问题 A. 制造一台计算机

你想制造一台计算机来实现特定的功能：给定一个整数 x，判断 x 是否位于区间 $[L, R]$ 内。为了实现这个目标，你设计了一个边权重为 0 和 1 的有向无环图 (DAG)，该图包含一个内程为 0 的起点节点和一个外程为 0 的终点节点。从起点节点开始，沿着一条路径到达终点节点，遍历的边权重序列就构成了范围 $[L, R]$ 内一个整数的二进制表示，不含前导零。范围 $[L, R]$ 内的每一个整数必须正好对应于该图中的一条唯一路径。这样，只要检查遍历该 DAG 是否能构造出整数的二进制表示，就能确定该整数是否位于 $[L, R]$ 范围内。

显然，您可以将每个整数的对应路径分成单独的链。但是，您意识到，对于一个大范围，这样的 DAG 需要太多节点，而您的计算机只有 256 MiB 内存，无法存储。因此，你需要压缩这个 DAG，允许不同的路径共享节点，以减少节点和边的数量。从形式上看，您需要构建一个节点数不超过 100 个的 DAG，其中每个节点的缺度不超过 200。该 DAG 的边权重必须为 0 和 1，其中一个起点节点的内度为 0，一个终点节点的外度为 0。在 $[L, R]$ 范围内的每个整数都必须与该 DAG 中从起点到终点的一条唯一路径相对应，并且任何路径都不能代表 $[L, R]$ 范围之外的任何整数。请注意，图中任何路径形成的二进制序列都不能有前导零。两个节点之间可能有两条权重不同的边。

输入

包含两个正整数 L、R 的单线（$1 \le L \le R \le 10^6$）。

输出

第一行应输出节点数 n（$1 \le n \le 100$）。

在接下来的 n 行中，第 i 行以整数 k 开头（$0 \le k \le 200$），表示从节点 i 输出的边的数量。然后输出 $2 \cdot k$ 个整数 $a_{i,k}$, $v_{i,k}$ （$1 \le a_{i,k} \le n, a_{i,k} \neq i, v_{i,k} \in \{0, 1\}$），这表示节点 i 有一条权重为 $v_{i,k}$ 的有向边通向节点 $a_{i,k}$。您必须确保输出表示的有向无环图满足要求。

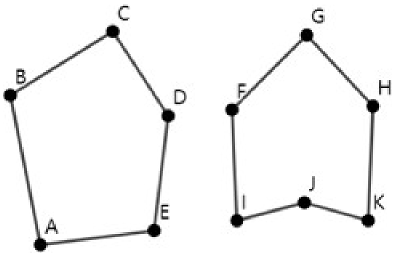
示例

标准输入	标准输出
5 7	8 3 2 1 3 1 4 1 1 5 0 1 6 1 1 7 1 1 8 1 1 8 0 1 8 1

问题 B.凹形船体

简单多边形是欧几里得平面上的一条封闭曲线，由两端相交的直线段组成。两条线段在每个端点相交，线段之间没有其他交点。

简单多边形可分为两类：凸多边形和凹多边形。凸多边形的定义是：对于多边形内部的任意两点，这两点之间线段上的所有点也位于多边形内部，或者位于多边形的边界上。不凸的简单多边形称为凹多边形。如下图所示，左边的是凸多边形，右边的是凹多边形。



现在，给定 n 个点，所有点都是不同的，且没有三个点是相邻的，你的任务就是从这 n 个点中选择一些（也许是全部），并以任意顺序将它们连接起来，以形成一个面积严格为正的**凹多边形**。你需要确定所能形成的凹多边形的最大面积。

输入

第一行包含一个整数 T（$1 \leq T \leq 10^4$），表示测试用例的数量。

对于每个测试用例，第一行包含一个正整数 n（$3 \leq n \leq 10^5$），表示点 的数量。

接下来的 n 行分别包含两个整数 x_i, y_i （$-10^9 \leq x_i, y_i \leq 10^9$），代表每个点的坐标。保证所有点都是不同的，没有三个点是相邻的。

所有测试案例的 n 之和不超过 $2 \cdot 10^5$。

输出

对于每个测试案例，如果不可能形成一个面积严格为正的凹多边形，则输出-1；否则，输出一个正整数，代表所能形成的凹多边形最大面积的两倍。可以证明答案总是正整数。

示例

标准输入	标准输出
2 6 -2 0 1 -2 5 2 0 4 1 2 3 1 4 0 0 1 0 0 1 1 1	40 -1

问题 C. 在哈尔滨指明方向

在一些地区，人们更习惯于用基本方向来指路，例如：向南走到第二个路口，然后向东走到第一个路口。但是，由于哈尔滨的路网规划比较复杂，很多街道并不完全符合基本方向。因此，如果您向在哈尔滨生活了很长时间的人提供绝对路径指示，他们可能很难理解您打算走的路线。

在哈尔滨，人们更习惯于用相对方向来指引方向。对于同一个地点，哈尔滨人可能会先指示你朝南走，然后说：沿路直走到第二个十字路口，然后左转，再直走到第一个十字路口。

为了解决这种差异，你决定编写一个程序，将使用红心方位的指路方式转换成哈尔滨居民喜欢的方式。当然，使用真实的哈尔滨地图会太复杂，所以在这个问题中，你可以假设地图是一个无限大的网格。

输入

第一行包含一个整数 T（$1 \le T \le 10^4$），表示测试用例的数量。

对于每个测试用例，第一行包含一个整数 n（$1 \le n \le 10$），表示方向指令的数量。

接下来的 n 行分别描述一条绝对位置指令，由一个字符 d（$d \in \{N, S, W, E\}$）和一个整数 x（$1 \le x \le 10$）组成，表示 "沿 d 方向前往第 x 个交叉点"。这里，N 代表北，S 代表南，W 代表西，E 代表东。

保证两个连续的指令不会有相同或相反的方向（南北方向相反，东西方向也相反）。

输出

对于每个测试用例，第一行输出一个整数 m（$1 \le m \le 20$）和一个字符 f（$f \in \{N, S, W, E\}$），代表哈尔滨式指令的数量和初始朝向，方向的含义与输入相同。

接下来，输出 m 行。每行以一个字符 $g \in \{Z, L, R\}$ 开始，其中 Z 表示直行，L 表示左转，R 表示右转。如果字符为 Z，该行还必须包含一个整数 y（$1 \le y \le 100$），表示直行到第 y 个交叉点。第一条输出指令必须以 Z 开头。连续指令不能有 相同的字符 g，L 和 R 指令不能相邻。

在这个问题中，你不需要最小化 m。如果有多种方法到达同一个目的地，那么任何有效的解法都是可以接受的。

示例

标准输入	标准输出
1 2 S 2	3 S Z 2 L

E 1	

问题 D. 一个简单的字符串问题

给您一个 2 行 n 列的字符网格，每个单元格包含一个小写字母。您可以从网格中的任意位置开始移动几步，每一步都可以向右或向下移动，并在任意单元格处停止。将依次访问的单元格中的字符连接起来，就形成了一个字符串。

当且仅当存在一个非空字符串 T，使得 $S = TT$ 时，字符串 S 称为双字符串。例如，aa 和 xyzxyz 是双字符串，而 a 和 xyzyz 不是。

给定字符网格，求最长双字符串的长度。

输入

第一行包含一个整数 n（$1 \le n \le 2 \times 10^5$），表示字符网格的列数。

接下来的两行包含两个长度为 n 的字符串，仅由小写英文字母组成，代表字符网格。

输出

输出一个整数，表示所能获得的最长双字符串的长度。

实例

标准输入	标准输出
5 abcab acabc	6
6 babbaa babaaa	6
2 宂	0

备注

在第一个例子中，最长的双字符串可以按如下方式得到（不唯一）：

abcab
acabc

大理石比赛

弹珠比赛是一种有趣的弹珠玩法，今天你想试试吗?

在 x 轴的负半轴上有 n 个起点，第 i 个起点位于 x_i 。共有 m 个弹珠，其中 m 为奇数，第 i 个弹珠的速度为 v_i 。在比赛中，每个弹珠以相等的概率随机选择一个起点，不同的弹珠可以选择相同的起点。然后，所有弹珠同时开始向 x 轴的正方向移动。设 c_i 为第 i 个弹珠选择的起点。在 t 时刻，第 i 个弹珠的坐标为 $x_{c_i} + v_i \cdot t$。

你是一名独特的弹珠比赛爱好者，并不关心哪颗弹珠最快。相反，你想知道所有 m 个弹珠坐标的**中位数**到达原点（即 $x = 0$）的确切时间。长度为 m（其中 m 为奇数）的序列的中位数定义为按升序排序（索引从 1 开始）时位于 $\frac{m+1}{2}$ 位置的元素。由于比赛尚未开始，起点也未确定，因此您感兴趣的是这一时间的预期值。为避免浮点错误，您只需输出结果 modulo $10^9 + 7$（详见输出格式）。

输入

第一行包含两个正整数 n 和 m（$1 \le n，m \le 500$，且 m 为奇数），分别代表起点数和弹珠数。

第二行包含 n 个整数 $x_1, x_2, \ldots, x_n$ （$-10^9 \le x_i < 0$），代表每个起点的坐标。保证所有 x_i 都是不同的。

第三行包含 m 个整数 $v_1, v_2, \ldots, v_m$ （$1 \le v_i \le 10^9$），表示每个弹珠的速度。

输出

输出一个整数，代表预期时间 modulo $10^9 + 7$。

形式上，设 $M = 10^9 + 7$。可以证明答案可以用不可约分数 $\frac{p}{q}$ 表示，其中 p 和 q 是整数，$q \not\equiv 0 \pmod M$)。输出等于 $p \cdot q^{-1} \pmod M$ 的整数，其中 q^{-1} 表示 q modulo M 的模乘逆。换句话说，输出这样一个整数 x：$0 \le x < M$ 且 $x \cdot q \equiv p \pmod M$。可以证明正好有一个 x 满足这个条件。

实例

标准输入	标准输出
2 3 -4 -5 1 2 3	250000004
3 3 -4 -5 -6 1 2 3	500000006

5 5 -4 -5 -6 -10 -2 1 2 3 2 4	4349866 72

备注

在第一个例子中，三个弹珠的速度分别为 1、2、3。考虑三个弹珠的初始位置：

- *-4*，-4，*-4*：在 $t = 2$ 时，三个弹珠的坐标分别为*-2*，0，2，中位数位于

出身

- -4、-4、-5：在 $t = 2$ 时，坐标为-2、0、1，中值位于原点。
- -4，-5，-4：在 $t = 2.5$ 时，坐标为-1.5，0，3.5，中位数位于原点。
- 对于 (-4，-5，-5)、(-5，-4，-4)、(-5，-4，-5)、(-5，-5，-4)、(-5，-5，-5)，中位数分别在 $t = 2.5$、$t = 2$、$t = 2.5$、$t = 2.5$ 时位于原点。

总之，预期时间为 $\frac{2+2+2.5+2.5+2+2+2.5+2.5}{8} = \frac{9}{4}$，因此需要输出 $9 \cdot 4^{-1} \bmod (10^9 + 7) = 250000004$。

问题 F. 1D 银河系

在一个神奇的一维空间中，有 n 颗行星，编号从 1 到 n。初始时（$t=0$），编号为 i 的行星位于位置 x_i，权重为 w_i（可以为负）。在现实世界中，行星是在万有引力的作用下运动的，同样，在这个一维星系中，行星也是在吸引力的作用下运动的。不过，这个星系中的运动并不遵循传统的物理定律。具体来说，对于这个一维星系中的任何行星来说，如果它左边行星的总重量大于它右边行星的总重量，那么它在下一个时间步就会向左移动一个单位。反之，如果右边行星的总重量大于左边行星的总重量，它就会向右移动一个单位。如果两边的重量相等，它就会保持原来的位置。可以假设行星不会发生物理碰撞，也就是说它们可以互相穿过。

形式上，让编号为 i 的行星在时间 t（$t=0$、1、2、......）的位置为 $x_{i,t}$。$w^l_{i,t}=\sum_{j:x_{j,t}<x_{i,t}} w_j$，其右侧行星的总重量为 $w^r_{i,t}=\sum_{j:x_{j,t}>x_{i,t}} w_j$。行星在下一个时间步的位置 $x_{i,t+1}$ 由以下公式得出：

$$x_{i,t+1}=\begin{cases} x_{i,t}-1, & w^l_{i,t}>w^r_{i,t} \\ x_{i,t}+1, & w^l_{i,t}<w^r_{i,t} \\ x_{i,t}, & w^l_{i,t}=w^r_{i,t} \end{cases}$$

有 q 个问题，每个问题都询问编号为 i 的行星在特定时间 t 的位置。

输入

第一行包含两个整数 n 和 q（$1 \le n, q \le 10^5$），分别代表行星数和查询次数。

接下来 n 行中的第 i 行包含两个整数 x_i, w_i（$-10^9 \le x_i, w_i \le 10^9$），代表编号为 i 的行星的初始位置和重量。

下面的 q 行分别包含两个整数 t 和 i（$0 \le t \le 10^9$，$1 \le i \le n$），代表一个查询。

输出

输出 q 行，每行代表相应查询的答案。

示例

标准输入	标准输出
4 12 0 1 1 2 -1 3 2 2 0 1 0 2 0 3 0 4 1 1 1 2 1 3 1 4 2 1 2 2 2 3 2 4	0 1 -1 2 1 0 0 1 0 1 1 0

问题 G.欢迎参加在线会议--"我很荣幸"。

您想在 MeLink 上组织一次在线会议，与会者人数为 n，从 1 到 n。

会议的组织流程如下：首先，一个人创建会议并加入。然后，已经加入会议的成员可以邀请一些尚未加入会议的熟人，直到 n 人全部到齐。但是，有 k 个参与者目前正忙于调试代码；这些人可以被邀请参加会议，但不能创建会议或邀请其他人。

您要确定是否有可能让所有 n 名与会者参加会议。如果可能，则确定邀请计划。

输入

第一行包含三个整数 n、m、k（$2 \le n \le 2 \times 10^5$，$1 \le m \le min\{5 \times 10$，$\frac{5n(n-1)}{2}\}$，$0 \le k \le n$），分别代表参与者人数、熟人关系人数和当前忙碌的参与者人数。

*第二行包含 k 个*整数 $a_1, \ldots, a$ *（$1 \le a \le n$）。*, a_k（$1 \le a_i \le n$），其中*第* i *个*整数表示参与者 a_i 忙碌。这些整数都是不同的。如果 $k = 0$，则这一行为空，但不能省略。

接下来的 m 行分别包含两个整数 p_i 和 q_i（$1 \le p_i, q_i \le n, p_i \neq q_i$），表示 p_i 和 q_i 彼此认识。熟人关系是相互的。保证同一熟人关系不会出现多次，而且每个参与者至少认识一个其他人。

输出

如果不可能组织 n 人参加的会议，则在第一行输出 "否"。

如果可以，则在第一行输出 "是"。然后，在第二行输出整数 t（$1 \le t \le n$），表示组织会议所需的步骤数。

在下面的 t 行中，每行描述组织会议的一个步骤。在*第* j 行，首先输出一个整数 x_j（$1 \le x_j \le n$）。如果 $j = 1$，x_j 代表创建会议的与会者；否则，x_j 必须是已经加入会议的与会者。所有 x_j 必须是不同的。接着，输出一个整数 y_j（$1 \le y_j \le n$），表示 x_j 在这一步中邀请的与会者人数。最后，输出 y_j 整数 z_l ($1 \le z_l \le n$)，代表被 x_j 邀请的参与者。所有 z_l 必须是不同的，在整个过程中，任何参与者都不能被邀请超过一次。

您不需要尽量减少 t，任何有效的计划都可以接受。

实例

标准输入	标准输出
4 5 2 3 4 1 2 1 3 2 3 3 4 2 4	是 2 1 2 2 3 2 1 4
4 5 3 2 4 3 1 2 1 3 2 3 3 4 2 4	没有

问题 H. 后续计算

给定长度为 m 的序列 $\{t\}$ 和长度为 L 的序列 $\{s\}$，其中 $\{s\}$ 由从左到右连续的 n 个片段组成。第 i 段包含 l_i 个相同元素，每个元素的值为 v_i。

序列 $\{s'\}$ 是根据一定规则对序列 $\{s\}$ 进行洗牌而形成的。具体来说
序列 $\{s'\}$ 满足 $s'_{i \cdot k \bmod L} = s_i$（索引从 0 开始）。这里，$k$ 是给定的正整数常数、并保证 gcd（k，L）= 1。

求 $\{t\}$ 作为子序列出现在 $\{s'\}$ 中的次数。形式上，如果有一个严格递增的索引序列 $0 \le i_1 < i_2 < \cdots < i_m < L$，使得对于每个 $j = 1, 2, \ldots, m$, $t_j = s'_{i_j}$，则 $\{t\}$ 被视为这些索引处 $\{s'\}$ 的子序列。您需要确定有多少个不同的索引组满足这一条件。由于答案可能很大，请输出结果，模数为 998244353。

输入

第一行包含四个整数 n、m、k、L（$1 \le n \le 2 \times 10^3$，$1 \le m \le 10$，$1 \le k < L \le 10^9$，gcd（$k$，$L$）= 1）。

第二行包含代表序列 $\{t\}$ 的 m 个整数（$1 \le t_i \le 10^3$）。
接下来的 n 行描述序列 $\{s\}$，每行包含两个整数 l_i, v_i（$1 \le l_i \le 10^9, 1 \le v_i \le 10^3$）。
可以保证 $\sum_{i=1}^{n} l_i = L$。

输出

输出一个整数，代表取模为 998244353 的结果。

实例

标准输入	标准输出
4 2 17 27 3 1 10 3 6 1 10 3 1 1	76
5 3 1789 15150 555 718 726 72 555 1029 718 5807 726 1002 718 7240 555	390415327

问题 I. 一个全新的几何问题

你是一个高维空间的魔术师，你有一个初始*的* n *维*超立方，它有边长 $a_1, a_2, \ldots, a_n$。对于 d *维*超微立方体，边长总和定义为 $\sum_{i=1}^{d} a_i$、其超体积为 $\prod_{i=1}^{d} a_i$。

您想获得一个边长总和为 S、超体积为 M 的超立方体。为此，您可以对当前超立方体执行降维和扩维操作。

- 减少尺寸：删除一个维度。
- 维度扩展：添加一个新维度，其边长为任意正整数。

这两种操作都非常耗时，所以你想确定获得边长总和为 S、超体积为 M 的超立方体所需的最少操作次数。

输入

第一行包含三个整数 n、S、M（$1 \le n \le 10^5$，$1 \le S, M \le 10^{10}$）。

第二行包含 n 个整数，代表超立方体$_i$ 的初始边长（$1 \le a_i \le 10^{10}$）。

输出

输出一个整数，代表所需的最小运算次数。如果不可能获得满足条件的超立方体，则输出-1。

实例

标准输入	标准输出
2 5 6 1 2	2
3 6 5 1 2 3	3
2 114514 735134400 114 514	20
2 4 7 1 3	-1

备注

对于第一个样本，一种可能的方法是：首先删除边长为 1 的维度，然后添加边长为 3 的维度。

Porblem J. 新能源汽车

一辆新能源汽车装有 n 个电池，其中第 i 个电池的容量为 a_i 单位。每个单位的电量可以让车辆行驶整整 1 公里。车辆只能前进，不能倒退。每行驶 1 公里，您可以选择使用哪块电池。

最初，所有电池都充满电。在行驶过程中，车辆将经过 m 个充电站。第 j 个充电站位于距离起点 x_j 公里处，只能为第 t_j -个电池充电。每个充电站提供的电量不受限制。

你们的任务是确定新能源汽车的最大行驶距离。

输入

第一行包含一个整数 T（$1 \le T \le 10^4$），表示测试用例的数量。

对于每个测试案例，第一行包含两个整数 n、m（$1 \le n，m \le 10^5$），分别代表电池数量和充电站数量。

第二行包含 n 个整数 $a_1, a_2, \ldots\ldots, a_n$（$1 \le a_i \le 10^9$），表示每块电池的容量。

接下来的 m 行分别包含两个整数 x_j, t_j（$1 \le x_j \le 10^9, 1 \le t_j \le n$），表示每个充电站的位置及其可充电的电池。

对于每个测试用例，保证 $1 \le x_1 < x_2 < \ldots < x_m \le 10^9$。所有测试用例的 n 或 m 之和均不超过 2 - 10 。^{5}

输出

对于每个测试用例，单行输出一个整数，代表车辆可行驶的最大距离。

示例

标准输入	标准输出
2 3 1 3 3 3 8 1 2 2 5 2 1 2 2 1	12 9

问题 K. 农场管理

你放弃了编程，来到三江平原开始务农。在田间劳作的日子里，你每天都有固定的作息时间，现在每天工作的时间**正好是** m 个单位。现在是收获季节，你需要收获和加工 n 种农作物。对于*第* i 种农作物，加工 1 个单位时间的利润为 w_i 。为了使你的日常工作不那么单调，对于每种*第* i 种农作物，每天加工的时间可以在 $[l_i, r_i]$ 之间（含整数）。

某一天，天气预报说明天会有一场大雨，你无法工作，所以你需要调整你的日程安排，在今天快速采集作物。具体来说，你最多可以选择一种农作物，并取消它的每日时间范围限制，允许处理这种农作物的时间是 $[0, m]$ 范围内的任意整数。所有其他作物的时间范围保持不变。您也必须**精确地**工作 m 个时间单位。

你要确定今天能赚取的最大利润。

输入

第一行包含两个整数 n 和 m（$1 \le n \le 10^5$，$1 \le m \le 10^{11}$），分别代表以时间为单位的作物种类数量和工作日长度。

接下来的 n 行分别包含三个整数 w_i、l_i 和 r_i（$1 \le w_i \le 10^6$，$1 \le l_i \le r_i \le 10^6$），表示作物的利润和时间限制。

保证 $\sum_{i=1}^{n} l_i \le m \le \sum_{i=1}^{n} r_i$。

输出

输出一个整数，代表你今天能赚取的最大利润。

示例

标准输入	标准输出
5 17 2 3 4 6 1 5 8 2 4 4 3 3 7 5 5	109

问题 L.树上的游戏

显然，树上任意两个节点之间都有一条唯一的简单路径。

小红和小兰在这棵树上玩游戏。在每次博弈中，双方都要从树上存在的所有$\frac{n(n-1)}{2}$简单路径（不考虑方向）中**独立、均匀地**随机选择一条简单路径。注意，他们可能会选择相同的路径。让 X 表示两条所选路径共有的边的数量，博弈得分为 X^2。

你的任务是求出小红和小兰玩一次游戏时得分的期望值 $E(X^2)$，并输出结果，模数为 998244353（详见输出格式）。

输入

第一行包含一个正整数 T（$1 \le T \le 10^4$），表示测试用例的数量。

对于每个测试用例，第一行包含一个正整数 n（$2 \le n \le 10^5$），表示树中的节点数。

接下来的 $n-1$ 行分别包含两个正整数 u、v（$1 \le u$、$v \le n$），表示节点 u 和 v 之间有一条边。

所有测试用例的 n 之和不超过 10^6。

输出

对于每个测试用例，输出一个整数，表示答案的模数为 998244353。

形式上，设 $M = 998244353$。可以证明答案可以用不可约分数$\frac{p}{q}$表示，其中 p 和 q 是整数，$q \not\equiv 0 \pmod M$。输出等于 $p \cdot q^{-1} \pmod M$ 的整数，其中 q^{-1} 表示 q modulo M 的模乘逆。换句话说，输出这样一个整数 x：$0 \le x < M$ 且 $x \cdot q \equiv p \pmod M$。可以证明正好有一个 x 满足这个条件。

示例

标准输入	标准输出
2 3 1 2 2 3 5 1 2 1 5 3 2 4 2	443664158 918384806

备注

对于示例中的第一个测试用例，不取模的答案是$\frac{10}{9}$。

在这 9 个可能的案例中

- 在 2 种情况下，两条路径之间的公共边数为 0；
- 在 6 种情况下，两条路径之间的公共边数为 1；
- 在 1 种情况下，两条路径之间的公共边数为 2。因此，答案

为 $E(X^2) = \frac{2\cdot 0^2+6\cdot 1^2+1\cdot 2^2}{9} = \frac{10}{9}$。

问题 M. 诡异的天花板

在学习上限函数时，一名学生写了以下伪代码：

```
函数 F(a, b)
    i ← b
    while i ≥ 2 do
        如果 a mod i = 0，那么
            返回 a/i
        end if
        i ← i - 1
    结束 while
    返回
结束函数
```

你知道这是不正确的，但你对函数 $f(a, b)$ 的特征很好奇的值。具体来说，你想计算 $\sum_{i=1}^{n} f(n, i)$ 的值。

输入

第一行包含一个整数 T（$1 \leq T \leq 10^3$），表示测试用例的数量。对于每个测试用例，有一行包含一个整数 n（$1 \leq n \leq 10^9$）。

输出

对于每个测试用例，输出一行包含一个代表答案的整数。

示例

标准输入	标准输出
3 5 451 114514	21 10251 7075858