

Problem A. Another Searching Problem

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

你正在研究下面这个图算法。图有 n 个点，编号为 $1, 2, \dots, n$ ，并且每个点的邻居按照编号升序访问。

dp():

```
    令 q 为空队列
    令 id[1..n] 的初值均为 n+1
    counter = 0, timer = 0
    将 1 压入 q

    当 q 非空时:
        弹出队首到 u
        如果 id[u] 是 n+1:
            timer = timer + 1
            id[u] = timer

        按升序枚举 u 的每个邻居 v:
            counter = counter + 1
            如果 id[v] > id[u]:
                将 v 压入 q

    返回 counter
```

注意，同一个点可能被多次压入和弹出队列。

给定整数 k ，请构造一个包含 n 个结点和 m 条无向边的简单图，使得上述算法返回 k 。如果不存在这样的图，请报告无解。

每组数据的构造必须满足 $1 \leq n \leq 200, 1 \leq m \leq 300$ 。

Input

第一行包含一个整数 t ($1 \leq t \leq 2 \times 10^4$)，表示测试数据组数。

接下来 t 行，每行包含一个整数 k ($1 \leq k \leq 10^{18}$)。

Output

依次对每组数据输出答案。

如果不存在合法构造，输出一行一个整数 -1 。

否则，先输出一行两个整数 n 和 m ，分别表示所构造图的点数和边数。接下来输出 m 行，每行两个整数 u, v ，表示一条连接结点 u 与结点 v 的无向边。

输出的图必须是简单图：不允许自环，也不允许重复边。边的输出顺序任意；算法始终按照邻居编号升序访问。图不要求连通。

如果存在多种合法构造，输出任意一种即可。

Example

standard input	standard output
3	-1
1	2 1
2	1 2
8	3 3
	1 2
	1 3
	2 3

Note

对于第一组数据, $k = 1$ 无解。如果结点 1 没有邻居, 算法返回 0; 否则第一次弹出结点 1 时至少会将一个邻居压入队列, 而该邻居之后被弹出时又至少会访问一条边, 因此返回值至少为 2。

对于第二组数据, 图中只有一条边 (1, 2)。算法会先从结点 1 访问这条边, 再从结点 2 访问这条边, 故返回 2。

对于第三组数据, 输出的是一个三角形。算法依次弹出结点 1、结点 2、结点 3, 之后结点 3 还会被重复弹出一次。每次弹出都会访问 2 条边, 所以返回值为 8。

Problem B. Bingo Game

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

给定一个 n 行 n 列的网格。网格中有 m 个格子已经被点亮，第 i 个已点亮的格子位于第 x_i 行、第 y_i 列。你可以继续点亮任意尚未被点亮的格子。

你要求出，至少还需要点亮多少个格子，才能使网格中存在至少一行或一列，其包含的 n 个格子全部被点亮。

Input

第一行包含两个整数 n, m ($1 \leq n, m \leq 10^5$)。

接下来 m 行，第 i 行包含两个整数 x_i, y_i ($1 \leq x_i, y_i \leq n$)。

保证所有给出的格子坐标两两不同。

Output

输出一个整数，表示至少还需要点亮的格子数。

Examples

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

Note

在样例 1 中，第 2 列已有第 2, 3, 4 行的格子被点亮。只需再点亮格子 (1, 2)，第 2 列就会全部被点亮。

在样例 2 中，第 2 行的所有格子都被点亮，因此不需要再点亮任何格子。

Problem C. Drifting Toward a Promised Sky 2

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

「就算我们不抬头仰望，星空，也永远注视着我们。」

无数曾伴你同行的水母化作了点点星光，缓缓升入无垠的夜空。它们可以用无限长的数轴上的 n 个点表示，第 i 个点初始位于 a_i 。同一个位置可以有多个点。

Index 想要对这些点进行最多 $10^{218105633}$ 次演化（次数可以为 0），在一次演化中，她会：

1. 选择一个当前至少包含两个点的位置 x ；
2. 将 x 处的点数量减少 2；
3. 将 $x-1$ 和 $x+1$ 处的点数量各增加 1。

Index 想知道在操作结束后，所有点的位置构成的集合的 mex 的最小可能值。

注：对于一个整数集合 S ， $\text{mex}(S)$ 定义为最小的不属于 S 的非负整数。例如 $\text{mex}\{1, 2, 3\} = 0$ ， $\text{mex}\{0, 2, 5\} = 1$ 。

Input

本题有多组测试数据，第一行包含一个正整数 t ($1 \leq t \leq 10$)，代表数据组数。

对于每组数据，第一行包含一个正整数 n ($1 \leq n \leq 5 \cdot 10^5$)，第二行包含 n 个整数 $a_1, \dots, a_n$ ($0 \leq a_i \leq 10^9$)。保证所有测试数据中 n 的总和不超过 $5 \cdot 10^5$ 。

Output

对于每组测试数据，输出一行一个整数，表示可达到的最小 mex。

Example

standard input	standard output
2	4
8	0
6 1 6 5 3 2 0 6	
4	
0 0 1 0	

Note

对于第一组数据，可以证明，初始时的 mex 就是可达到的最小 mex。

对于第二组数据，可以依次进行如下操作：

$$\begin{aligned} [0, 0, 0, 1] &\rightarrow [-1, 0, 1, 1], \\ [-1, 0, 1, 1] &\rightarrow [-1, 0, 0, 2], \\ [-1, 0, 0, 2] &\rightarrow [-1, -1, 1, 2]. \end{aligned}$$

其中，红色表示本次操作所选择的两个点，蓝色表示本次操作新产生的点。

此时，集合的 mex 为 0，显然 mex 不可能小于 0，故其最小值为 0。

Problem D. Escape Root

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

给定一棵有 n 个结点的树，结点编号为 1 到 n ，树根为 1，每条边长度为 1。

有 m 个人。第 i 个人会在第 s_i 秒出现在结点 x_i ，并立刻以每秒 1 单位长度的速度沿最短路向结点 1 移动。

如果某一时刻，有至少两个人处在同一个位置，则这些人会同时消失。位置可以是结点，也可以是边的内部。

如果某个人到达结点 1 时没有与其他人相遇，则他成功逃脱，并立刻从树上移出。

请问哪些人会成功逃脱。

Input

第一行包含两个整数 n, m ($1 \leq n, m \leq 2 \cdot 10^5$)。

接下来 $n - 1$ 行，每行包含两个整数 u, v ($1 \leq u, v \leq n, u \neq v$)，表示树上有一条连接 u 和 v 的边。

接下来 m 行，每行包含两个整数 x_i, s_i ($1 \leq x_i \leq n, 0 \leq s_i \leq 10^9$)，表示第 i 个人出现的结点和出现时间。

Output

输出一个长度为 m 的 01 串。

第 i 个字符为 1 表示第 i 个人成功逃脱，为 0 表示第 i 个人消失。

Examples

standard input	standard output
5 4 1 2 1 3 2 4 2 5 4 0 5 0 3 0 2 3	0011
4 5 1 2 2 3 2 4 3 0 4 0 2 1 1 2 4 5	00011
2 2 1 2 2 0 1 1	00

Note

在样例 1 中，第 1 个人和第 2 个人分别从结点 4 和 5 出发，它们都在第 1 秒到达结点 2，因此同时消失。第 3 个人从结点 3 出发，第 1 秒到达根并逃脱。第 4 个人第 3 秒从结点 2 出发，第 4 秒到达根并逃脱。

在样例 2 中，前 3 个人在第 1 秒同时位于结点 2，因此全部消失。虽然第 4 个人第 2 秒出现在根结点，但前 3 个人已经在更早的时候消失，所以第 4 个人可以成功逃脱。

在样例 3 中，第 1 个人第 1 秒到达根结点，同时第 2 个人第 1 秒出现在根结点。两人相遇并同时消失。

Problem E. Grand Mex 2

Input file: standard input
Output file: standard output
Time limit: 2.5 seconds
Memory limit: 512 megabytes

有 n 个非负整数 $a_1, a_2, \dots, a_n$, 初始时它们全部等于 0。

给定一棵包含 n 个顶点的树。对于树中的每条边 (u, v) , 你必须恰好选择以下两种操作之一:

- 令 $a_u \leftarrow a_u + 1$;
- 令 $a_v \leftarrow a_v + 1$ 。

每条边恰好操作一次, 因此共有 2^{n-1} 种选择方案。不同的选择方案分别计数, 即使它们产生了相同的最终数组。

一个非负整数数组的 mex 定义为没有在该数组中出现的最小非负整数。例如 $\text{mex}\{1, 2, 3\} = 0$, $\text{mex}\{0, 2, 5\} = 1$ 。

求所有选择方案产生的最终数组的 mex 之和。答案对 998244353 取模。

Input

第一行包含一个整数 n ($2 \leq n \leq 150$)。

接下来 $n - 1$ 行, 每行包含两个整数 u, v ($1 \leq u, v \leq n, u \neq v$), 表示树中的一条边。保证给出的边构成一棵树。

Output

输出一个整数, 表示所有 2^{n-1} 种选择方案产生的最终数组的 mex 之和对 998244353 取模后的结果。

Examples

standard input	standard output
3 2 1 2 3	7
10 10 5 8 9 1 8 2 9 7 3 3 4 8 5 1 4 6 8	1566

Note

共有四种选择方案。其中一种方案得到的数组的 mex 为 1, 其余三种方案得到的数组的 mex 均为 2, 因此答案为 $1 + 2 + 2 + 2 = 7$ 。

Problem F. Light the Lamp

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

在一个 n 行 n 列的网格图上，每个格子中都有一盏灯。第 0 秒时，每一行、每一列都恰好有一盏灯亮着。更具体地说，第 i 行第 p_i 列的灯是亮着的，其中 p 是一个长度为 n 的排列。

之后时间按整秒变化。对于每盏尚未亮起的灯，如果在第 $t-1$ 秒时，与它有公共边的相邻格子中至少有两盏灯已经亮着，那么这盏灯会在第 t 秒亮起。已经亮起的灯不会熄灭。

你需要判断是否所有灯最终都会亮起；如果所有灯最终都会亮起，则你还要求出所有灯都亮起的最早时刻。

Input

第一行包含一个整数 n ($2 \leq n \leq 5 \cdot 10^5$)。

第二行包含 n 个两两不同的整数 $p_1, p_2, \dots, p_n$ ($1 \leq p_i \leq n$)，表示初始时第 i 行第 p_i 列的灯亮着。

Output

如果存在至少一盏灯永远不会亮起，输出 -1 。

否则输出一个整数，表示所有灯都亮起的最早时刻。

Examples

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

Note

在样例 1 中，初始亮灯位于主对角线上。每一秒会向两侧扩展一格，最远的两个角落会在第 3 秒亮起。

在样例 3 中，亮灯过程会提前停止，无法点亮整个网格。

Problem G. Passing Game

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

有 n 个人按顺时针顺序围成一个环，编号为 $1, 2, \dots, n$ 。第 1 个人的上一个人是第 n 个人，第 n 个人的下一个人是第 1 个人。

一次传球中，当前持球的人必须把球传给自己的上一个人或下一个人。

第 i 个人最多只能传出 a_i 次球，接球次数不受限制；如果某个人接到球后游戏立即结束，那么这次接球不会消耗他的传球次数。

你可以任意选择最开始持球的人，也可以在任意一次传球后结束游戏。请你求最多能进行多少次传球。

Input

第一行包含一个整数 n ($3 \leq n \leq 5 \cdot 10^5$)。

第二行包含 n 个整数 $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$)。

Output

输出一个整数，表示最多能进行的传球次数。

Examples

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

Note

在样例 1 中，可以按传出者编号依次为 $1, 3, 2, 1, 3, 2$ ，这样每个人都恰好传出 2 次，共传球 6 次。

在样例 2 中，编号为 2 和 5 的人不能传球，因此传球过程不能穿过他们。最优方案之一是让传出者依次为 $3, 4, 3$ ，共传球 3 次。

在样例 3 中，最优方案之一是让传出者依次为 $2, 3, 4, 5, 1, 5$ ，共传球 6 次。

Problem H. Set

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

给定一个长度为 n 的互不相同的整数序列 $a_1, a_2, \dots, a_n$, 记初始集合 $S = \{a_1, a_2, \dots, a_n\}$ 。
你可以重复进行如下操作:

1. 从 S 中任选一个元素 x 。
2. 将 x 拆分为两个正整数 u, v , 满足 $u + v = x$ 。
3. 依次对 S 进行对称差操作: $\oplus x, \oplus u, \oplus v$ 。

其中, 定义集合 $S \oplus y$ 为:

- 如果 $y \in S$, 则从 S 中删去 y ;
- 如果 $y \notin S$, 则向 S 中加入 y 。

问: 通过若干次 (可以为零次) 操作, 最终能够得到多少种只包含单个元素的集合 S ?

Input

第一行包含一个正整数 n ($1 \leq n \leq 10^5$)。

第二行包含 n 个正整数 $a_1, \dots, a_n$ ($1 \leq a_i \leq 10^9$)。

Output

输出一行, 包含一个整数表示答案。

Examples

standard input	standard output
2 2 4	2
3 2 3 4	2

Note

在样例 1 中, 选择 $x = 4, u = 2, v = 2$ 可以使 $S = \{2\}$, 选择 $x = 2, u = 1, v = 1$ 可以使 $S = \{4\}$ 。可以证明不存在其他只包含单个元素的集合 S 。

Problem I. Slay the Spire

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

你正在进行一场战斗。敌人初始有 x 点生命值，初始中毒层数为 0。
每个回合中，你必须从下面两种行动中选择一种：

1. **施加中毒**：使敌人的中毒层数增加 a ；
2. **普通攻击**：立即对敌人造成 b 点伤害。

在你的行动结束后，如果敌人的中毒层数大于 0，则立刻进行一次中毒结算：

1. 对敌人造成等于当前中毒层数的伤害；
2. 随后使敌人的中毒层数减少 1。

中毒层数不会在回合之间清空。特别地，本回合新施加的中毒会在本回合行动结束后参与中毒结算。
当敌人的生命值小于等于 0 时，敌人被击杀，战斗立即结束。发生击杀的回合也计入回合数。
请你求出，在采取最优策略的情况下，击杀敌人最少需要多少个回合。

Input

共一行，包含三个整数 x, a, b ($1 \leq x, a, b \leq 10^9$)，分别表示敌人的初始生命值、每次施加的中毒层数和普通攻击造成的伤害。

Output

输出一个整数，表示击杀敌人所需的最少回合数。

Examples

standard input	standard output
10 3 4	3
100 5 1	7

Note

在样例 1 中，一种最优策略如下：

- 第 1 回合施加 3 层中毒，中毒造成 3 点伤害，剩余 2 层中毒；
- 第 2 回合普通攻击造成 4 点伤害，中毒再造成 2 点伤害，剩余 1 层中毒；
- 第 3 回合普通攻击，击杀敌人。

前 2 回合内至多造成 9 点伤害，因此答案为 3。

在样例 2 中，前 6 回合内的最大伤害为 90 点。该伤害可以通过每回合都施加中毒达到，此时中毒伤害依次为 5, 9, 13, 17, 21, 25；第 7 回合继续施加中毒时会再造成 29 点伤害，因此第 7 回合可以击杀敌人。

Problem J. Start to the End

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

给定 n, m, a, b , 求有多少个满足以下条件的有向二分图:

- 左部点为 n 个有标号的点, 右部点为 m 个有标号的点。
- 每个点出度不超过 1, 左部点至多有 a 个零出度点, 右部点至多有 b 个零出度点。
- 从任意一个非零出度点出发, 不能走到零出度点。

由于答案可能很大, 所以你只需要输出答案对 2^{64} 取模的结果。

多测, t 组数据。

Input

本题有多组测试数据, 第一行包含一个正整数 t ($1 \leq t \leq 10^3$), 表示数据组数。

对于每组数据, 共一行, 包含四个整数 n, m, a, b ($0 \leq a \leq n \leq 10^7, 0 \leq b \leq m \leq 10^7$)。

Output

对于每组数据, 输出一行一个整数, 表示答案对 2^{64} 取模后的结果。

Example

standard input	standard output
6	1
1 1 0 0	2
1 1 1 1	16
2 2 0 0	28
2 2 1 1	150
2 3 1 2	1369
3 3 3 3	

Note

对于第一组数据, 左右两侧各有一个点, 且两个点的出度都不能为零。因此, 它们必须互相连边, 所以只有 1 个合法的有向二分图。

对于第二组数据, 两个点都可以没有出边。合法情况共有两种: 两个点都没有出边, 或两个点互相连边。因此, 答案为 2。

Problem K. Tree Reflection

Input file: standard input
 Output file: standard output
 Time limit: 8 seconds
 Memory limit: 512 megabytes

给定一个整数 n 和一个由 n 个正数组成的数组 a 。

有一个整数 x ，初始值为 0。你需要依次进行 n 次随机操作。

此外还会给出 n 棵有根外向树 $T_1, T_2, \dots, T_n$ 。对于每个 $1 \leq i \leq n$ ，树 T_i 的点集为 $\{0, 1, \dots, a_i\}$ ，根为 0，并且序列 $0, 1, \dots, a_i$ 是 T_i 的一个拓扑序。换言之，若 $u > 0$ ，则 u 在 T_i 中的父亲 $f_{i,u}$ 满足 $0 \leq f_{i,u} < u$ 。

还会给出 n 个整数 $b_1, b_2, \dots, b_n$ 。在第 i 次操作中：

- 以 $\frac{b_i}{998244353}$ 的概率将 x 增加 a_i ；
- 以 $1 - \frac{b_i}{998244353}$ 的概率将 x 减少 a_i 。

如果这次选择后 $x < 0$ ，令 $u = -x$ ，则 x 会立刻被替换为 $f_{i,u}$ 。由于操作前 x 一定非负，因此一旦出现 $x < 0$ ，必有 $1 \leq u \leq a_i$ ，所以上述父亲一定存在。

请你求出所有 n 次操作结束后， x 的期望。

由于答案是一个有理数，你需要输出它对 998244353 取模后的结果。更具体地，若答案为不可约分数 $\frac{A}{B}$ ，你需要输出 $A \cdot B^{-1} \bmod 998244353$ 。可以证明 B 与 998244353 互质。

Input

第一行包含一个整数 n ($1 \leq n \leq 5 \cdot 10^5$)。

第二行包含 n 个正整数 $a_1, a_2, \dots, a_n$ ($1 \leq \sum_{i=1}^n a_i \leq 5 \cdot 10^5$)。

第三行包含 n 个整数 $b_1, b_2, \dots, b_n$ ($0 \leq b_i \leq 998244353$)。

接下来 n 行，第 i 行包含 a_i 个整数 $f_{i,1}, f_{i,2}, \dots, f_{i,a_i}$ ($0 \leq f_{i,u} < u$)，表示树 T_i 中每个非根点的父亲。

Output

输出一行一个整数，表示最终 x 的期望对 998244353 取模后的结果。

Example

standard input	standard output
2	499122178
2 1	
499122177 499122177	
0 1	
0	

Note

第一次操作后， x 有 $\frac{1}{2}$ 的概率为 2，有 $\frac{1}{2}$ 的概率为 1。

第二次操作选择加法的概率也为 $\frac{1}{2}$ 。最终：

- $x = 3$ 的概率为 $\frac{1}{4}$ ；
- $x = 1$ 的概率为 $\frac{1}{4}$ ；
- $x = 2$ 的概率为 $\frac{1}{4}$ ；

- $x = 0$ 的概率为 $\frac{1}{4}$ 。

因此期望为 $3 \cdot \frac{1}{4} + 1 \cdot \frac{1}{4} + 2 \cdot \frac{1}{4} = \frac{3}{2}$ ，在模 998244353 意义下等于 499122178。

Problem L. XOR Team

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

有一个 10^{18} 长度的初始全为 0 的 01 序列 b 。

有一个长度为 $2 \cdot n$ 的序列 a ，你需要将其分为 n 组 x_i, y_i ，并且需要满足 $x_i \leq y_i$ ，然后对 b 序列进行 n 次操作，每次将 x_i 到 y_i 的所有数异或上 1。

你需要求出任意分组后 b 序列操作后的和的最小值。

Input

本题有多组测试数据，第一行包含一个正整数 t ($1 \leq t \leq 10^5$)，代表数据组数。

对于每组数据，第一行包含一个正整数 n ($1 \leq n \leq 10^6$)，第二行包含 $2n$ 个正整数 $a_1, a_2, \dots, a_{2n}$ ($1 \leq a_i \leq 10^{18}$)。

保证所有测试数据中 n 的总和不超过 $2 \cdot 10^6$ 。

Output

对于每组测试数据，输出一个非负整数，表示任意分组后 b 序列操作后的和的最小值。

Example

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

Note

对于第一组测试数据，只有一种分组方式，即将 2 和 5 分为一组。对应的操作区间为 $[2, 5]$ ，操作后位置 2, 3, 4, 5 的值均为 1，因此答案为 4。

对于第二组测试数据，一种最优分组方式为 (1, 4) 和 (2, 3)。执行操作后，位置 1 和 4 的值为 1，位置 2 和 3 的值为 0，因此答案为 2。

对于第三组测试数据，一种最优分组方式为 (1, 3)、(1, 5) 和 (5, 5)。执行操作后，位置 1 到 5 的值依次为 0, 0, 0, 1, 0，因此答案为 1。