

2026~2027 NUIT 新生选拔赛

NUIT 2026~2027 Freshman Selection Contest



2026 年 10 月 11 日

October 11, 2026

题目列表 Problem List

ID	题目 Title	时间 Time	空间 Memory
A	X-GCD	2.0 s	512 MB
B	Combo	2.0 s	512 MB
C	检查点	2.0 s	512 MB
D	组合数乘积	3.0 s	512 MB
E	环形操作	2.0 s	512 MB
F	鼠鼠的开销	2.0 s	512 MB
G	迟到的色彩刚刚好	3.0 s	512 MB
H	野树的曙	2.0 s	512 MB
I	安全传输	2.0 s	256 MB
J	划分区间	2.0 s	512 MB
K	exString	2.0 s	512 MB
L	MEX 子数组	2.0 s	512 MB
M	巴巴博弈	2.0 s	512 MB

本次比赛共计 13 道题目

13 problems in total

Problem A. X-GCD

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

hushuqi 和 Liuguang_Ji 正在讨论数论问题，Liuguang_ji 随口说了一句求 $\gcd(a^b, b^a)$ ，其中 $\gcd(x, y)$ 表示 x 与 y 的最大公因数。
hushuqi 发现这个问题很简单，于是他决定把这个问题交给你。
具体地，你需要回答 $\gcd(a^b, b^a)$ 的值，这个数可能很大，你需要把答案对 998244353 取模后再输出。

Input

第一行输入一个正整数 T ($1 \leq T \leq 10^5$)，表示询问次数。
接下来 T 行，每行输入两个正整数 a, b ($1 \leq a, b \leq 10^6$)，表示一次询问。

Output

对于每次询问，输出一行一个整数，表示答案。

Example

standard input	standard output
4	27
12 3	3125
5 10	304170860
114 514	306982982
91 78	

Problem B. Combo

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

小 K 手上有 n 张卡牌，每张卡牌的能量值分别为 $a_1, a_2, \dots, a_n$ 。
现在到小 K 的回合，小 K 可以打出若干张卡牌，打出去的总能量是每张打出卡牌能量的乘积，现在小 K 可以打出任意数量张卡牌（可以不打，此时能量为 0）使得总能量尽可能大。
小 K 想知道这个总能量有多大，由于答案可能很大，你需要将答案对 998244353 取模*后再输出。

Input

第一行输入 1 个正整数 n ($1 \leq n \leq 2 \times 10^5$)，表示卡牌数量。
第二行输入 n 个正整数 a_i ($1 \leq a_i \leq 10^9$)，表示每张卡牌的能量。

Output

共输出 1 行，表示答案对 998244353 取模后的结果。

Examples

standard input	standard output
4 2 3 4 5	120
4 1000000000 1000000000 1000000000 1000000000	433041149

*取模： x 对 y 取模表示求 x 除以 y 的余数。例如：5 对 3 取模等于 2。

Problem C. 检查点

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

给定 n 个检查点，每个检查点坐标分别为 $p_1, p_2, \dots, p_n$ ，Yezhka 在场地上放了 m 枚棋子，其坐标为 $x_1, x_2, \dots, x_m$ 。

现在 Yizhka 可以操作若干次，对于每次操作，仅能将其中一枚之前未移动过的棋子，可以移动到新的位置，移动到距离为 d 的位置需要消耗 d^2 的代价。已知 Yizhka 最多能消耗 k 点代价，使得所有棋子都位于检查点内。

很明显，不是每局游戏 Yizhka 都能使所有棋子到达检查点，于是他决定尝试放宽检查点判定，他可以只需要把棋子移动至检查点附近即可。

现在，他可以将所有检查点都放宽至 $[p_i - r, p_i + r]$ 的范围之中。在他可以修改检查点范围的前提下，他想知道他需要至少放宽判断范围 r 才能胜利。（由于游戏存储限制， r 仅可以是非负整数值）

Input

第一行输入 3 个正整数 n, m, k ($1 \leq n, m \leq 2 \times 10^5, 0 \leq k \leq 10^{16}$)。

第二行输入 n 个整数 p_i ($0 \leq |p_i| \leq 10^8$)，表示检查点坐标。

第三行输入 m 个整数 x_i ($0 \leq |x_i| \leq 10^8$)，表示每枚棋子的初始位置。

Output

共输出一行 1 个非负整数，表示至少需要放宽的半径 r 。

Examples

standard input	standard output
1 3 15 10 5 9 16	3
12 23 998244353 12199216 90749213 -62833415 -38258866 -14446715 -35650520 -49600331 33618438 -54860314 5014791 -97029528 -23175106 -68729860 96557776 89224060 26314437 8623611 -10558853 78829390 -20209361 -62085211 -50453124 82907520 -22446844 -47205672 74576319 -130016 -78318307 -58367447 -44618411 -93888793 -80921259 39756497 292969 -69694719	16141299

Note

对于第一组样例：

当 $r = 3$ 时，检查点检查范围扩大至 $[7, 13]$ ，此时，符合条件的移动方案是 $5 \rightarrow 7, 16 \rightarrow 13$ ，此时需要的代价为 $4 + 9 = 13$ ，可以证明没有更优的方案。

Problem D. 组合数乘积

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

小文是一个数学苦手，对于一些组合数求和他一窍不通。
现在他有一个由 n 个组合表达式 $C_{a_1}^{b_1} \cdot C_{a_2}^{b_2} \cdots C_{a_n}^{b_n}$ ，其中 $C_x^y = \frac{x!}{y!(x-y)!}$ ，你需要帮小文计算他们的乘积，这个结果可能很大，你需要输出对 m 取模后的结果即可。

Input

第一行输入两个正整数 n, m ($1 \leq n \leq 10^6, 2 \leq m \leq 10^9$)，表示需要求和的项数和模数。
接下来 n 行，每行输入两个非负整数 a_i, b_i ($0 \leq b_i \leq a_i \leq 10^6$)，表示第 i 项组合数。

Output

输出一个非负整数，表示答案。

Examples

standard input	standard output
3 114514 19 1 9 8 1 0	171
8 998244353 24 11 28 19 34 2 92 30 289 12 200 39 76 73 325 25	535371832
3 999999999 1000000 500000 1000000 114514 1000000 1000000	72955971

Problem E. 环形操作

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

Papy 手里有一个大小为 n 的环形数组 $a_1, a_2, \dots, a_n$ 。刚开始的时候，数组里的每一个元素都是 1（即对于所有的 $1 \leq i \leq n$ ，都有 $a_i = 1$ ）。

环形数组的下标按照顺时针循环排列，即 a_n 的下一个元素是 a_1 ， a_1 的上一个元素是 a_n 。

Papy 现在可以进行若干次操作。每次操作中，你可以选择一个下标 i ($1 \leq i \leq n$)，并执行以下修改：

- $a_{i-1} \leftarrow a_{i-1} - 1$
- $a_i \leftarrow a_i + 2$
- $a_{i+1} \leftarrow a_{i+1} - 1$

（特别地，当 $i = 1$ 时， $i - 1$ 对应的就是 n ；当 $i = n$ 时， $i + 1$ 对应的就是 1。）

Papy 希望每次调整后，数组里的每一个元素都必须保持是非负整数。这意味着，只有当位置 i 的左右两个邻居都至少为 1 时，你才可以对位置 i 进行操作。当再也找不到任何可以合法调整的位置时，过程就会自然停止。

Papy 希望这个操作过程能尽可能多地玩几轮。请你帮帮他，设计一种能让总操作次数达到最多的方案，并输出最终停下来时数组的样子。如果有多种方案都能达到最多的操作次数，输出其中任意一种合法的最终状态即可。

Input

第一行输入一个正整数 T ($1 \leq T \leq 10^4$)，表示测试组数。

接下来的 T 组测试中，每组输入格式如下：

- 第一行输入一个整数 n ($3 \leq n \leq 2 \times 10^5$)，表示环形数组的大小。

此外，数据保证所有测试组 n 的总和不超过 2×10^5 。

Output

对于每组测试，输出一行包含 n 个用空格隔开的整数，代表在操作次数最多的前提下，最终数组各个位置的数值 $a_1, a_2, \dots, a_n$ 。

如果有多种合法的最终状态都能达到最大的操作次数，你只需要输出其中的任意一种最终状态即可。

Example

standard input	standard output
1 3	3 0 0

Problem F. 鼠鼠的开销

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

鼠鼠是一名精打细算的大学生。每个月，他的生活费由三部分组成：基础生活费 n 元、上班天数 m 天以及每天的工资 k 元。因此，鼠鼠每个月的总预算为 $S = n + m \times k$ 元。

这个月，鼠鼠计划完成 t 件事，每件事都有一个对应的开销。第 i 件事 ($1 \leq i \leq t$) 的开销为 a_i 元。

鼠鼠决定按照事情发生的先后顺序（即索引 i 从 1 到 t 的顺序）来考虑这些事。他的消费策略非常简单：对于当前考虑的第 i 件事，如果他当前剩余的钱大于或等于 a_i ，他就会选择做这件事，并扣除相应的金额 a_i ；否则，他只能无奈地跳过这件事，转而考虑下一件事。

请你帮助鼠鼠记录下，按照这个策略，他最终能完成哪些事。

Input

第一行包含一个整数 T ($1 \leq T \leq 100$)，表示测试数据的组数。

接下来 T 组数据，输入格式如下：

- 第一行包含四个整数 n, m, k, t ($0 \leq n, m, k \leq 100, 1 \leq t \leq 100$)，分别代表基础生活费、上班天数、每天工资以及事情的总数。
- 第二行包含 t 个整数 $a_1, a_2, \dots, a_t$ ($1 \leq a_i \leq 100$)，表示每件事的开销。

数据保证所有测试数据的 t 之和不超过 100，保证每组测试数据中至少有 1 件事可以被选择（即初始总预算 S 至少大于等于某一件事情的开销）。

Output

对于每组测试数据，输出一行，包含若干个由空格隔开的整数。这些整数是鼠鼠能够完成的事的编号（按照从小到大的顺序排列）。

Example

standard input	standard output
1 10 2 5 5 15 6 4 1 10	1 3 4

Note

对于样例 1：

基础生活费 $n = 10$ ，上班 $m = 2$ 天，工资 $k = 5$ 。总预算 $S = 10 + 2 \times 5 = 20$ 元。

- 第 1 件事：开销 $a_1 = 15$ 。当前余额 $20 \geq 15$ ，可以做。余额变为 $20 - 15 = 5$ 元。

- 第 2 件事：开销 $a_2 = 6$ 。当前余额 $5 < 6$ ，无法做，跳过。
- 第 3 件事：开销 $a_3 = 4$ 。当前余额 $5 \geq 4$ ，可以做。余额变为 $5 - 4 = 1$ 元。
- 第 4 件事：开销 $a_4 = 1$ 。当前余额 $1 \geq 1$ ，可以做。余额变为 $1 - 1 = 0$ 元。
- 第 5 件事：开销 $a_5 = 10$ 。当前余额 $0 < 10$ ，无法做，跳过。

最终完成的事的编号为 1,3,4 。

Problem G. 迟到的色彩刚刚好

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

Yvonne 想要画一幅作品，她在为这幅画寻找灵感。

四号谷地有 n 个颇有特色的地点，且两点之间形成了 $n-1$ 条路径，经过每条路径都需要消耗 1 分钟，且恰好形成一棵树，其中有部分地点能够为 Yvnone 提供灵感，我们称之为**有价值的地点**。

Yvonne 提出了 q 个方案，你需要回答从 x 号点到 y 号点且至少经过一个**有价值的地点**最少需要多少分钟？

Input

第一行输入一个正整数 $T(1 \leq T \leq 10^4)$ ，表示测试组数。

接下来 T 组测试中：

- 第一行输入两个正整数 n, q ($1 \leq n, q \leq 2 \times 10^5$)，含义见题意。
- 第二行输入长度为 n 的 01 串，表示第 i 号点是否为**有价值的地点**，当 s_i 为 1 时表示该地点为有价值地点，数据保证至少存在一个**有价值的地点**。
- 接下来的 $n-1$ 行，每行输入一对 u, v ($1 \leq u, v \leq n, u \neq v$)，表示一条连通点 u 和点 v 的路径，数据保证给出的边一定是一棵树。
- 接下来的 q 行，每行输入一对 x, y ($1 \leq x, y \leq n$)，表示一次询问。

此外，数据保证所有测试点 n 的总和不超过 2×10^5 ， q 的总和不超过 2×10^5 ，每个测试点的 01 串一定存在 1。

Output

对于每组测试，共输出一行 q 个非负整数，表示每次查询的答案。

Example

standard input	standard output
3	3 0
3 2	1 2 2
001	5 2 0 5 6 3
1 2	
2 3	
2 1	
3 3	
5 3	
01100	
1 2	
1 3	
3 4	
4 5	
1 3	
3 5	
1 1	
7 6	
1000010	
1 2	
2 3	
3 4	
3 5	
3 6	
4 7	
5 7	
4 6	
1 1	
4 7	
7 7	
1 6	

Problem H. 野树的曙

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

小曙有一棵包含 n 个节点的无根树 T , 节点编号为 $1, 2, \dots, n$ 。同时, 他手中有一个包含 n 个正整数的多重集 $S = \{s_1, s_2, \dots, s_n\}$ 。

小曙需要为树上的每个节点 u 分配一个来自集合 S 的点权 w_u , 使得集合 S 中的每个元素恰好分配给一个节点。

对于树上的任意一对有序节点 (x, y) , 定义其路径权值 $f(x, y)$ 为节点 x 到节点 y 的简单路径上所有节点的点权之和 (包含端点 x 和 y)。

请你帮小曙确定一种点权分配方案, 使得整棵树的总路径权值之和最大, 即最大化:

$$\sum_{u=1}^n \sum_{v=1}^n f(u, v)$$

Input

第一行输入一个正整数 T ($1 \leq T \leq 10^4$), 表示测试组数。

接下来的 T 组测试中, 输入格式如下:

- 第一行输入一个正整数 n ($2 \leq n \leq 2 \times 10^5$), 表示节点个数。
- 第二行输入 n 个正整数 $s_1, s_2, \dots, s_n$ ($1 \leq s_i \leq 10^9$), 表示多重集 S 。
- 接下来 $n-1$ 行, 每行共输入两个正整数 x, y ($1 \leq x, y \leq n, x \neq y$), 表示一条树边, 数据保证给出的树边一定是一棵树。

此外, 数据保证所有测试点 n 的总和不超过 2×10^5 。

Output

对于每组测试, 共输出一行:

- 共输出 n 个正整数, 第 i 个正整数表示节点 i 的点权 w_i 。

如果存在多种合法解, 输出任意一种合法方案即可通过本题。

Example

standard input	standard output
4	2 4 1
3	5 8 7 1
2 1 4	10 12 33 9 15 6 10 5
1 2	6 9 5 8 4 7 3 10 2 1
2 3	
4	
5 7 8 1	
1 2	
2 3	
3 4	
8	
10 5 6 15 9 12 33 10	
1 2	
2 5	
4 5	
5 3	
8 7	
3 7	
3 6	
10	
1 2 3 4 5 6 7 8 9 10	
1 2	
2 4	
4 9	
8 3	
8 7	
5 4	
2 8	
6 8	
6 10	

Note

对于样例二：
一种最优的点权分配方案： $w_1 = 1, w_2 = 8, w_3 = 7, w_4 = 5$ 。
计算部分代表性有序点对的简单路径权值：

- $f(1, 1) = 1, \quad f(2, 2) = 8, \quad f(3, 3) = 7, \quad f(4, 4) = 5$
- $f(1, 2) = f(2, 1) = 1 + 8 = 9$

- $f(2, 3) = f(3, 2) = 8 + 7 = 15$
- $f(1, 4) = f(4, 1) = 1 + 8 + 7 + 5 = 21$

将所有 $4 \times 4 = 16$ 个有序点对的路径权值 $f(u, v)$ 全部累加，可得总权值和的最大值为 207。

Problem I. 安全传输

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

这是一个 **Run-Twice** (通信) 问题。在这些问题中, 你的程序将运行两次。存储在内存中的所有变量都将在运行之间丢失, 但在第一次运行中提供给你的信息对于在第二次运行中正确解决问题可能很重要。因此, 主要的挑战是找到一种策略, 使用您可以使用的有限输出在两次运行之间进行通信。

时间限制和内存限制在运行之间不共享。例如, 在这个问题中, 由于时间限制是 2 秒, 因此只有当任一运行超过 2 秒时, 您才会收到 **Time Limit Exceeded** 的结果, 但如果程序的两次运行都持续 1.5 秒, 则完全没问题。

第一次运行时, 系统会给出一个正整数 x , 你需要加密成一个仅由小写字母组成的字符串 s , 且字符串长度不得超过 50。

第二次运行时, 系统会根据你给出的字符串 s , 随机打乱顺序后返回给你, 你需要通过给出的字符串, 复原出原先的数字 x 。

Input

第一次运行:

第一行输入一个字符串 **first**, 这样做的目的是让程序识别出这是第一次运行程序, 此时你需要完成加密功能。

第二行输入一个正整数 $T(1 \leq T \leq 10^4)$, 表示测试组数。

接下来 T 组测试中, 每组测试仅输入一个正整数 $x(1 \leq x \leq 10^9)$ 。

第二次运行:

第一行输入一个字符串 **second**, 这样做的目的是让程序识别出这是第二次运行程序, 此时你需要完成解密功能。

第二行输入一个正整数 $T(1 \leq T \leq 10^4)$, 表示测试组数。

接下来 T 组测试中, 每组测试仅输入选手第一次运行给出的字符串 s , 但顺序被随机打乱。

Output

第一次运行:

对于每组测试:

- 仅输出一个由小写字母组成的非空字符串 s , 且长度不超过 50。

第二次运行:

对于每组测试:

- 仅输出一个正整数 x , 表示解密后的结果。

注意:

在每一组测试输出完毕后，你需要刷新缓存区，使得交互程序能够读取你的结果并给出下一组测试数据，否则可能会导致 `Idleness limit exceeded`。

你可以通过一下方式刷新缓存区：

- **C++:** `fflush(stdout)` 或 `cout.flush()`
- **Python:** `sys.stdout.flush()`
- **Java:** `System.out.flush()`
- **Pascal:** `flush(output)`
- 其他语言见各自语言的帮助文档。

Examples

standard input	standard output
first 3 11 45 14	eleven fortyfive fourteen
second 3 elevne ofvfrteiy reetfoun	11 45 14

Note

注意：样例仅作为输入输出实例演示。

Problem J. 划分区间

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

给定两个长度为 n 的序列 $a_1, a_2, \dots, a_n$ 和 $b_1, b_2, \dots, b_n$ 。现在你可以将数组划分成若干段非空的不交区间，将这些区间按下标从小到大编号为 $1, 2, \dots, k$ ，其中第 i 个区间 $[l, r]$ 的权值为 $\sum_{j=l}^r a_j \cdot b_i$ 。
现在，请求出对于序列 a 恰好划分出 $k = 1, 2, \dots, n$ 个区间的情况下，通过划分可能得到的最大权值之和是多少？

Input

第一行输入一个正整数 T ($1 \leq T \leq 10^3$)，表示测试组数。
接下来 T 组数据中，每组数据输入格式如下：

- 第一行输入一个正整数 n ($1 \leq n \leq 5000$)，表示序列长度。
- 第二行输入 n 个整数 $a_1, a_2, \dots, a_n$ ($0 \leq |a_i| \leq 10^6$)，表示 a 序列。
- 第三行输入 n 个整数 $b_1, b_2, \dots, b_n$ ($0 \leq |b_i| \leq 10^6$)，表示 b 序列。

此外，数据保证 n^2 的总和不超过 5000^2 。

Output

对于每组测试，每组测试输出一行 n 个整数，第 i 个整数表示恰好划分成 i 个区间的最优划分能得到最大权值。

Example

standard input	standard output
4	-2 7 23
3	1330 11891 9259
1 -5 2	0 12 7 -30
1 -2 6	3100 160 190 34 34
3	
11 45 14	
19 198 10	
4	
1 -2 -3 4	
-1 2 3 -4	
5	
0 -1 2 0 30	
100 2 3 1 1	

Problem K. exString

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

给定一个串 s ，你需要构造一个长度恰好为 n 的字符串 t ，使得 t 存在子串[†] s 。

Input

第一行输入一个正整数 $n(1 \leq n \leq 10^4)$ ，表示需要构造的字符串 t 的长度。

第二行输入一个由小写字母组成字符串 $s(1 \leq |s| \leq n)$ ，其中 $|s|$ 表示字符串 s 的长度。

Output

输出一个长度为 n 且仅由小写字母组成的字符串 t 。

本题可能存在多组符合条件的答案，输出任意一种即可通过本题

Example

standard input	standard output
12 judge	specialjudge

[†]子串：字符串 S 的一个子串是由 S 中连续的一段字符组成的序列。具体地，对于字符串 $S = s_1s_2 \dots s_n$ ，其子串 $S[i \dots j]$ ($1 \leq i \leq j \leq n$) 定义为字符串 $s_is_{i+1} \dots s_j$ 。

Problem L. MEX 子数组

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

给定一个由 $0 \sim n-1$ 组成的排列[‡] p , 请求出对于排列 p 所有非空子数组[§] b 的 MEX[¶] 值总和。

Input

第一行输入一个正整数 T ($1 \leq T \leq 10^4$), 表示测试组数。

接下来每组测试数据, 第一行输入一个正整数 n ($1 \leq n \leq 10^6$), 表示排列长度。

第二行输入 n 个非负整数 a_i ($0 \leq a_i < n$), 表示 $0 \sim n-1$ 的排列。

数据保证所有测试组 n 的总和不超过 10^6 。

Output

对于每组数据, 输出一个整数, 表示答案。

[‡]排列: 对于一个 $0 \sim n-1$ 排列, 排列中仅有恰好一个整数 $x \in [0, n-1]$ 。例如:

- $[1, 0, 2, 3]$ 是一个排列。
- $[1, 1, 2, 0]$ 不是一个排列, 因为 1 出现了超过一次。
- $[0, 3, 1, 4]$ 不是一个排列, 因为 2 没有出现。

[§]对于一个长度为 m 的数组 c , 选择 2 个正整数 $1 \leq l \leq r \leq m$, 则所有取出下标为 $l \leq i \leq r$ 的元素 c_i , 并按原顺序排列组成的序列, 则叫数组 c 的子数组。其中区间长度不为 0 的子数组, 为非空子数组。

[¶]MEX: 对于序列 a , 它的 MEX 值表示最小未出现的自然数。例如:

- $[1, 2, 0]$ 的 MEX 值为 3, 因为 3 是序列中最小未出现的自然数。
- $[1, 1, 4, 5, 1, 4]$ 的 MEX 值为 0。
- $[0, 1, 2, 5, 5, 4]$ 的 MEX 值为 3。

Example

standard input	standard output
9	18
5	12
2 4 1 0 3	23
4	19
1 0 2 3	82
6	52
5 1 2 4 0 3	35
6	43
1 0 5 4 3 2	68
10	
8 5 3 1 9 0 6 4 2 7	
9	
2 7 6 3 0 8 1 4 5	
10	
1 8 6 5 7 3 4 2 0 9	
12	
11 0 7 4 5 8 10 9 3 2 6 1	
15	
1 12 0 14 3 9 8 4 13 6 7 10 11 2 5	

Note

对于第二组数据，所有连续子数组的 mex 值计算如下：

- 长度为 4: $\text{mex}([1, 0, 2, 3]) = 4$
- 长度为 3: $\text{mex}([1, 0, 2]) = 3$, $\text{mex}([0, 2, 3]) = 1$
- 长度为 2: $\text{mex}([1, 0]) = 2$, $\text{mex}([0, 2]) = 1$, $\text{mex}([2, 3]) = 0$
- 长度为 1: $\text{mex}([1]) = 0$, $\text{mex}([0]) = 1$, $\text{mex}([2]) = 0$, $\text{mex}([3]) = 0$

将上述所有结果相加: $4 + 3 + 1 + 2 + 1 + 0 + 0 + 1 + 0 + 0 = 12$ 。

Problem M. 巴巴博弈

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

Alice 和 Bob 正在玩一个游戏，游戏开始时，一共有 n 堆石子，每堆石子的数量分别为 $a_1, a_2, \dots, a_n$ ，Alice 和 Bob 轮流做以下操作：

- 选择若干堆石子，取走其中一颗石子。
- 如果本轮取完剩下所有石子，则获胜。

假设 Alice 和 Bob 每轮操作均采用最优策略的情况下，假设 Alice 先手，她想知道能否获胜。

Input

第一行输入一个正整数 T ($1 \leq T \leq 10^4$)，表示测试组数。

接下来的 T 组测试，每组测试个数如下：

- 第一行输入一个正整数 n ($1 \leq n \leq 5 \times 10^5$)，表示石子堆数量。
- 第二行输入 n 个正整数 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$)，表示每堆石子的数量。

此外，数据保证所有测试组 n 的总和不超过 5×10^5 。

Output

对于每组测试，输出格式如下：

- 如果 Alice 可以获胜，则输出 Alice，否则输出 Bob。

Example

standard input	standard output
5	Bob
1	Alice
4	Alice
2	Bob
1 2	Alice
3	
11 45 14	
6	
1000000000 1000000000 1000000000	
1000000000 1000000000 1000000000	
9	
9 9 8 2 4 4 3 5 3	