题目 A. 舞者僵尸

月黑风高之夜，僵尸博士为了攻破戴夫的防线，派出了一支由 n 个舞者僵尸组成的先遣队。当动感的乐曲响起时，舞者僵尸们将按照预定的节奏在数轴上依次登场，并开始向周围扩散。

具体规则如下：

- 第 i 个舞者僵尸会在第 a_i 秒时出现在数轴上的位置 b_i。
- 每一个位于坐标 x 的舞者僵尸，在出现一秒后，及其后的每一秒，都会在相邻的两个整数坐标 $x-1$ 和 $x+1$ 处各召唤出一名新的舞者僵尸。新召唤出的舞者僵尸同样具有上述召唤能力，且同一个坐标位置允许同时存在多个舞者僵尸。

僵尸博士的目标是让舞者僵尸覆盖整个战场。请问最少需要多少秒，才能使数轴上 $[0,k]$ 区间内的每一个整数坐标点都至少存在一个舞者僵尸？

输入

第一行输入两个整数 n,k（$1 \le n \le 10^5, 0 \le k \le 10^{12}$），分别代表舞者僵尸的总数以及需要覆盖的区间右端点。

接下来 n 行，每行两个整数 a_i,b_i（$0 \le a_i,b_i \le k$），分别代表第 i 个舞者僵尸出现的时刻与初始位置。

输出

输出一行一个整数，代表舞者僵尸填满数轴上 $[0,k]$ 区间的所有整数坐标所需要的最短时间。

样例

standard input	standard output
3 5 0 0 0 5 1 2	2
2 102 0 102 6 66	72

注释

对于第一个样例，每一秒结束时，数轴上 $[0,5]$ 区间内各位置的舞者僵尸的数量如下：

- 第 0 秒：$\{1,0,0,0,0,1\}$；
- 第 1 秒：$\{1,1,1,0,1,1\}$；
- 第 2 秒：$\{2,3,2,2,2,2\}$。

在第 2 秒的时候，数轴上 $[0,5]$ 区间内的每一个整数坐标上都有至少一个舞者僵尸。

题目 B. 梦中星河

每个人都有自己可望而不可及的星河。

在梦中，星河向你展示了一个非负整数 x，你需要找到三个非负整数 a, b, c，满足 $a \times b + c = x$。记这三个数中的最大值为 $M = \max(a, b, c)$，最小值为 $m = \min(a, b, c)$，你的目标是使极差 $M - m$ 尽可能小。请你求出在所有满足条件的 (a, b, c) 三元组中，$M - m$ 的最小可能值。

输入

第一行一个整数 T（$1 \le T \le 10^5$），表示数据组数。

对于每组数据，一行一个整数 x（$0 \le x \le 10^9$），代表星河给出的数字。

输出

对于每组数据，输出一行一个整数，表示 $M - m$ 的最小值。

样例

standard input	standard output
4 1 12 1000 729320	1 0 8 5

题目 C. 二进制串

给定一个长度为 n 的 01 串 $S = S_1S_2 \dots S_n$。你需要在每两个相邻的数字 S_i 与 S_{i+1} 之间（共 $n-1$ 个位置）各插入一个位运算符，使生成的表达式最终运算结果为 0。

可供选择的运算符有三种，其运算规则如下：

- 按位与（`&`）：对于表达式 a `&` b，此表达式的值为 1 当且仅当 $a = b = 1$，否则为 0。
- 按位异或（`^`）：对于表达式 a `^` b，此表达式的值为 1 当且仅当 $a = 1, b = 0$ 或 $a = 0, b = 1$，否则为 0。
- 按位或（`|`）：对于表达式 a | b，此表达式的值为 0 当且仅当 $a = b = 0$，否则为 1。

注意，本题中位运算的优先级与 C 语言一致，规则如下：

1. 按位与（`&`）的优先级最高。
2. 按位异或（`^`）的优先级低于按位与，但高于按位或。
3. 按位或（`|`）的优先级最低。
4. 对于优先级相同的运算，按照自左向右的顺序结合计算。

你需要构造出一组合法的运算符序列，使得整个表达式最终的运算结果等于 0。如果存在多种方案，输出任意一种即可。可以证明对于所有合法的输入总能找到至少一组合法的构造方案。

输入

第一行包含一个正整数 n（$2 \le n \le 10^5$），表示字符串的长度。

第二行包含一个长度为 n 且仅由 0 和 1 组成的字符串 S。

输出

输出一行一个长度为 $n-1$ 的字符串，表示依次插入的运算符。如果有多种合法方案，输出任意一种即可。

样例

standard input	standard output
3 000	&&

注释

样例中原串为 `000`，可在中间加入两个按位与使表达式变为 `0&0&0`，其结果为 `0`，因此输出 `&&`。

题目 D. 括号凸包

给定二维平面上的 n 个点，所有点互不相同，且不存在三点共线。每个点上写有一个括号，可能是左括号 (，也可能是右括号)。

我们称一个字符串 S 是一个合法括号序列，当且仅当它可以由如下递归规则生成:

- 空串是一个合法括号序列;
- 如果 A 是合法括号序列，那么字符串 (A) 也是合法括号序列;
- 如果 A 和 B 都是合法括号序列，那么 AB 也是合法括号序列。

例如，(), (()), ()() 都是合法括号序列，而)(, (()), ())(不是合法括号序列。

现在，你需要从给定的点中选出若干个互不相同的点作为顶点，构成一个凸多边形。

在本题中，一个由 m 个点 $p_{a_1}, p_{a_2}, \ldots, p_{a_m}$ 构成的多边形被称为凸多边形，当且仅当满足:

- $m \geq 3$;
- 这些点按照 $a_1, a_2, \ldots, a_m$ 的顺序依次连接，并连接 a_m 与 a_1 后，形成一个简单多边形（即多边形的边仅在相邻边的端点处相交，不相邻边互不相交）;
- 对于该多边形的每一条边，其余所有顶点都严格位于这条边所在直线的同一侧。

换句话说，所选出的点必须恰好按照它们在自身凸包上的环形顺序排列，并且所有内角都严格小于 $180°$，凸多边形上不存在三点共线。

对于一个凸多边形，任选其边界上的一个顶点作为起点，并沿着多边形边界按顺时针或逆时针方向依次遍历所有顶点，最后回到起点前停止。这样可以得到一个长度为 m 的括号序列：若当前顶点上写有左括号，则写下 (；若当前顶点上写有右括号，则写下)。

你的任务是找到一个包含左括号的凸多边形，使得对于该多边形上的任意一个写有左括号的顶点，以它作为起点沿多边形边界并以任意方向遍历得到的括号序列都是合法括号序列。

如果存在这样的凸多边形，请输出任意一个；否则输出无解。

输入

第一行包含一个整数 T（$1 \leq T \leq 400$），表示数据组数。

每组数据的第一行包含一个整数 n（$1 \leq n \leq 2000$），表示点的数量。

接下来 n 行，每行包含三个整数 x_i, y_i, t_i（$0 \leq x_i, y_i \leq 10^9, t_i \in \{0, 1\}$），表示第 i 个点的坐标和括号类型，其中 $t_i = 0$ 表示左括号，$t_i = 1$ 表示右括号。

保证每组数据中所有点互不相同，且不存在三点共线。

保证所有数据的 n 之和不超过 2000。

输出

对于每组数据，如果无解，输出一行一个整数 -1。

否则，第一行输出一个整数 m，表示选出的凸多边形的顶点数量。第二行输出 m 个两两不同的整数 $a_1, a_2, \ldots, a_m$，表示选出的点的编号。 这些点必须按照凸多边形的边界顺序输出，可以顺时针，也可以逆时针。

样例

standard input	standard output
5 4 1 1 0 2 4 1 3 9 0 4 16 1 5 1 1 0 2 4 0 3 9 0 4 16 1 5 25 1 6 47 58 0 30 23 0 27 34 1 35 7 1 10 30 1 1 25 1 8 10 5 0 10 16 0 1 5 0 24 9 0 6 2 0 6 12 0 7 18 1 3 13 1 1 0 0 0	4 1 2 3 4 -1 4 3 2 4 1 -1 -1

题目 E. 高维几何

当一个几何问题扩展到三维甚至更高维时，其求解难度会显著增加。为了应对复杂的 n 维几何结构，研究人员提出了一种模型：将空间中的 2^n 个关键位置抽象为图的顶点，并将位置之间的某种特定几何约束关系抽象为图中的边。

你现在就得到了一个抽象成图之后的模型，在这个模型中，复杂的几何特征被转化为一张包含 2^n 个顶点且无重边、无自环的简单无向图。但由于一些原因，你无法直接观测到图中的边。你想知道该图中边的总数 m。

你可以使用一台「子集感应探测器」，该探测器的工作原理如下：

- 你向探测器输入一个包含 k 个顶点的子集 S。探测器要求每次探测的集合大小必须 **恰好**为图中总点数的一半，即 $k = 2^{n-1}$。
- 探测器会扫描这些点及其关联的边，并返回图中 **至少有一个端点**属于集合 S 的边的总数。

由于代价过于昂贵，探测器最多只能使用 2^{n+1} 次。请在受限的查询次数内，求出图中的边的总数 m。

交互协议

首先，从标准输入中读取一个整数 n（$1 \le n \le 10$），表示图包含 2^n 个顶点，顶点编号从 1 到 2^n。

要进行一次查询，请向标准输出中输出 `? s`，其中 s 是一个长度为 2^n 的 01 字符串。字符串的第 i 个字符表示：

- 若为 `1`，表示选择第 i 个顶点进入子集 S；
- 若为 `0`，表示不选择。

你必须保证字符串中 `1` 的数量恰好为 2^{n-1}。如果你进行了非法查询（如集合大小错误）或超过了查询次数限制，交互器将立即终止你的程序并判定为错误。

每次查询后，你需要从标准输入中读入一个整数 d，表示至少有一个端点在所选子集 S 中的边的数量。

当确定答案后，向标准输出输出 `! m`，其中 m 为你计算得到的总边数。输出答案后，你需要立即终止程序。

交互器是 **非自适应的**。这意味着图在交互开始之前就已经固定，并且不会根据你的查询发生改变。

注意：每次输出后必须换行，并刷新标准输出缓冲区。刷新方式如下：

- 对于 C 或 C++，使用 `fflush(stdout)` 或 `cout.flush()`
- 对于 Java，使用 `System.out.flush()`
- 对于 Python，使用 `stdout.flush()`

样例

standard input	standard output
2 1 0	? 1010 ? 0101 ! 1

题目 F. 进行一个数的数

小 L 手中有一个长度为 n 的非负整数数组 $a = [a_1, a_2, \ldots, a_n]$，数组中的元素可以重复，且可以是任意非负整数（即 $a_i \geq 0$，没有上限）。小 L 选择了一个正整数 m，并将数组 a 中的每个元素对它取模，得到了一个新的数组 b，即 $b_i = a_i \bmod m$。

现在小 L 将数组 b 告诉你，但原数组 a 和模数 m 都是保密的。你想知道，在所有可能的原数组 a 和模数 m 的组合中，原数组的 mex 值有多少种不同的可能取值。

mex 定义为：一个数组中最小的未出现过的非负整数。例如，$\text{mex}\{0, 1, 3\} = 2$，$\text{mex}\{1, 2, 3\} = 0$。

输入

第一行一个整数 n（$1 \leq n \leq 10^5$），表示数组的长度。

第二行 n 个整数 $b_1, b_2, \ldots, b_n$（$0 \leq b_i \leq 10^9$），表示取模后得到的数组 b。

输出

输出一个整数，表示原数组 a 可能的 mex 取值的个数。

样例

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

注释

对于样例，所有可能的 mex 值有 $\{0, 1, 2, 3, 4\}$。以 $\text{mex} = 3$ 为例，一种可能的方案是 $m = 5$，$a = \{0, 1, 2, 8, 6, 7\}$。

题目 G. 传球谜题

一支球队由 n 名球员（编号 1 至 n）和一名守门员组成，他们正在进行传球训练。

训练流程如下：

1. 守门员首先开球，将球等概率地传给 n 名球员中的任意一人。此时所有球员的传球次数均为 0。
2. 球员得到球后，根据特定的偏好将其传出。若球员 i 持球，他将球传给球员 j 的概率为 $p_{i,j}$。
3. 每当球被一名球员踢出，该球员的累计传球次数增加 1。如果某次传球后，**所有 n 名球员的传球次数都恰好为偶数**，守门员将立刻出动将球停下，本轮训练宣告结束。

现在守门员想知道，在停球时，球员们总传球次数的期望是多少。

输入

第一行一个正整数 n（$1 \le n \le 16$），表示球员个数。

接下来 n 行，每行 n 个整数 $a_{i,j}$（$0 \le a_{i,j} < 998244353$）。球员 i 得球后传给球员 j 的概率定义为：$p_{i,j} = \frac{a_{i,j}}{\sum_{k=1}^{n} a_{i,k}}$。输入保证对于任意 $1 \le i \le n$，均有 $(\sum_{k=1}^{n} a_{i,k}) \not\equiv 0 \pmod{998244353}$。

输出

如果期望次数收敛，请输出答案对 998244353 取模后的结果；反之，输出 `infinity`。

可以证明如果期望次数收敛，则这个期望次数一定可以表示为有理数 $\frac{p}{q}$（p, q 为整数，$p \ge 0$ 且 $q \ge 1$）的形式。你需要输出 $p \cdot q^{-1} \bmod 998244353$ 的值，其中 q^{-1} 指 q 在模 998244353 意义下的逆元，满足 $q \cdot q^{-1} \equiv 1 \pmod{998244353}$。

样例

standard input	standard output
5 1 0 0 0 0 2 1 0 0 0 3 0 1 0 0 4 0 0 1 0 5 0 0 0 1	infinity
5 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0	10
6 5 6 499122175 10 10 998244352 7 998244351 7 4 499122176 3 4 10 2 9 8 9 6 6 6 3 10 10 998244351 3 1 499122175 10 9 3 7 6 499122176 499122176 3	53368493
1 8	2

题目 H. 金银岛

Sail 现在在金银岛上。他预知到金银岛上将会发生 n 个事件，并且他会在所有事件发生完毕后撤离。具体而言，有如下三种事件：

1. 获得一些金币：对于每一枚金币，他要么将其兑换为 k 块钱，要么投喂给财神爷以换取一颗伤害为 k 的子弹；
2. 获得一些银币：对于每一枚银币，他要么将其兑换为 1 块钱，要么投喂给财神爷以换取一颗伤害为 1 的子弹；
3. 出现一头怪兽：每当出现一头血量为 x 的怪兽，Sail 必须立刻使用手中若干颗子弹攻击怪兽，要求所选子弹的伤害总量之和 **大于等于** x。若无法满足该条件，Sail 将被怪兽杀死。子弹一旦使用即被消耗，不能重复使用。

Sail 想知道他能否安全撤离，如果能，撤离时最多能剩余多少钱。

输入

第一行包含一个正整数 T ($1 \le T \le 10^4$)，表示测试数据组数。

对于每组数据，第一行包含两个正整数 n ($1 \le n \le 2 \times 10^5$) 和 k ($1 \le k \le 10^6$)，其中 n 为事件总数，k 为金币兑换的钱数或子弹伤害值。

接下来 n 行，每行包含两个正整数 o ($o \in \{1, 2, 3\}$) 和 x ($1 \le x \le 10^6$)，描述一个事件。

- 若 $o = 1$ 或 $o = 2$，则 x 表示出现的金币或银币数量；
- 若 $o = 3$，则 x 表示怪兽的血量。

保证所有数据的 n 之和不超过 2×10^5。

输出

对于每组数据，如果 Sail 无法安全撤离，输出 -1；否则输出撤离时最多能剩余的钱数。

样例

standard input	standard output
5 4 3 1 2 3 5 2 3 3 3 4 3 1 2 2 2 3 5 3 1 4 3 1 2 2 2 3 5 3 3 4 3 1 2 3 5 2 2 3 3 4 3 1 2 1 3 3 4 3 9	0 1 0 -1 0

注释

对于第二组样例，Sail 依次做了：

1. 出现了两枚金币。Sail 用它们换取了两颗伤害为 3 的子弹。
2. 出现了两枚银币。Sail 用一枚银币换取了一颗伤害为 1 的子弹，另一枚换取了 1 块钱。
3. 一只生命值为 5 的怪物出现。Sail 使用两颗伤害为 3 的子弹将其击杀。
4. 一只生命值为 1 的怪物出现。Sail 使用一颗伤害为 1 的子弹将其击杀。

最后，Sail 带着 1 块钱撤离了。

对于第四组样例，Sail 依次做了：

1. 出现了两枚金币。Sail 用它们换取了两颗伤害为 3 的子弹。
2. 一只生命值为 5 的怪物出现。Sail 使用两颗伤害为 3 的子弹将其击杀。
3. 出现了两枚银币。Sail 用它们换取了两颗伤害为 1 的子弹。
4. 一只生命值为 3 的怪物出现。Sail 现在拥有的子弹伤害不足 3，所以他会被杀死。

无论如何兑换，Sail 都无法安全撤离，因此输出 -1。

题目 I. 聪明猪的考验

小 H 是一头会使用人工智能的聪明猪。今天他正在参加「小猪智商测试大赛」，但卡在了最后一道压轴题上：

- 给出两个由小写字母组成的字符串 s 和 t，请你找出它们所有长度为 2 的公共子序列中，字典序最小的一个。

字符串 a 是字符串 b 的子序列，当且仅当 a 可以由 b 删除若干个字符（可以不删除）且不改变剩余字符的相对顺序得到。

作为小 H 的人工智能助手，请你帮帮他。

输入

第一行一个整数 T（$1 \le T \le 2 \times 10^4$），代表测试数据组数。

对于每组测试数据，分两行各给出一个字符串 s 和 t（$1 \le |s|, |t| \le 10^5$）。保证所有字符串 **仅由小写字母构成**。

保证对于所有测试数据，$|s| + |t|$ 之和不超过 2×10^5。$|s|$ 和 $|t|$ 分别表示字符串 s 和 t 的长度。

输出

对于每组测试数据，如果存在长度为 2 的公共子序列，输出其中字典序最小的一个；如果不存在，输出 `HENG!`。

样例

standard input	standard output
3 azzza zazaz azzza zbzbz you ak	aa zz HENG!

题目 J. 宝石商人

小 S 非常喜欢玩桌游，尤其喜欢宝石商人。现在他对规则做了一些简化和修改。

游戏包含以下组件：

- 筹码：共有 5 种颜色，编号为 $1, 2, 3, 4, 5$，每种颜色的筹码数量均可视为无限多。
- 卡牌：共有 n 张卡牌，按顺序堆放形成 **牌堆**，卡牌编号依次为 $1, 2, \ldots, n$。每张卡牌 i 具有一个颜色 c_i（$1 \le c_i \le 5$）和兑换所需的五种筹码数量 $p_{i,1}$、$p_{i,2}$、$p_{i,3}$、$p_{i,4}$、$p_{i,5}$。

游戏开始时，将牌堆顶的前 4 张牌（卡牌 1 到 4）翻开放在桌面上，作为初始的 **可兑换牌**。此时牌堆中剩余卡牌为 5 到 n。

每个回合，你可以执行以下三种操作之一：

1. 选择一种颜色，拿取 2 个该颜色的筹码。
2. 选择三种不同的颜色，各拿取 1 个该颜色的筹码。
3. 从桌面的 **可兑换牌**中选择一张卡牌 i，支付对应筹码并将其兑换。当你兑换卡牌 i 时，所需支付的每种颜色 k 的筹码数量受 **此前已兑换卡牌**的折扣：若你之前已兑换了 d_k 张颜色为 k 的卡牌，则兑换当前卡牌时，颜色 k 实际需要支付的筹码数为 $\max(0, p_{i,k} - d_k)$。

每当你成功兑换一张桌面上的 **可兑换牌**后：

1. 将该卡牌从桌面移走；
2. 如果牌堆中还有未翻开的卡牌，则按顺序从牌堆顶补上 1 张到桌面作为 **可兑换牌**；
3. **当前手中所有剩余筹码清空**（**注意这条规则和原版规则不同**）。

你的目标是使 **牌堆**为空。注意，此时桌面上可能仍然剩下一些尚未兑换的 **可兑换牌**，这种情况也视为完成目标。请你计算，最少需要多少个回合才能达成目标。

输入

第一行一个整数 n（$4 \le n \le 100$），表示卡牌张数。

接下来 n 行，每行 6 个整数。第 i 行为 c_i、$p_{i,1}$、$p_{i,2}$、$p_{i,3}$、$p_{i,4}$、$p_{i,5}$，表示第 i 张卡牌的颜色和所需筹码数（$1 \le c_i \le 5$, $0 \le p_{i,j} \le 10^9$）。

输出

输出一个整数，表示使牌堆为空最少需要的回合数。

样例

standard input	standard output
5 1 4 1 1 7 1 1 2 2 8 4 8 1 2 3 6 7 2 1 6 5 7 3 1 1 9 9 9 9 9	7
7 2 15 34 0 15 24 4 11 20 0 18 0 1 4 0 2 48 20 3 17 0 8 43 70 1 0 58 7 1 52 1 64 0 81 40 0 2 4 0 0 0 10	86

注释

对于样例 1：对于前四张卡牌，通过拿取筹码操作攒够对应需求分别需要 6，8，7，8 次操作。

最优策略是：先耗费 6 次操作拿取筹码以满足第一张卡牌的需求，第 7 次操作执行「兑换卡牌」。兑换完成后，牌堆中的第 5 张卡牌（最后一张）被补入桌面，此时牌堆为空。总计 7 回合。

具体步骤如下：

1. 拿取颜色 1、4、5 的筹码各 1 个。
2. 拿取颜色 1、3、4 的筹码各 1 个。
3. 拿取颜色 1、4、5 的筹码各 1 个。
4. 拿取颜色 2、4、5 的筹码各 1 个。
5. 拿取颜色 1、2、4 的筹码各 1 个。
6. 拿取 2 个颜色 4 的筹码。经过前 6 个回合，手中的筹码数量为：颜色 1 有 4 个，颜色 2 有 2 个，颜色 3 有 1 个，颜色 4 有 7 个，颜色 5 有 3 个，足以支付第一张卡牌的筹码需求 $(4, 1, 1, 7, 1)$。
7. 选择第一张卡牌，支付相应的筹码并将其兑换。兑换完成后，牌堆中的第 5 张卡牌（最后一张）被补入桌面，此时牌堆为空。总计 7 回合达成目标。

题目 K. 股票交易

Alice 是一名量化交易员，她正在回测一套高频交易策略。已知某股票在 n 个时间点的价格序列为 $v_1, v_2, \ldots, v_n$。

Alice 设定的交易机器人由两个参数控制：买入阈值 a 和卖出阈值 b $(a < b)$，并严格遵循以下规则：

- 若某一时间点该股票的价格 $v_i \le a$，机器人就会强制买入 1 股；
- 若某一时间点该股票的价格 $v_i \ge b$，且当前机器人手中持有至少 1 股，机器人就会立刻卖出 1 股；
- 每个时间点最多买或卖 1 股；
- 在第 n 个时间点的交易结束后，如果机器人手中还有未卖出的股票，会触发强制平仓机制，将手中剩余的所有股票以 v_n 的价格全部卖出。

给定固定的买入阈值 a，现有 m 次独立询问，每次给出一个卖出阈值 b。对于每一次询问，请你帮 Alice 计算出该策略下的总收益（卖出总金额减去买入总金额）。

输入

第一行一个正整数 T $(1 \le T \le 10^4)$，表示数据组数。

对于每一组数据，第一行输入两个正整数 n, a $(1 \le n \le 2 \times 10^5,\ 1 \le a < 10^9)$，表示时间点个数以及买入阈值。

第二行输入 n 个整数 $v_1, v_2, \ldots, v_n$ $(1 \le v_i \le 10^9)$，按时间顺序给出该股票在各个时间点上的历史价格。

第三行输入一个整数 m $(1 \le m \le 2 \times 10^5)$，表示询问数量。

接下来输入一行 m 个整数 $b_1, b_2, \ldots, b_m$ $(a < b_i \le 10^9)$，表示每次询问的卖出阈值。

对于所有数据，保证 n 之和与 m 之和均不超过 2×10^5。

输出

对于每一组数据，输出一行 m 个整数，用空格隔开，表示每次询问对应的交易策略下的总收益。

样例

standard input	standard output
2 5 10 5 8 15 12 1 3 11 13 16 6 10 5 5 15 12 20 11 3 11 13 16	14 3 -11 17 25 21

注释

对于第一组数据的第一次查询 $(a = 10, b = 11)$，交易流程如下：

1. $v_1 = 5 < a$：买入 1 股，累计收益 -5，持仓 1；
2. $v_2 = 8 < a$：买入 1 股，累计收益 -13，持仓 2；

3. $v_3 = 15 > b$：卖出 1 股，累计收益 2，持仓 1；
4. $v_4 = 12 > b$：卖出 1 股，累计收益 14，持仓 0；
5. $v_5 = 1 < a$：买入 1 股，累计收益 13，持仓 1；
6. 平仓：剩余 1 股按 $v_5 = 1$ 卖出，最终收益 14。

题目 L. 二人游戏

洞烛和摩耶正在玩一个游戏。最初，她们有一个长度为 n 的正整数数组 $a = [a_1, a_2, \ldots, a_n]$。游戏由 **洞烛先手**，两人轮流进行操作。每一轮中，当前玩家必须执行以下步骤:

1. 在数组中选择一个下标 i（$1 \le i \le n$），要求 $a_i > 0$;
2. 对于所有满足 $j \ne i$ 的下标 j，将 a_j 更新为 $\min(a_j, a_i)$;
3. 将 a_i 的值减 1。

当轮到某位玩家行动时，如果数组中所有元素都为 0（此时无法选出正数 a_i），则该玩家输掉游戏，另一方获胜。假设两人都采取使自己获胜的最优策略，请判定最后谁将赢得游戏。

输入

第一行包含一个整数 T $(1 \le T \le 10^4)$，表示测试数据的组数。

对于每组测试数据，第一行包含一个整数 n $(1 \le n \le 5 \times 10^5)$，表示数组的长度。

第二行包含 n 个正整数 $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$，表示数组的初始状态。

保证所有测试数据的 n 之和不超过 2×10^6。

输出

对于每组测试数据，输出一行。如果摩耶获胜，输出 `Maya`；如果洞烛获胜，输出 `Insight`。

样例

standard input	standard output
3 3 2 1 3 4 2 1 3 1 4 2 1 1 4	Insight Insight Maya