, \ldots, E_n$（$0 \leq |E_i| \leq 5000$）。

输出

输出只有一行，包含两个整数，分别表示当答案不为 ∞ 时，所有的答案之和对 998244353 取模后的值，

对应的答案为 ∞ 的情况数对 998244353 取模后的值。

样例

standard input	standard output
2 1 1 1 2 2 1 -1 -1 -1 -1	8 12
3 -1 -2 2 1 3 0 -3 0 -2 1 -3 0 1 3 3	408 228

题目 D. 比赛

土豆鸡厨神比赛（Cook Chicken Potato Contest）是厨师界最为知名的赛事之一。赛场场地固定提供 k 个灶台，参赛选手会被主办方小 Q 分成 k 个**人数相同**的队伍来进行比赛。

为了考验选手之间的团队配合,以及为比赛增加更多看点,安排选手队伍时,小 Q 会使相同队伍的选手实力差距**尽量大**。假定参加比赛的选手的实力分别为 $a_1, a_2, \ldots, a_n$, 以及他们所属于的队伍分别为 $t_1, t_2, \ldots, t_n$, 小 Q 定义比赛的精彩度为:

$$D = \min_{1 \le i < j \le n} \begin{cases} |a_i - a_j| & t_i = t_j \\ +\infty & t_i \ne t_j \end{cases}$$

现在按照按实力顺序从小到大给定 n 名可能会参加比赛的选手。由于选手的实力并不固定，因此会用一个区间 $[l_i, r_i]$ 来描述第 i 名选手，表示其在某场比赛的实际实力可能会是该区间的**任意实数**。又由于选手的实力随编号增大单调不降，因此保证对于 $\forall 1 \le i < j \le n$，有 $l_i \le l_j$，$r_i \le r_j$。

小 Q 有 q 个办赛计划，其中第 i 个计划会邀请编号在 L_i 和 R_i 之间的选手参加，你需要帮小 Q 计算是否存在一种分配选手队伍的方式，使得比赛的精彩度**可能**不低于 D_i。

输入

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^5$）表示测试数据组数。对于每组测试数据:

第一行包含两个整数 n, k（$1 \le n \le 5 \times 10^5$，$1 \le k \le \min(5, n)$），表示可能的选手数量以及队伍的数量。

接下来 n 行，第 i 行包含两个整数 l_i, r_i（$0 \le l_i \le r_i \le 10^{12}$），表示第 i 名选手可能发挥出的实力。保证 $\forall 1 \le i < n$，有 $l_i \le l_{i+1}$，$r_i \le r_{i+1}$。

接下来一行包含一个整数 q（$1 \le q \le 10^5$），表示办赛计划的数量。

接下来 q 行，第 i 行包含三个整数 L_i, R_i, D_i（$1 \le L_i \le R_i \le n$，$k \mid (R_i - L_i + 1)$，$0 \le D_i \le 10^{12}$），表示第 i 个办赛计划会邀请标号在 L_i 到 R_i 之间的选手，小 Q 预期的精彩度为 D_i。

保证所有测试数据的 n 之和不超过 10^6，q 之和不超过 10^5。

输出

每组测试数据输出 q 行，第 i 行输出 “YES” 或 “NO” 分别表示小 Q 第 i 个计划的预期是否有可能被达成。你可以以任意形式输出答案（大写或小写），比如 “yEs”，“yes”，“Yes” 和 “YES” 都会被认为是肯定的答案。

样例

standard input	standard output
2 4 2 1 1 3 3 4 4 6 6 3 1 2 3 3 4 2 1 4 2 5 1 1 3 2 3 4 6 7 10 8 12 6 1 3 2 1 3 3 2 4 4 2 4 5 3 5 4 3 5 5	YES YES YES YES NO YES NO YES NO

题目 E. 函数求和

考虑正整数 n 质因数分解的结果：

$$n = \prod_{i=1}^{k} p_i^{\alpha_i}, \text{ 其中 } p_1 < p_2 < \cdots < p_k$$

现在给出一个长为 m 的序列 r，保证序列 r 中的元素两两不同。定义 $f(n)$ 如下：

$$f(n) = \prod_{i=1}^{m} (p_{r_i} \times \alpha_{r_i})$$

如果 $r_i > k$，我们认为此时 $p_{r_i} \times \alpha_{r_i} = 1$。

现在有 q 次查询，每次给出一个 x，查询 $\left(\sum_{i=1}^{\lfloor \frac{n}{x} \rfloor} f(ix)\right) \bmod 2^{32}$ 的结果。

为了减少输出量，请输出所有答案的异或和。特别地，**我们以质因数分解的形式给出 x**。

输入

输入第一行包含三个整数 n, m, q（$1 \le n \le 7 \times 10^8$，$1 \le m \le 25$，$1 \le q \le 5 \times 10^5$），分别表示查询上界的参数，序列 r 的长度，以及询问的数量。

输入第二行包含 m 个整数 $r_1, r_2, \ldots, r_m$（$1 \le r_i \le 25$，r_i 互不相同）。

接下来 q 行，第 i 行会先输入一个整数 L 表示 x 的质因子分解项数。接下来输入 $2L$ 个整数 $P_1, A_1, P_2, A_2, \ldots, P_L, A_L$（$P_i$ 为质数且互不相同），表示 $x = \prod_{i=1}^{L} P_i^{A_i}$，保证 $1 \le x \le n$，$A_i \ge 1$。**特别地**，若 $L = 0$，则表示 $x = 1$。

输出

输出一行包含一个整数，表示所有询问答案的异或和。

样例

standard input	standard output
10 2 5 2 3 0 1 5 1 1 2 1 1 7 1 2 2 1 3 1	31

该页留空

题目 F. 1-2-按位或子序列问题

给定一个长度为 n 的只包含 1 和 2 的序列 $a_1, a_2, \ldots, a_n$。你可以进行若干次如下操作:

- 选择 $1 \le i < n$，将 a_i 和 a_{i+1} 从序列中删除，并在它们的原来的位置插入 $a_i \mid a_{i+1}$，其中 $\mid$ 表示按位或。**在每次操作后 n 的大小会减 1。**

例如若序列 $a = [1, 2, 1]$，选择对 $i = 2$ 进行操作，操作后序列会变为 $a = [1, 3]$。

求在进行若干次操作后，能产生多少种本质不同的序列，你需要输出答案对 $10^9 + 7$ 取模的结果。两个序列不同当且仅当它们的长度不同或某个数不同。

n 可能很大，因此序列会通过将相同数字压缩成同一段的格式输入。特别地，**保证每一段相同数字的长度，从前往后单调不降**。

输入

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^6$）表示测试数据组数。对于每组测试数据:

第一行输入两个整数 m, a_1（$1 \le m \le 10^6, 1 \le a_1 \le 2$）表示序列分成的段数，以及 a_1 的值。

第二行输入 m 个整数 $l_1, l_2, \ldots, l_m$（$\boldsymbol{1 \le l_1 \le l_2 \le \ldots \le l_m \le 10^9}$），其中 l_i 表示序列中第 i 段数的长度。

由于相邻的段内数的值不同，故可以通过 a_1 和 $l_1, l_2, \ldots, l_m$ 唯一确定这个长度为 $n = \sum_{i=1}^{m} l_i$ 的序列。

保证所有数据中 m 之和不超过 10^6。

输出

对于每组数据，输出一个整数表示答案对 $10^9 + 7$ 取模的结果。

样例

standard input	standard output
2 3 1 1 1 2 8 2 1 2 3 4 5 6 7 8	7 2961300

注释

样例一中第一组测试数据表示的序列为 $a = [1, 2, 1, 1]$，进行若干次操作后能表示出的本质不同的序列有:

- $[1, 2, 1]$
- $[1, 2, 1, 1]$
- $[1, 3]$
- $[1, 3, 1]$
- $[3]$
- $[3, 1]$
- $[3, 1, 1]$

该页留空

题目 G. 扫雪

大雪过后，小 w 需要打扫一下自己的院子，使凹凸不平的雪堆看起来更整齐。小 w 的院子可以视作一个 $n \times m$ 的网格，第 i 行第 j 列的格子内覆盖了相对高度为 $h_{i,j}$ 的积雪。小 w 有以下几种清理雪堆的操作：

1. 选择 $1 \le i < n, 1 \le j \le m$，将第 i 行第 j 列的雪堆推到下一行上，此操作会使得 $h_{i,j}$ 减 1，$h_{i+1,j}$ 加 1。**此操作不需要代价。**
2. 选择 $1 \le i \le n, 1 \le j < m$，将第 i 行第 j 列的雪堆推到下一列上。此操作会使得 $h_{i,j}$ 减 1，$h_{i,j+1}$ 加 1。**此操作不需要代价。**
3. 选择 $1 \le i \le n, 1 \le j \le m$，给第 i 行第 j 列的雪堆造雪。此操作会使得 $h_{i,j}$ 加 1。此操作需要 1 的代价。
4. 选择 $1 \le i \le n, 1 \le j \le m$，给第 i 行第 j 列的雪堆抽雪。此操作会使得 $h_{i,j}$ 减 1。此操作需要 1 的代价。

小 w 想要进行若干次操作使得所有 $h_{i,j}$ 都变成 0，且需要的代价最小。你能帮他计算一下最小的代价吗？

输入

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^6$）表示测试数据组数。对于每组测试数据：

输入第一行包含两个整数 n, m（$1 \le n, m \le 10^3$），表示院子的行数与列数。

接下来 n 行第 i 行包含 m 个整数 $h_{i,1}, h_{i,2}, \ldots, h_{i,m}$（$-10^9 \le h_{i,j} \le 10^9$），第 j 个整数表示第 i 行第 j 列格子上积雪的相对高度。

对于所有数据，保证它们的 $n \times m$ 之和不超过 10^6。

输出

对于每组测试数据，输出一行一个整数，表示小 w 完成目标需要的最小代价。

样例

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

该页留空

题目 H. 匹配

现有 n 个集合，第 i 个集合有 a_i 个元素，所有集合一共有 $2m$ 个互不相同的元素（$\sum a_i = 2m$），每个元素有一个唯一的所属集合。

对于每轮操作，我们将所有元素随机匹配为 m 对，对于每对匹配，我们随机挑选一个元素，将其从原本所属的集合中取出，放入与其匹配的元素的所属集合中。请问期望操作多少轮，所有元素将属于一个集合。

输入

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 100$）表示测试数据组数。对于每组测试数据：

第一行输入两个整数 n, m（$1 \le n \le 2m \le 400$），表示初始集合的数量以及每轮操作的匹配数。

第二行输入 n 个整数 $a_1, a_2, \ldots, a_n$（$1 \le a_i \le 2 \times m$，$\sum a_i = 2 \times m$），其中 a_i 表示第 i 个集合有 a_i 个元素。

保证所有测试数据的 n 之和不超过 800，m 之和不超过 400。

输出

对于每组数据，输出一行一个整数表示答案在模 998244353 意义下的值。

可以证明答案是一个有理数 $\frac{P}{Q}$。您需要输出 $PQ^{-1} \bmod 998244353$ 的值，其中 Q^{-1} 是满足 $QQ^{-1} \bmod 998244353 = 1$ 的整数。

样例

standard input	standard output
4 2 1 1 1 2 2 2 2 2 2 1 3 3 2 1 1 2	1 3 499122179 499122180
3 6 5 2 1 1 3 1 2 10 100 90 12 18 1 1 24 7 4 37 6 10 200 102 19 25 11 43 97 19 28 11 45	347465837 225202828 437065763

注释

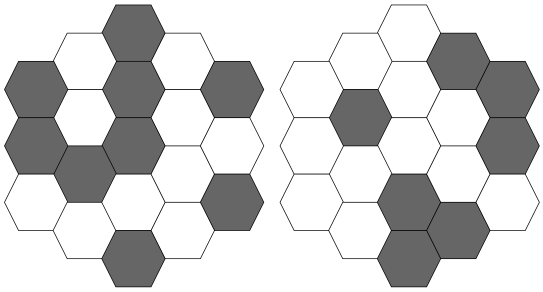
对于样例 1：

- 第一组测试数据：我们将初始状态表示为 $[1, 1]$，只有 1 种可能的匹配方式，并且这 2 个元素无论谁被选中，都有操作一轮后状态为 $[2]$，因此期望答案为 1。
- 第二组测试数据：我们将初始状态表示为 $[2, 2]$，有 3 种匹配方式，每种匹配方式有 4 种挑选元素的方法，共 12 种不同操作，其中有 4 种操作在操作后状态为 $[4]$，其余的 8 种操作在操作后状态为 $[2, 2]$，即有 $\frac{1}{3}$ 的概率操作后只剩一个集合，其余情况状态不变，因此期望为 3。

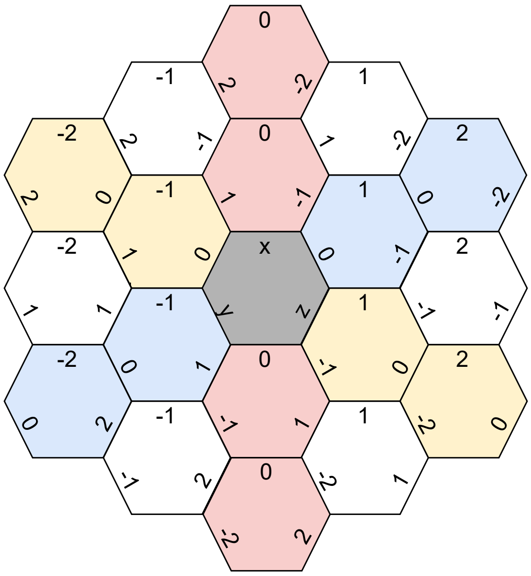
- 第三组测试数据：我们将初始状态表示为 $[1,3]$，共有 12 种不同操作，其中有 6 种操作在操作后状态为 $[4]$，其余 6 种操作在操作后状态为 $[2,2]$，因此期望为 $1+\frac{1}{2}\times 3=\frac{5}{2}$，即模 998244353 意义下的 499122179。
- 第四组测试数据：我们将初始状态表示为 $[1,1,2]$，共有 12 种不同操作，其中有 2 种操作在操作后状态为 $[4]$，其余 10 种操作在操作后状态为 $[2,2]$，因此期望为 $1+\frac{5}{6}\times 3=\frac{7}{2}$，即模 998244353 意义下的 499122180。

题目 I. 六边形翻转

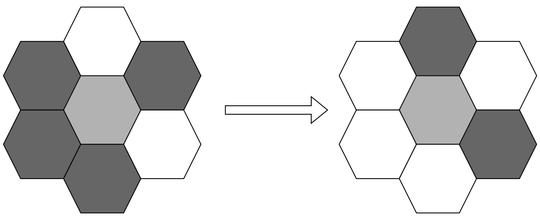
给出两个**无限大**六边形网格图，其中有些格点为黑色，有些格点为白色，如下图所示。



我们用一个三维坐标 (x, y, z) $(x, y, z \in \mathbb{Z}, x + y + z = 0)$ 来描述该网格图中的每个格点，具体如下图所示。



我们可以进行如下图所示的翻转操作，每次选择一个三维坐标 (x, y, z) $(x, y, z \in \mathbb{Z}, x + y + z = 0)$ ，将该坐标周围一圈的格点进行颜色翻转（黑色翻转为白色，白色翻转为黑色），即对格点 $(x, y-1, z+1)$ ，$(x+1, y-1, z)$ ，$(x+1, y, z-1)$ ，$(x, y+1, z-1)$ ，$(x-1, y+1, z)$ ，$(x-1, y, z+1)$ 进行颜色翻转。



问能否对第一个六边形网格进行若干次翻转操作，使其每个格点上的颜色均与第二个六边形网格相同。

输入

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 100$）表示测试数据组数。对于每组测试数据：

第一行包含两个整数 n, m $(0 \le n, m \le 10^5)$ ，分别表示两个六边形网格图上黑色格点的数量。

接下来 n 行第 i 行输入三个整数 x_i, y_i, z_i $(-10^9 \le x_i, y_i, w_i \le 10^9, x_i + y_i + z_i = 0)$，表示第一个六边形网格图上第 i 个黑点的坐标。

接下来 m 行第 i 行输入三个整数 u_i, v_i, w_i ($-10^9 \le u_i, v_i, w_i \le 10^9, u_i + v_i + w_i = 0$)，表示第二个六边形网格图上第 i 个黑点的坐标。

保证所有测试数据的 n 之和以及 m 之和均不超过 2×10^5。

输出

对于每组测试数据，输出 “YES” 如果第一个六边形网格可以进行若干次翻转操作，使其每个格点上的颜色均与第二个六边形网格相同，否则输出 “NO” 。你可以以任意形式输出答案（大写或小写），比如 “yEs”，“yes”，“Yes” 和 “YES” 都会被认为是肯定的答案。

样例

standard input	standard output
1 9 7 0 2 -2 -2 2 0 0 1 -1 2 0 -2 -1 0 1 2 -2 0 0 -2 2 0 0 0 -2 1 1 -1 1 0 1 1 -2 2 0 -2 2 -1 -1 0 -1 1 0 -2 2 1 -2 1	YES
2 5 3 0 0 0 -1 1 0 -1 0 1 0 -1 1 1 0 -1 0 0 0 0 1 -1 1 -1 0 4 3 -1 1 0 -1 0 1 0 -1 1 1 0 -1 0 0 0 0 1 -1 1 -1 0	YES NO

注释

样例 1 给出的两个格点图即图片顶部的两个格点图。

题目 J. 幻想乡的裁判长

图片来源：*Bad Apple!!* PV【影絵】

Shiki 是幻想乡的审判官，经常要纸笔记录很多文字。Shiki 发现，许多英文字母如 o, v, w，在书写时会形成连字。例如，两个连续书写的 v 连在一起形如 w；连续出现的 v 和 w 也会连在一起，例如 wvvwvwv 长得像一串长度为 10 的 v。

Shiki 认为一个字符串是好的，当且仅当它写在纸上的外观完全镜像对称。例如，wvowv 是镜像的，因为写在纸上是三个尖角、一个圆、三个尖角；而 vowow 不是对称的。

现在 Shiki 给了你她记录下来的一串字符串 s，保证 s 仅由 $\{\texttt{o}, \texttt{v}, \texttt{w}\}$ 内的字母组成。你需要找到 s 的一个最长的好的子串。

输入

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 2 \times 10^6$）表示测试数据组数。对于每组测试数据：

第一行输入一个整数 n（$1 \le n \le 10^7$），表示字符串 s 的长度。

第二行输入一个长度为 n 的字符串 s，保证仅由 $\{\texttt{o}, \texttt{v}, \texttt{w}\}$ 内的字母组成。

保证所有测试数据的 n 之和不超过 10^7。

输出

对于每组测试数据，输出一行字符串表示 s 最长的好的子串。

如果有很多解，你可以输出任意一个。

样例

standard input	standard output
3 8 wwwovvvv 16 wwwoooooooooovwww 11 wwwovoovvvv	wwovvvv ooooooooo vvvv

该页留空

题目 K. 01 背包

01 背包问题是一个算法竞赛中经典的组合优化的问题，小 w 学会了一种求解该问题的贪心算法。01 背包问题的定义以及小 w 的贪心算法如下。

01 背包问题

给定 n 个物品，物品的重量分别为正整数 $w_1, w_2, \ldots, w_n$，物品的价值分别为正整数 $v_1, v_2, \ldots, v_n$，再给定背包容量 W。要求 $x_1, x_2, \ldots, x_n$ ($\forall 1 \le i \le n,\ x_i \in \{0, 1\}$)，满足:

$$\sum_{i=1}^{n} w_i x_i \le W$$

并最大化:

$$V = \sum_{i=1}^{n} v_i x_i$$

贪心算法

1. 将 n 个物品按照 $\frac{v_i}{w_i}$ 的值从大到小排序，$\frac{v_i}{w_i}$ 相同则按照 w_i 从大到小排序。
2. 设一初始置 0 的变量 V_0，并从 1 到 n 枚举 i，如果 $V_0 + w_i \le W$，则置 $x_i \leftarrow 1, V_0 \leftarrow V_0 + w_i$，否则置 $x_i \leftarrow 0$。
3. 枚举完之后即可得到所求的 $x_1, x_2, \ldots, x_n$ 以及 V。

你当然知道这个算法是错误的，但小 w 并不相信。即使你给了小 w 一些反例，小 w 依然认为这个算法能在很多不同的 W 下都能得到最优的 V，所以你现在希望构造一组 $w_1, w_2, \ldots, w_n$ 以及 $v_1, v_2, \ldots, v_n$ 使得:

1. 对于任意的 $2 \le W \le W_{lim}$（W_{lim} 是一个给定的常数），小 w 的算法都无法得到最优的 V。
2. 在满足条件 1 的情况下，n 尽量小。
3. 在满足条件 1, 2 的情况下，$\max(w_1, w_2, \ldots, w_n)$ 尽量小。
4. 在满足条件 1, 2, 3 的情况下，$\max(v_1, v_2, \ldots, v_n)$ 尽量小。

现在你需要构造一组满足上述条件的 01 背包来说服小 w，你能做到吗? 如果构造方法有多种，你可以输出任意一种。

输入

输入共一行包含一个整数 W_{lim}（$2 \le W_{lim} \le 5 \times 10^3$），表示 W 的上界。

输出

输出第一行包含一个整数 n（$1 \le n \le 10^4$），表示构造的 01 背包的物品数量。

第二行输出 n 个整数 $w_1, w_2, \ldots, w_n$（$1 \le w_i \le W_{lim}$）表示物品的重量。

第三行输出 n 个整数 $v_1, v_2, \ldots, v_n$（$1 \le v_i \le 10^9$）表示物品的价值。

可以证明，在给定的问题以及输入条件下，总能找到满足上述数据范围的解。

样例

standard input	standard output
2	2 1 2 2 3

题目 L. 网格避障

给定一个 $n \times m$ 的网格，行从上到下编号为 1 到 n，列从左到右编号从 1 到 m。除第 1 列与第 m 列外，每一列至多有一个障碍物；第 1 列与第 m 列保证没有障碍物。

你可以选择从第 1 列的**任意一个**格子出发，目标是到达第 m 列的**任意一个**格子。假设你当前在第 i 行第 j 列，每一步你可以选择下列三种操作之一:

- 向右：从 (i, j) 走到 $(i, j + 1)$；
- 向上：从 (i, j) 走到 $(i - 1, j)$；
- 向下：从 (i, j) 走到 $(i + 1, j)$。

不允许向左移动，且任何时刻都不能进入障碍格子，也不能走出网格。

共有 k 个障碍（按列号从小到大排序后编号为 $i = 0$ 到 $k - 1$）。对每个障碍，你必须选择 "从上方绕过" 或 "从下方绕过"。

若第 i 个障碍在第 r_i 行 c_i 列:

- 选择 "从上方绕过" 时，你在列 c_i 时的行编号必须始终小于 r_i；
- 选择 "从下方绕过" 时，你在列 c_i 时的行编号必须始终大于 r_i。

不同障碍的选择相互独立，共有 2^k 种方案。

你的任务如下：对每一种方案，计算在满足该方案所有限制的情况下，从第 1 列某行到第 m 列某行的最小步数；若该方案下无可行路径，输出 -1。

输入

输入第一行包含一个整数 T（$1 \le T \le 5000$），表示测试数据组数。对于每组测试数据:

第一行包含两个整数 n, m（$1 \le n \le 100, 2 \le m \le 100$），分别表示网格的行数与列数。

第二行包含一个整数 k（$0 \le k \le \min(m - 2, 10)$），表示障碍的数量。

接下来 k 行第 i 行包含两个整数 r_i, c_i（$1 \le r_i \le n$，$1 < c_i < m$），表示在第 r_i 行第 c_i 列有一个障碍。保证 $1 < c_1 < c_2 < \ldots < c_k < m$。

对于所有测试数据，保证 $n \times m$ 之和不超过 5000，注意对于所有测试数据的 k 之和**并没有约束**。

输出

对于每组测试数据，输出一行，包含 2^k 个整数，依次表示编号从 0 到 $2^k - 1$ 的每一种方案的最小步数。相邻数字之间用一个空格分隔。若某种方案无解，则输出 -1。

方案编号说明：每种方案可看作一个长度为 k 的二进制数。二进制数从低位到高位依次对应第 $0, 1, \ldots, k - 1$ 个障碍。某一位为 0 表示从上方绕过该障碍；为 1 表示从下方绕过。

例如：若 $k = 3$，则共有 $2^3 = 8$ 种方案，对应如下:

方案编号	二进制表示（低位 $\rightarrow$ 高位）	绕行选择（第 0, 第 1, 第 2 障碍）
0	000	上, 上, 上
1	100	下, 上, 上
2	010	上, 下, 上
3	110	下, 下, 上
4	001	上, 上, 下
5	101	下, 上, 下
6	011	上, 下, 下
7	111	下, 下, 下

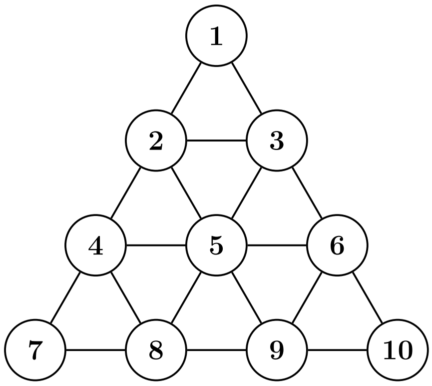
程序应按照方案编号从小到大的顺序输出每种方案的最小步数。

样例

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

题目 M. 连通的正三角形

A 镇上有 $\frac{(n+1)(n+2)}{2}$ 个路口，这些路口由 $3n$ 条**路径**连接，构成了一个边长有 $\boldsymbol{n}$ **条道路**的正三角形。$n = 3$ 的情况如下图所示：



这 $3n$ 条**路径**可以分成左斜，水平和右斜三个方位，每个方位各包含 n 条**路径**。每个方位的第 i 条**路径**由 i 条**道路**组成。

例如在 $n = 3$ 时：

- 左斜的**路径**从**道路**数量由少至多分别为 $(6 \leftrightarrow 9)$, $(3 \leftrightarrow 5 \leftrightarrow 8)$, $(1 \leftrightarrow 2 \leftrightarrow 4 \leftrightarrow 7)$。
- 水平的**路径**从**道路**数量由少至多分别为 $(2 \leftrightarrow 3)$, $(4 \leftrightarrow 5 \leftrightarrow 6)$, $(7 \leftrightarrow 8 \leftrightarrow 9 \leftrightarrow 10)$。
- 右斜的**路径**从**道路**数量由少至多分别为 $(4 \leftrightarrow 8)$, $(2 \leftrightarrow 5 \leftrightarrow 9)$, $(1 \leftrightarrow 3 \leftrightarrow 6 \leftrightarrow 10)$。

我们称三个**不同**的路口 (u, v, w) 是一个长度为正整数 l 的 “正三角形” 路口，当且仅当我们可以在**任意重排** u, v, w **三个点的顺序**后满足：

- 从 u 出发经过 l 条左斜的**道路**到达 v;
- 从 v 出发经过 l 条水平的**道路**到达 w;
- 从 w 出发经过 l 条右斜的**道路**到达 u。

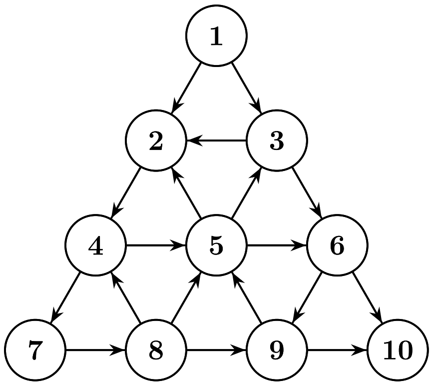
或者：

- 从 u 出发经过 l 条右斜的**道路**到达 v;
- 从 v 出发经过 l 条水平的**道路**到达 w;
- 从 w 出发经过 l 条左斜的**道路**到达 u。

例如上图中 $(2, 4, 5)$，$(2, 3, 5)$ 是 “正三角形” 路口，而 $(2, 3, 4)$ 不是。

A 镇为了防止交通过于拥堵，决定给每条**道路**定向。在定向之后，每条**道路**存在唯一固定的方向，且每条**路径**上的**道路**方向相同。

如下图是一种可能的定向情况（对应着样例 1 的第三组测试数据）：



称图中的三个不同的路口 (u, v, w) 为 "连通的正三角形" 路口，当且仅当它们在图中构成了一个长度为正整数 l 的 "正三角形" 路口，且它们可以通过构成该 "正三角形" 路口的 $3l$ 条**道路**相互到达。例如在上图中:

- $(2, 4, 5)$ 是一个 "连通的正三角形" 路口。因为它们不仅是 "正三角形" 路口，且只通过该 "正三角形" 路口的**道路** $(2 \to 4, 4 \to 5, 5 \to 2)$ 相互到达。
- $(2, 3, 4)$ 不是一个 "连通的正三角形" 路口。因为它们不是一个 "正三角形" 路口。
- $(2, 3, 5)$ 不是一个 "连通的正三角形" 路口。因为它们虽然是 "正三角形" 路口，但不能通过该 "正三角形" 路口的**道路** $(5 \to 3, 3 \to 2, 2 \leftarrow 5)$ 相互到达。

现在告知所有有向**路径**的方向，询问这个图中到底有多少个 "连通正三角形路口"。

输入

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^5$）表示测试数据组数。对于每组测试数据:

第一行输入一个整数 n（$1 \le n \le 10^5$），表示正三角形的长度。

第二行输入一个长度为 n 的字符串 $s1$，其中 $s1_i \in \{0, 1\}$，若 $s1_i = 0$ 则表示由 i 条**道路**构成的左斜**路径**的方向是右上向左下，反之亦然。

第三行输入一个长度为 n 的字符串 $s2$，其中 $s2_i \in \{0, 1\}$，若 $s2_i = 0$ 则表示由 i 条**道路**构成的水平**路径**的方向是左向右，反之亦然。

第四行输入一个长度为 n 的字符串 $s3$，其中 $s3_i \in \{0, 1\}$，若 $s3_i = 0$ 则表示由 i 条**道路**构成的右斜**路径**的方向是右下向左上，反之亦然。

保证所有数据的 n 之和不超过 10^5。

输出

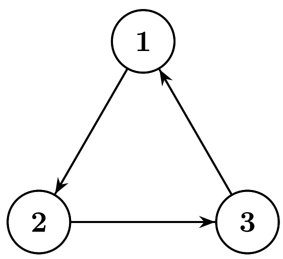
对于每组数据，输出一个整数，表示 "连通正三角形路口" 的数量。

样例

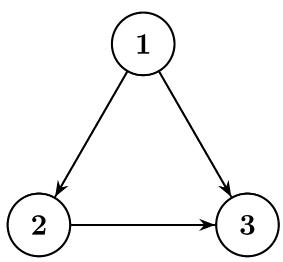
standard input	standard output
3 1 0 0 0 1 0 0 1 3 010 100 001	1 0 4
1 4 0011 1100 0011	6

注释

样例一中，第一组测试数据的图形：



样例一中，第二组测试数据的图形：



样例一中，第三组测试数据的“连通正三角形路口”有：

- $(2,4,5)$;
- $(4,7,8)$;
- $(5,6,9)$;
- $(2,7,9)$。

该页留空