

Problem A. 两条

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

给定平面直角坐标系中的两条直线:

$$y = k_1x + b_1$$

和

$$y = k_2x + b_2$$

再给定一条水平直线:

$$y = c$$

水平直线 $y = c$ 会分别与这两条直线相交于两个点。

请你求出这两个交点之间的距离。

Input

一行五个整数 k_1, b_1, k_2, b_2, c 。

其中第一条直线为 $y = k_1x + b_1$, 第二条直线为 $y = k_2x + b_2$, 水平直线为 $y = c$ 。

保证:

$$-10^4 \leq k_1, b_1, k_2, b_2, c \leq 10^4$$

$$k_1 \neq 0, \quad k_2 \neq 0$$

Output

输出一个实数, 表示两个交点之间的距离。

你的答案与标准答案的绝对误差或相对误差不超过 10^{-6} 即可通过。

Example

standard input	standard output
2 3 -4 5 7	2.5000000000

Problem B. 格兰法洛

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

在伊比利亚的沉寂之镇——格兰法洛，黄金时代的余晖早已被灰暗的阴云掩盖。蔓延的"溟痕"如同有呼吸的活物，正一点点吞噬着这座小镇残存的街道。

为了开辟通往海岸的道路，小 H 委托罗德岛干员卡卡来清理小镇主干道上的溟痕。这条主干道可以看作一条数轴，上面有 n 个待清理的区块，按位置编号为 $1, 2, \dots, n$ 。受深海侵蚀程度不同，第 i 个区块的溟痕浓度为 p_i ($1 \leq p_i \leq W$)。

工程部紧急调用了一台由伊比利亚前修道院侍从改装的清洁机器——"小帮手"。卡卡需要操控"小帮手"进行 m 次清理操作。第 i 次操作可以表示为三元组 (op_i, x_i, w_i) ，其中 w_i 为本次设定的清洁功率 ($1 \leq w_i \leq W$)：

- 若 $op_i = 0$ ，"小帮手"会从主干道左侧推进，清理所有满足 $1 \leq j \leq x_i$ 且浓度 $p_j < w_i$ 的区块 j ；
- 若 $op_i = 1$ ，"小帮手"会从主干道右侧推进，清理所有满足 $x_i \leq j \leq n$ 且浓度 $p_j < w_i$ 的区块 j 。

现已知所有区块的浓度 p_i 以及每次操作的方向 op_i 和范围 x_i ，请问有多少种为机器设定功率 (w_1, w_2, \dots, w_m) 的方案，能够使得每个区块至少被成功清理一次？

请输出满足条件的方案数对 $10^9 + 7$ 取模后的结果。两种方案不同，当且仅当存在至少一个 $i \in \{1, 2, \dots, m\}$ ，使得这两种方案设定的功率 w_i 不同。

Input

本题一个测试点内包含多个测试用例。

第一行输入一个正整数 T ($1 \leq T \leq 5 \times 10^3$) 表示测试用例的个数。对于每个测试用例：

第一行输入三个正整数 n, m, W ($1 \leq n, W \leq 5 \times 10^3$, $1 \leq m \leq 10^6$) 分别表示区块个数、清理次数以及溟痕浓度的上限。

第二行输入 n 个正整数 p_1, p_2, \dots, p_n ($1 \leq p_i \leq W$) 表示各个区块的溟痕浓度。

接下来 m 行，第 i 行输入两个正整数 op_i, x_i ($op_i \in \{0, 1\}$, $1 \leq x_i \leq n$) 表示第 i 次清理的参数。

保证所有测试用例的 n 之和不超过 5×10^3 ，所有测试用例的 m 之和不超过 10^6 。

Output

对于每个测试用例，输出一行一个整数，表示满足"所有区块至少被清理一次"的功率设定方案数，结果对 $10^9 + 7$ 取模。

Example

standard input	standard output
2	18
3 3 3	2897
1 2 1	
1 1	
1 2	
0 3	
5 5 6	
5 1 4 2 3	
0 4	
0 2	
0 1	
1 3	
0 5	

Problem C. 秋水横渡

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **256 megabytes**

随着武陵工业 325 期的开展，武陵地区的生态发生了巨大的变化。现如今，清波寨到武陵城的水路上按顺序分布着 n 座关口，编号依次为 $1, 2, \dots, n$ 。乘坐竹筏从 $i - 1$ 号关口航行至 i 号关口需要花费 t_i 单位的时间 ($2 \leq i \leq n$)。

每座关口均设有周期性的通行限制：关口会在"开启"和"关闭"两种状态之间交替切换，每次维持某一状态的时间均为 k 个单位时间（即一个完整的开闭周期长度为 $2k$ ）。当关口处于开启状态时，竹筏可以瞬间渡过（不花费时间）；当关口处于关闭状态时，竹筏必须在关口前等待，直到关口切换为开启状态方可通行。

作为终末地的干员，小 H 在伊冯的帮助下侵入了中枢系统。他拥有最高权限，可以在出发前，任意设定每一座关口的初始启动时间。然而，由于水路环境的复杂性与系统底层的不可控延迟，小 H 抵达起点的时机无法与系统精准同步。通俗地说：当小 H 在时刻 0 恰好到达 1 号关口前准备出发时，1 号关口刚好处于周期的哪个时刻是完全随机的。

形式化地说：

- 每个关口都有一个时钟 S_i ($S_i \in [0, 2k)$, $S_i \in \mathbb{R}$)。小 H 在到达 1 号关口前，可以任意设定所有 S_i ($1 \leq i \leq n$) 的值。
- 系统中存在一个不可控的全局未知偏移量 Δ ($\Delta \in \mathbb{R}$)，并且 Δ 在 $[0, 2k)$ 的连续区间内独立且均匀分布。
- 时钟随着时间的推移连续且均匀地流逝。在出发后的任意时刻 t ($t \geq 0$, $t \in \mathbb{R}$)， i 号关口的实际时钟值为 $T_i(t) = (S_i + \Delta + t) \bmod 2k$ 。
- 在时刻 t ，若 $T_i(t) < k$ ，则该时刻 i 号关口开启，允许通行；若 $T_i(t) \geq k$ ，则该时刻 i 号关口关闭，必须等待。

现在，小 H 需要乘坐竹筏依次渡过这 n 座关口前往武陵城。已知他会采取最完美的策略来设定所有的 S_i 以最小化行程耗时。请问从时刻 0 出发开始计时，直到小 H 成功渡过最后一座 (n 号) 关口，所花费总时间（包含航行时间与所有等待时间）的最小数学期望是多少？

Input

本题一个测试点内包含多组测试用例。

第一行输入一个整数 T ($1 \leq T \leq 10^5$) 表示测试用例的个数。对于每组测试用例：

第一行输入两个整数 n, k ($2 \leq n \leq 2 \times 10^5$, $1 \leq k \leq 10^9$) 分别表示关口数量和每座关口的状态切换时长。

第二行输入 $n - 1$ 个整数 t_2, \dots, t_n ($1 \leq t_i \leq 10^9$) 表示竹筏在相邻两座关口之间的通行时间。

保证所有测试用例的 n 之和不超过 2×10^5 。

Output

对于每个测试用例，输出一行实数表示花费总时间的数学期望，如果您的答案与标准答案的相对误差或绝对误差不超过 10^{-9} ，则认为您的答案是正确的。

具体来说，假设您的答案是 a_i ，标准答案是 b_i ，则当且仅当 $\frac{|a_i - b_i|}{\max(1, |b_i|)} \leq 10^{-9}$ 时，您的答案才会被接受。

Example

standard input	standard output
2	27.750000000
4 27	50.750000000
9 2 10	
6 51	
3 3 16 6 10	

Problem D. counter!

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

此方正在玩横版格斗游戏！

为了让自己更强，她现在在训练场研究连招。

具体而言，此方现在操作的角色有 s 种招式，根据造成伤害从大到小，用 $1 \rightarrow s$ 对这些招式标号。

现在此方已经研究出一套连招，需要使用 n 次招式，第 i 次释放第 c_i 种招式。

这套连招虽然能连起来，不过在对局中，招式随时都有可能因被敌人打康而中断，因此她希望尽可能早地打出高伤害的招式。

于是经过一番研究，她发现这些招式中，有些造成的效果是类似的（比如都是让敌人僵直）。她称这些招式为相容招式。

如果A与B相容，那么B一定与A相容。不过如果C也与B相容，A却不一定与C相容。（毕竟只是类似嘛）

如果两个招式是相容的，并且在连招中是相邻的，那么她可以交换这两个招式的顺序（而不影响连招）。

不过由于明天就要考试了，此方还要忙着临时抱佛脚，所以她希望你帮帮忙，为她修改这套连招，让高伤害的招式尽可能提前。换句话说，希望最终连招的字典序尽可能小。

Input

第一行，两个整数 s, n 。

接下来 s 行，每行一个长为 s 的 01 字符串。

其中第 i 行第 j 列的字符 $A_{i,j} = 1/0$ 表示两种招式相容/不相容。

下一行， n 个整数 c_1, c_2, \dots, c_n ，表示初始的连招序列。

Output

一行， n 个整数，表示通过交换能得到的最优的连招序列。

Example

standard input	standard output
3 6	1 2 3 3 3 1
110	
111	
011	
1 3 3 2 3 1	

Note

$1 \leq s \leq 60, 1 \leq n \leq 5 \times 10^5$ 。

保证 $A_{i,j} = A_{j,i}, A_{i,i} = 1$ 。

Problem E. 六个核桃

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

小核正在整理精品核桃礼盒，他将编号为 $1 \sim n$ 的核桃排成一列，对应位置编号也为 $1 \sim n$ 。最终的摆放顺序构成一个 $1 \sim n$ 的排列 p ，其中 p_i 表示第 i 个位置摆放的核桃编号。为了衡量礼盒的观赏价值，小核定义了单颗核桃的价值贡献为 $|p_i - i|$ 与 m 中较小的那个值。其中 p_i 表示第 i 个位置摆放的核桃编号， m 为给定的数。小核希望所有核桃的总价值分最大，请你构造满足条件的任意一个排列 p 。

Input

一行，输入两个整数 n, m 。

保证：

$$1 \leq n \leq 100000, \quad 1 \leq m \leq n$$

Output

输出一行 n 个整数，表示你构造出的排列 p 。

如果有多种合法答案，输出任意一种即可。

Example

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

Problem F. 杯级评定

Input file: **standard input**
Output file: **standard output**
Time limit: **1 second**
Memory limit: **256 megabytes**



都安静，全场杯级由我评定！

现在有 n 名干员在Dr.小E面前排成一列，其中第 i 名干员的DPS是 d_i 。

小E不希望这些干员的DPS差距太大，所以他让这些干员组成若干个小队。一个小队可以是单人成军，也可以是**相邻的**两个人组成。

他定义小队的杯级指数是组合中所有干员DPS的总和。

作为一名资深的兔头党，小E会通过计算所有小队的杯级指数极差，来大致评定这些小队的杯级差距。

如果存在杯级差距过大的干员，小E会考虑换下TA。不过这些干员跟了他很久了，他也不想丢下某个干员。

所以他想问你，在所有可能的组合方式，所有小队的杯级指数极差最小值是多少？

Input

第一行一个整数 n ，代表干员总数。

第二行 n 个整数，其中第 i 个数 d_i 表示第 i 名干员的DPS。

Output

一行一个整数，表示通过组合后，所有小队的杯级指数极差最小值。

Example

standard input	standard output
3 325 799 7999	6875

Note

数据保证 $n \leq 10^5, 1 \leq d_i \leq 10^9$ 。

Problem G. 排列

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

对于 $\{1, \dots, n\}$ 的一个排列 $\{p_1, \dots, p_n\}$, 称其为「好的」, 当且仅当其能通过以下操作变为 $\{1, \dots, n\}$:

- 枚举 $i = 1, \dots, n$:
 - 你可以将当前排列中值为 i 的元素与任一相邻元素交换;
 - 也可以不做任何操作。

求所有 n 阶好排列中, 字典序第 k 小者; 或判定如此排列不存在。

Input

本题有多组测试数据。

第一行包含一个整数 t , 表示测试数据组数。

接下来依次输入每组测试数据。对于每组测试数据:

输入一行两个整数 n, k , 分别表示所求排列长度和排名。

Output

对于每组测试数据:

- 若所求排列存在, 输出一行 n 个整数, 表示所求排列;
- 否则, 输出一行一个整数 -1 。

Examples

standard input	standard output
3	1 2
2 1	2 1
2 2	-1
2 3	
19	1 2 3
3 1	1 3 2
3 2	2 1 3
3 3	2 3 1
3 4	3 1 2
3 5	-1
3 6	1 2 3 4
4 1	1 2 4 3
4 2	1 3 2 4
4 3	1 3 4 2
4 4	1 4 2 3
4 5	2 1 3 4
4 6	2 1 4 3
4 7	2 3 1 4
4 8	2 4 1 3
4 9	3 1 2 4
4 10	3 1 4 2
4 11	4 1 2 3
4 12	-1
4 13	

Note

- $1 \leq t \leq 2 \times 10^5$;
- $1 \leq n \leq 10^3$;
- $\sum n \leq 2 \times 10^5$;
- $1 \leq k \leq 10^{18}$.

Problem H. Granate Raus!

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

作为一名资深的捞薯人，小E很清楚自己的枪法有限，解决距离不同的敌人所需要的子弹数不同。不过由于小E钟爱突击兵，因此他可以一枪不开，用手雷来解决敌人。

具体而言，现在有 n 名敌人，小E还剩下 m 颗子弹， k 个手雷。

第 i 名敌人需要用 c_i 颗子弹解决，或者用**一颗**手雷解决。

小E想问你，凭借现有的资源，他最多能解决多少名敌人？

Input

第一行有三个整数 n, m, k

第二行有 n 个整数 c_1, c_2, \dots, c_n ，表示解决每名敌人所需要的子弹数。

Output

一行一个整数，表示小E使用现有的资源能够解决的最多敌人数。

Example

standard input	standard output
5 31 1 3 25 11 45 14	4

Note

$n \leq 100, m \leq 100, k \leq 100, 1 \leq c_i \leq 100$

Problem I. Ich brauche Munition!

Input file: **standard input**
Output file: **standard output**
Time limit: **1 second**
Memory limit: **256 megabytes**

小E在战地V猛猛绕后捞薯条，但是由于小E是突击兵，现在他没有子弹了，需要去补给点拿摸你穷 (Munition, 弹药包)

具体而言，现在小E所处的区域可以看作一张 n 个点， m 条边的带权无向图，通过第 i 条边需要耗费 w_i 的时间。

当前小E处在顶点 1，而图中有 k 个补给点，第 i 个补给点在编号为 s_i 的点上。只要小E能够到达任何一个补给点，那么他就可以继续开捞。

不过路途总是困难的。从地图上看，每个顶点 u 附近有 d_u 个红点，代表着这里有 d_u 名敌人，他们会**随机**把守着与 u 相连的 d_u 条边。由于没有子弹，小E无法通过敌人把守的边。

并且由于战场的多变性，每次小E到达点 u ，敌人都有可能把守着**不同**的边。

为了小E的KD，也为了热心的萨尼铁塔不受到伤害，小E希望你计算最坏情况下，他最快多久能到达任意一个补给点？或者告诉他在最坏的情况下无法安全地到达补给点（那就只能重新部署了）。

Input

第一行三个整数 n, m, k

第二行输入 k 个整数 s_1, s_2, \dots, s_k ，表示所有补给点的编号。保证按照升序给出。

第三行输入 n 个整数 d_1, d_2, \dots, d_n ，表示每个点的敌人数量。

接下来 m 行，每行三个整数 u, v, w ，表示有一条连接 u, v 的无向边，边权为 w 。

Output

一行一个整数，表示在最坏情况下，小E到达补给点的最短时间。如果无法到达补给点，输出 -1 。

Examples

standard input	standard output
6 10 3 2 3 6 0 1 0 1 0 1 1 2 5 2 3 3 2 4 6 1 5 4 3 6 2 5 4 4 6 4 3 1 6 2 3 1 2 3 5 3	2
6 10 3 2 4 6 0 1 0 0 1 0 1 2 5 1 3 4 1 4 5 3 5 2 2 6 5 6 5 3 3 2 4 6 1 5 2 5 6 4 3 6	5

Note

数据范围保证 $1 \leq n, m \leq 2 \times 10^5, 1 \leq k \leq n, 0 \leq d_i \leq m, 1 \leq w_i \leq 10^6$
保证不会有重边或自环

Problem J. 三元组

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

给定正整数 n ，满足 $n \bmod 6 \in \{1, 3\}$ 。

请构造 $\frac{n(n-1)}{6}$ 个三元组 (a_i, b_i, c_i) ，使得 $\{1, \dots, n\}$ 中任意两个不同整数都同时出现在恰好一个三元组中。

Input

共一行一个整数 n ，含义如上。

Output

共输出 $\frac{n(n-1)}{6}$ 行。

第 i 行包含三个整数 a_i, b_i, c_i ，表示第 i 个三元组。

如果有多种合法构造，输出任意一种即可。

Example

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

Note

$1 \leq n \leq 3000$; $n \bmod 6 \in \{1, 3\}$ 。

Problem K. 抛硬币

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

小 Z 有很多枚硬币。

这些硬币被分成了 k 类。第 i 类有 m_i 枚硬币，每枚硬币正面朝上的概率都是

$$\frac{p_i}{q_i}.$$

所有硬币的抛掷结果相互独立。

现在小 Z 将每一枚硬币都抛一次。

请你求出：最终正面朝上的硬币数量为偶数的概率。

小 Z 有个非常喜欢的硬币，所以它一定是标准的。也就是说，存在某个 i ，满足

$$m_i \geq 1$$

且

$$\frac{p_i}{q_i} = \frac{1}{2}.$$

由于答案可能是分数，请你输出答案对 998244353 取模后的结果。

也就是说，如果答案为

$$\frac{x}{y},$$

你需要输出

$$x \cdot y^{-1}$$

对 998244353 取模后的结果，其中 y^{-1} 表示 y 在模 998244353 意义下的逆元。

Input

第一行包含一个整数 k ，表示硬币的种类数。

接下来 k 行，每行包含三个整数 m_i, p_i, q_i ，表示第 i 类有 m_i 枚硬币，每枚硬币正面朝上的概率为 $\frac{p_i}{q_i}$ 。

数据保证：

$$1 \leq k \leq 2 \cdot 10^5,$$

$$1 \leq m_i \leq 10^{18},$$

$$0 \leq p_i \leq q_i,$$

$$1 \leq q_i < 998244353,$$

$$\sum_{i=1}^k m_i \leq 10^{18}.$$

保证至少存在一个 i , 满足 $m_i \geq 1$ 且 $2p_i = q_i$ 。

Output

输出一个整数, 表示正面朝上的硬币数量为偶数的概率对 998244353 取模后的结果。

Example

standard input	standard output
3 2 1 3 1 1 2 5 1 7	499122177

Problem L. ? ! 11! ?

Input file: **standard input**
Output file: **standard output**
Time limit: **1 second**
Memory limit: **256 megabytes**

小E最近闲来无事，所以开始在草纸上写数字。

他从 0 开始写，一直写一直写，直到写到 $10^x - 1$ 才停下。

然后他开始好奇，这些数字里面，1 这个数字出现了多少次？

比如 1 在 114 中出现两次，在 514 中出现一次。

不过数字实在是太长了，看的他眼睛都花了。所以他想问你，1 这个数字究竟出现了多少次？

他写了很多次，每次他都想知道答案。答案可能很大，所以对 998244353 取模。

Input

第一行一个数字 T ，表示询问次数。

接下来 T 行，每行一个整数 x ，表示询问在 $[0, 10^x - 1]$ 中，1 出现了多少次。

Output

一共 T 行，每行一个整数，表示答案。

Example

standard input	standard output
2	0
0	20
2	

Note

数据范围保证 $1 \leq T \leq 10^6, 0 \leq n \leq 10^6$ 。