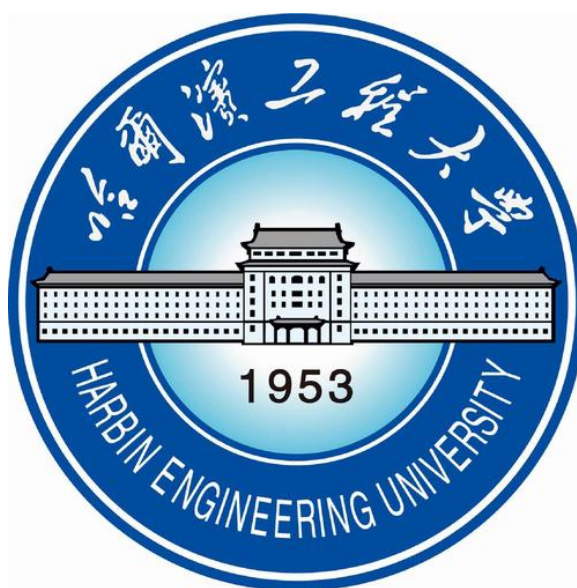


The 18th Harbin Engineering University Collegiate Programming Contest

Harbin Engineering University

April 16, 2023



Problem A. Keep Marching on Instead of Running off
Problem B. Pure Hacking Attempt (ULL version)
Problem C. If I Could Be a Constellation
Problem D. Tadokoro Flipping
Problem E. Mabinogion
Problem F. The Threshold of Masterpieces
Problem G. Among the Petals Dancing in the Wind
Problem H. 距离
Problem I. 海选
Problem J. Bit Operation
Problem K. 卡片游戏
Problem L. 投票
Problem M. 一元四次方程

请勿在比赛正式开始前打开题面！

Problem A. Keep Marching on Instead of Running off

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

Suletta 在营救受困的 Miorine!

为了营救 Miorine, Suletta 需要驾驶 Aerial 击败 n 台量产机。Aerial 和量产机均有生命值、攻击力和防御力三种属性。

其中 Aerial 的初始生命值为 10^{100} , 初始攻击力为 1, 初始防御力为 0。

而量产机的生命值均为 hp , 攻击力均为 atk , 防御力均为 0。

Suletta 需要操纵 Aerial 依次与 n 台量产机战斗。假设此时 Aerial 的攻击力为 a , 防御力为 d , 那么战斗后 Aerial 的生命值将减少 $\left\lfloor \frac{hp-1}{a} \right\rfloor \cdot \max(0, atk-d)$ 。

每次击败一台量产机后, Aerial 将获得性能加成。更具体地, Suletta 可以从以下两项加成中选择一项:

- 使 Aerial 的攻击力永久加 1;
- 使 Aerial 的防御力永久加 1。

为了尽可能的保全战斗力, Suletta 希望合理地选择每个加成, 使得 Aerial 受到的总伤害最小。现在, 你需要求出这个最小总伤害。

为避免 I/O 开销过大, 本题采用特殊方式给出数据。更具体地, 记 $f(n, hp, atk)$ 是量产机数量为 n 、生命值为 hp 、攻击力为 atk 时的最小总伤害, 输入数据将包含 t_1 个 n 的可选值、 t_2 个 hp 的可选值与 t_3 个 atk 的可选值, 你需要计算

$$\bigoplus_{i=1}^{t_1} \bigoplus_{j=1}^{t_2} \bigoplus_{k=1}^{t_3} f(n_i, hp_j, atk_k)$$

, 即对于从 $\{n_1, hp_1, atk_1\}$ 到 $\{n_{t_1}, hp_{t_2}, atk_{t_3}\}$ 的 $t_1 \cdot t_2 \cdot t_3$ 种输入组合分别计算得到的最小总伤害的异或和。

Input

第一行包含一个正整数 t_1 ($1 \leq t_1 \leq 200$), 代表量产机的数量的可选值的个数。

第二行包含 t_1 个正整数 n_1, n_2, \dots, n_{t_1} ($1 \leq n_i \leq 10^6$), 代表量产机的数量的可选值。

第三行包含一个正整数 t_2 ($1 \leq t_2 \leq 200$), 代表量产机的生命值的可选值的个数。

第四行包含 t_2 个正整数 $hp_1, hp_2, \dots, hp_{t_2}$ ($1 \leq hp_i \leq 10^6$), 代表量产机的生命值的可选值。

第五行包含一个正整数 t_3 ($1 \leq t_3 \leq 200$), 代表量产机的攻击力的可选值的个数。

第六行包含 t_3 个正整数 $atk_1, atk_2, \dots, atk_{t_3}$ ($1 \leq atk_i \leq 10^6$), 代表量产机的攻击力的可选值。

输入数据保证 $\min(atk_1, atk_2, \dots, atk_{t_3}) \geq \max(n_1, n_2, \dots, n_{t_1}) + \sqrt{\max(hp_1, hp_2, \dots, hp_{t_2})}$ 。

Output

输出一个正整数, 代表对于从 $\{n_1, hp_1, atk_1\}$ 到 $\{n_{t_1}, hp_{t_2}, atk_{t_3}\}$ 的 $t_1 \cdot t_2 \cdot t_3$ 种输入组合分别计算得到的最小总伤害的异或和。

Examples

standard input	standard output
3 1 2 3 1 4 2 5 6	20
4 11 4 5 14 4 19 19 8 10 2 893 364364	540678
3 114514 1919 810 1 810893 1 364364	686622249686

Note

对于第一个样例，共有 $\{1, 4, 5\}$, $\{2, 4, 5\}$, $\{3, 4, 5\}$, $\{1, 4, 6\}$, $\{2, 4, 6\}$, $\{3, 4, 6\}$ 六种输入组合，它们的答案分别为 15、20、24、18、24、29，故答案为 $15 \oplus 20 \oplus 24 \oplus 18 \oplus 24 \oplus 29 = 20$ 。

以输入组合 $\{3, 4, 5\}$ 为例：

- 第一回合时，量产机的生命值为 4，攻击力为 5，Aerial 的攻击力为 1，防御力为 0，因此战后 Aerial 受到 15 点伤害。此时可以选择使 Aerial 的攻击加 1。
- 第二回合时，量产机的生命值为 4，攻击力为 5，Aerial 的攻击力为 2，防御力为 0，因此战后 Aerial 受到 5 点伤害。此时可以选择使 Aerial 的防御加 1。
- 第三回合时，量产机的生命值为 4，攻击力为 5，Aerial 的攻击力为 2，防御力为 1，因此战后 Aerial 受到 4 点伤害。

此时 Aerial 受到的总伤害为 24。可以证明这是输入组合 $\{3, 4, 5\}$ 可以得到的最小总伤害。

Problem B. Pure Hacking Attempt (ULL version)

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

Papika 与 Cocona 正在研究算法！

这次她们在洛谷上遇到了一个名为 P3792 由乃与大母神原型和偶像崇拜 的题目。

这道题要求她们判断一个长为 n ($1 \leq n \leq 2 \cdot 10^5$) 的**正整数**数组 a ($1 \leq a_i \leq 10^9$) 在排序后是否为一个**连续**的数组（即后一项等于前一项加一的结果）。例如数组 $[6, 4, 3, 5]$ 满足要求，因为其排序后是 $[3, 4, 5, 6]$ ；而数组 $[1, 1, 4, 5, 1, 4]$ 则不满足要求。

Papika 很快就想到了一个维护 k 次方和的做法：

- 对于长度为 n 的正整数数组 a ，令其中的最小值为 mn ，最大值为 mx ；
- 同时令 $sum_1 = \sum a_i^k$ ， $sum_2 = \sum_{i=mn}^{mx} i^k$ ；
- 为了判断数组 a 是否满足要求，我们可以判断数组 a 是否同时满足：
 - $mx - mn + 1 = n$ ；
 - $sum_1 = sum_2$ 。
- sum_1 与 sum_2 可能很大，因此判断这两个数模 2^{64} （即用 C++ 的 `unsigned long long int` 储存并自然溢出）的结果。

然而 Cocona 很快就发现 Papika 的做法是错误的。现在 Cocona 希望构造一个长度为 n ($1 \leq n \leq 2 \cdot 10^5$) 的正整数数组 a ($1 \leq a_i \leq 10^9$)，使得 Papika 的做法对数组 a 返回**错误**的结果。

请注意 2^{64} 这个不寻常的模数。

Input

第一行包含一个正整数 k ($1 \leq k \leq 10^5$)，代表 Papika 的做法维护的 k 次方和。

Output

输出共两行。第一行包含一个正整数 n ($1 \leq n \leq 2 \cdot 10^5$)，代表你构造的正整数数组 a 的长度。

第二行 n 个正整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$)，代表你构造的正整数数组 a 。

你需要保证 Papika 的做法对于数组 a 返回**错误**的结果。

Examples

standard input	standard output
4	10 1 1 1 2 4 4 7 7 10 10
5	14 1 1 2 2 5 6 9 10 10 11 11 11 12 14
6	11 1 2 2 2 4 7 8 9 9 9 11

Note

对于第一个样例，当 $k = 4$ 时有 $mn = 1, mx = 10$ ，因此，

- $sum_1 = \sum a_i^4 = 25333$,
- $sum_2 = \sum_{i=1}^{10} i^4 = 25333$,

满足 Papika 的做法判断的两个条件，因此 Papika 的做法会认为输入数组是连续的，但实际上输入数组显然不是连续的。

Problem C. If I Could Be a Constellation

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

糟糕，Bocchi 的吉他突然断弦了！

Bocchi 的吉他可以视作一个有 n 个弦钮的平面图，其中，平面图指的是可以画在平面上并且使得不同的边只在顶点处相交的图。在吉他上，每个弦钮所处的位置都可以用平面直角坐标系上的坐标描述，其中，第 i 个弦钮的坐标为 (x_i, y_i) 。

弦钮间通过琴弦相连接。琴弦可以视为平面图中的**有向边**。我们定义该图中第 i 个弦钮可以调控第 j 个弦钮，当且仅当第 i 个弦钮可以通过若干条有向边连接到第 j 个弦钮，也即图中存在有向边序列 $(i, a), (a, b), \dots, (c, d), (d, j)$ ，其中 a, b, c, d 等中间记号表示经过的中继节点。

想要修复吉他，Bocchi 需要在图中添加有向边，使得整张图满足下列条件：

- 当且仅当 $x_i \geq x_j$ 且 $y_i \geq y_j$ 时，第 i 个弦钮可以调控第 j 个弦钮。

现在 Kita Ikuyo 正在帮助 Bocchi 救场，为了避免影响演出，Bocchi 希望以最快的速度修复吉他。因此 Bocchi 想知道修复吉他所需要的有向边的**最少数量**。

Input

输入共 $n + 1$ 行。

第一行包含一个正整数 n ($1 \leq n \leq 3 \cdot 10^5$)，表示弦钮的个数。

接下来 n 行中，第 i 行包含两个正整数 x_i, y_i ($1 \leq x_i, y_i \leq 3 \cdot 10^5$)，表示第 i 个弦钮的坐标。输入保证所有弦钮的坐标互不相同。

Output

输出一行一个正整数，表示需要添加的有向边的最少数量。

Examples

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

Note

对于第一个样例，一种可行方案如下：

1. 由 (2,1) 向 (1,1) 连有向边;
2. 由 (1,2) 向 (1,1) 连有向边;
3. 由 (2,2) 向 (1,2) 连有向边;
4. 由 (2,2) 向 (2,1) 连有向边。

这种方案一共添加了 4 条边。可以证明这是最优方案。

Problem D. Tadokoro Flipping

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

Tadokoro 和好友 Tono 正在玩抛硬币游戏！

众所周知，硬币是一种具有「正面」和「反面」的物体，其中抛出「正面」和抛出「反面」的概率均为 $\frac{1}{2}$ 。

现在 Tadokoro 将抛出 $n + 1$ 枚硬币，而 Tono 将抛出 n 枚硬币。如果 Tadokoro 抛出的「正面」数严格大于 Tono，那么 Tadokoro 获胜；否则 Tono 获胜。

现在 Tadokoro 想知道自己获胜的概率。请输出这个概率对 $10^9 + 7$ 取模后的结果。

记 $M = 10^9 + 7$ 。可以证明，本题所求的概率是一个有理数 $\frac{p}{q}$ ，其中 p, q 均为整数且 $q \not\equiv 0 \pmod{M}$ 。为了在模 M 意义下表示概率 $\frac{p}{q}$ ，你需要输出 $p \cdot q^{-1} \bmod M$ ，可以证明，这等于 $p \cdot q^{M-2} \bmod M$ 。

Input

第一行包含一个正整数 n ($1 \leq n \leq 114514$)，表示 Tadokoro 将抛出 $n + 1$ 枚硬币，而 Tono 将抛出 n 枚硬币。

Output

输出一行一个正整数，表示 Tadokoro 获胜的概率对 $10^9 + 7$ 取模后的结果。

Examples

standard input	standard output
1	500000004
2	500000004
3	500000004

Note

对于第一组样例， $n = 1$ 时存在如下情况：

1. Tadokoro 投出 反反，而 Tono 投出 正；
2. Tadokoro 投出 反反，而 Tono 投出 反；
3. Tadokoro 投出 正反，而 Tono 投出 正；
4. Tadokoro 投出 正反，而 Tono 投出 反；
5. Tadokoro 投出 反正，而 Tono 投出 正；
6. Tadokoro 投出 反正，而 Tono 投出 反；
7. Tadokoro 投出 正正，而 Tono 投出 正；
8. Tadokoro 投出 正正，而 Tono 投出 反。

其中 Tadokoro 在第 4, 6, 7, 8 四种情况中获胜，因此 Tadokoro 的胜率为 $\frac{1}{2}$ ，对 $10^9 + 7$ 取模后即 500000004。

Problem E. Mabinogion

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

MoriSummer 正为夺回 Sanja 与 FakeMoriSummer 展开决斗！

在决斗中，MoriSummer 需要吟唱咒文。一段咒文可以视为一个长度为 n 的，只包含小写英文字母的字符串 S ，而咒文蕴含的「魔力值」可以视为字符串 S 中只出现一次的子序列的个数。

我们定义一个字符串 S 的子序列为删除 S 中的一些元素，且不改变剩余元素的相对顺序所得到的，长度至少为 1 的字符串。

例如 `hhaaa` 是 `henhenhenaaaa` 的子序列，但 `aaaa` 不是 `henhenhenaaaa` 的子序列。

现在 MoriSummer 想知道咒文蕴含的「魔力值」，即字符串 S 中只出现一次的子序列的个数。这个数可能很大，因此请输出其对 998 244 353 取模的结果。

Input

本题有多组数据。第一行包含一个正整数 T ($1 \leq T \leq 2 \cdot 10^4$)，表示数据组数。

对于每组数据，第一行包含一个正整数 n ($1 \leq n \leq 10^5$)，表示字符串 S 的长度为 n 。

第二行包含一个字符串 S ，代表题目所询问的，只包含小写英文字母的字符串。

输入数据保证 $\sum n \leq 10^5$ 。

Output

对于每组询问，输出一个正整数，代表字符串 S 中只出现一次的子序列的个数，对 998 244 353 取模的结果。

Example

standard input	standard output
4	2045
11	3
abcdefghija	1
4	549
aabb	
5	
aaaaa	
14	
henhenhenaaaaa	

Note

对于第二个样例，在字符串 `aabb` 中：

- 子序列 `a` 共出现 2 次；
- 子序列 `b` 共出现 2 次；
- 子序列 `aa` 共出现 1 次；
- 子序列 `ab` 共出现 4 次；
- 子序列 `bb` 共出现 1 次；

- 子序列 `aab` 共出现 2 次;
- 子序列 `abb` 共出现 2 次;
- 子序列 `aabb` 共出现 1 次。

共有 `aa`、`bb`、`aabb` 三个子序列在字符串 `aabb` 中只出现一次，因此第二个样例的答案为 3。

Problem F. The Threshold of Masterpieces

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

销量预测是名作之壁吧的传统活动，在这个活动中，选手将预测给定的 n 部作品的销量。

预测结果由**预测偏差** c 决定。假设其中第 i 个选手对第 j 部作品的预测销量为 $a_{i,j}$ ，而这部作品的实际销量为 b_j ，那么第 i 个选手的**预测偏差**为 $c_i = \sum_{j=1}^n |a_{i,j} - b_j|$ 。

如果 c_i 是所有预测偏差中最小的，则 i 号选手将成为「销量明灯」。如果有多人的预测偏差同时达到最小值，那么他们将同时成为「销量明灯」。

这次的预测活动共有 Tadokoro、Miura 和 Kimura 三人参加。其中的 Tadokoro 非常希望成为「销量明灯」。为了成为「销量明灯」，Tadokoro 可以任意次进行如下操作：

- 花费 1 代价，选择 n 部作品中的一部，使其销量增加 1。

现在 Tadokoro 想知道使自己成为「销量明灯」所需的最小代价。

Input

第一行包含一个正整数 n ($1 \leq n \leq 12$)，代表作品的数量。

第二行包含 n 个整数 b_1, \dots, b_n ($0 \leq b_i \leq 2 \cdot 10^5$)，代表作品的实际销量。

第三行包含 n 个整数 $a_{1,1}, a_{1,2}, \dots, a_{1,n}$ ($0 \leq a_{1,i} \leq 2 \cdot 10^5$)，代表 Tadokoro 的预测销量。

第四行包含 n 个整数 $a_{2,1}, a_{2,2}, \dots, a_{2,n}$ ($0 \leq a_{2,i} \leq 2 \cdot 10^5$)，代表 Miura 的预测销量。

第五行包含 n 个整数 $a_{3,1}, a_{3,2}, \dots, a_{3,n}$ ($0 \leq a_{3,i} \leq 2 \cdot 10^5$)，代表 Kimura 的预测销量。

Output

输出一个非负整数，代表使 Tadokoro 成为「销量明灯」所需的最小代价。

如果无论如何操作都无法使 Tadokoro 成为「销量明灯」，则输出 -1。

Examples

standard input	standard output
4 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0	2
4 1 1 1 1 0 0 0 0 1 0 1 0 0 1 0 1	-1
3 16793 23383 1735 1919 810 114514 2574 1872 14060 875 1210 18450	64425
5 14062 9915 5805 5256 7331 2410 2263 1764 10090 18640 4212 3765 4030 5021 7652 4062 4915 3805 5555 88888	10900

Note

对于第一个样例，Tadokoro 可以购买第一部作品和第二部作品各一份。

对于第二个样例，可以证明 Tadokoro 不可能成为「销量明灯」。

Problem G. Among the Petals Dancing in the Wind

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

Nigomiku 有一个 $n \times n$ 的下三角表格。对于 $1 \leq j \leq i \leq n$, Nigomiku 在表格的第 i 行第 j 列填入 $\frac{j}{i}$, 结果如下:

$$\begin{array}{ccccccc} \frac{1}{1} & & & & & & \\ \frac{1}{2} & \frac{2}{2} & & & & & \\ \frac{1}{3} & \frac{2}{3} & \frac{3}{3} & & & & \\ \frac{1}{4} & \frac{2}{4} & \frac{3}{4} & \frac{4}{4} & & & \\ \frac{1}{5} & \frac{2}{5} & \frac{3}{5} & \frac{4}{5} & \frac{5}{5} & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \end{array}$$

。这真是太无聊了——Nigomiku 这样想着, 决定把表格中的每个分数 $\frac{p}{q}$ 都约分到最简形式。回忆小学数学知识, 你意识到这意味着令 p, q 同除 $\gcd(p, q)$ 使得分子分母互质。于是, 最终的表格如下:

$$\begin{array}{ccccccc} \frac{1}{1} & & & & & & \\ \frac{1}{2} & \frac{1}{1} & & & & & \\ \frac{1}{3} & \frac{2}{3} & \frac{1}{1} & & & & \\ \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & \frac{1}{1} & & & \\ \frac{1}{5} & \frac{2}{5} & \frac{3}{5} & \frac{4}{5} & \frac{1}{1} & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \end{array}$$

。现在, 对于给定的表格高度 n , Nanarikom 想要计算最终的表格中所有分子的和。由于答案可能很大, 你只需要帮 Nanarikom 计算答案对 998 244 353 取模后的结果。

Input

第一行包含一个正整数 n ($1 \leq n \leq 10^9$), 代表表格的高度。

Output

输出一个整数, 代表最终的表格中所有分子的和对 998 244 353 取模后的结果。

Examples

standard input	standard output
3	7
5	24
2539	996900927

Note

对于第一组样例, 容易看出 $(1) + (1 + 1) + (1 + 2 + 1) = 7$ 。

Problem H. 距离

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

在一条长为 l 的线段上独立地、均匀随机地取两个点，求这两个点之间距离的期望。

你的答案将被认为是正确的，当且仅当你的答案和标准答案的绝对误差或相对误差不超过 10^{-6} 。形式化地，如果你的答案是 a ，标准答案是 b ，则你的答案被认为是正确的当且仅当 $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$ 。

Input

第一行包含一个整数 l ($1 \leq l \leq 100$)，表示线段的长度。

Output

输出一个实数，表示两个点之间距离的期望。

Example

standard input	standard output
1	0.3333333333

Note

对于样例，根据简单概率论知识可以得出答案为 $\frac{1}{3} \approx 0.3333333333$ 。

Problem I. 海选

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

有 $10^{10^{10}}$ 个选手参加海选，他们分别被编号为 $0, 1, 2, \dots, 10^{10^{10}} - 1$ 。

海选的评委发现选手实在是太多了，所以干脆直接摆烂了，他打算象征性地进行 n 轮选拔。具体地，第 i 轮开始时，他会将所有人按照编号从小到大排成一队，然后进行两种操作中的一种：

1. 淘汰站在奇数位置的人；
2. 淘汰站在偶数位置的人。

例如，如果第一轮选拔采用第一种操作，则会剩下编号为 $1, 3, 5, 7, \dots$ 的选手；如果第一轮选拔采用第二种操作，则会剩下编号为 $0, 2, 4, 6, \dots$ 的选手。

n 轮选拔结束后，剩下的人中编号最小的选手将成为冠军。你需要计算冠军的编号。由于答案可能很大，你只需要输出冠军的编号对 $998\,244\,353$ 取模的结果。

Input

第一行包含一个正整数 n ($1 \leq n \leq 10^5$)，代表选拔轮数。

第二行有 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 2$)，代表第 i 轮选拔采用第 a_i 种操作。

Output

输出一个整数，表示冠军的编号对 $998\,244\,353$ 取模的结果。

Examples

standard input
3 1 1 2
standard output
3
standard input
30 1 1 2 2 1 2 1 2 1 1 1 1 1 2 1 1 2 2 2 2 2 2 1 1 1 2 1 1 1
standard output
114514

Note

对于第一个样例：

- 第 1 轮选拔中采用了第一种操作，因此操作后剩下的选手编号分别为 $1, 3, 5, 7, \dots$ ；
- 第 2 轮选拔中采用了第一种操作，因此操作后剩下的选手编号分别为 $3, 7, 11, 15, \dots$ ；
- 第 3 轮选拔中采用了第二种操作，因此操作后剩下的选手编号分别为 $3, 11, \dots$ ；

由此可见，冠军的编号为 3 。

Problem J. Bit Operation

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

给定两个正整数 a 和 b ，你可以对 a 进行任意次操作，使得最终 $a = b$ 。在每次操作中，你可以选择 a 的二进制表示中的一个为 1 的数位并将其变为 0。

现在，你需要计算有多少种不同的操作序列可以将 a 变成 b 。

两种操作序列被视为不同的当且仅当两种操作序列的操作次数不同，或两种操作序列存在某次操作操作了不同的数位。

Input

第一行包含两个正整数 a, b ($1 \leq a, b \leq 2^{10} - 1$)。

Output

输出一个整数，表示不同操作序列的数量。

Examples

standard input	standard output
5 4	1
5 5	1
7 9	0

Note

对于第一个样例，只有一种方案，即将 $5 = 101_{(2)}$ 的最低位的 1 变为 0。

对于第二个样例，因为 $5 = 5$ ，所以只有一种方案，即什么操作也不做。

对于第三个样例，7 不可能变成 9，所以不存在任何方案。

Problem K. 卡片游戏

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

小明正在打算使用数字卡片将数进行 k 进制表示。例如，114514 的十进制表示（即 $114514_{(10)}$ ）需要三个 1，两个 4，一个 5；而 9 的二进制表示（即 $1001_{(2)}$ ）则需要两个 0 和两个 1。

给定两个正整数 n, k ，你需要准备足够多的数字卡片，使得小明可以使用你的数字卡片将 $[1, n]$ 中任意一个数进行 k 进制表示。

现在，你想知道，为满足以上条件，你最少需要准备多少卡片。

Input

第一行包含两个整数 n, k ($1 \leq n \leq 10^8, 2 \leq k \leq 10$)。

Output

输出一个整数，表示最少需要的卡片数量。

Examples

standard input	standard output
11 10	11
11 4	6
11451 4	26

Note

对于第一个样例，在十进制下表示 1 到 11 显然需要 2 张 1 和 0, 2, 3, 4, 5, 6, 7, 8, 9 各 1 张，因此一共需要 11 张卡片。

对于第二个样例，在四进制下表示 1 到 11 需要 1, 2 各 2 张和 0, 3 各 1 张，因此一共需要 6 张卡片。

Problem L. 投票

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

DoubleHappy 所在的班级需要选出两名优秀成员，为此，他们需要进行一次投票。

DoubleHappy 所在的班级有 n 位同学，从 1 到 n 编号，其中 DoubleHappy 是 1 号。在投票过程中，每位同学都有一张选票，可以投给任意一位同学（包括自己），最终得票最多的两人当选优秀成员。当票数相同时，相同票数的人进行剪刀石头布，赢的人排名较前。

DoubleHappy 想要一定当选，即他希望他的票数至少是第二多的且严格多于第三。为此，他已经打探好了班级里每个人的投票意向，并且知道让每个同学改变主意分别需要多少原石。具体地， i 号同学在初始状态下打算投票给 a_i 号同学，而收受 b_i 个原石的贿赂之后可以改变主意投票给 DoubleHappy。

现在，DoubleHappy 想知道他最少花费多少原石才能贿赂足够多的同学，以保证自己当选优秀成员。

Input

第一行包含一个整数 n ($2 \leq n \leq 5000$)，表示班级中的人数。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$)，表示 i 号同学打算投票给 a_i 号同学。DoubleHappy 当然会投自己，所以输入数据保证 $a_1 = 1$ 。

第三行包含 n 个整数 b_1, b_2, \dots, b_n ($0 \leq b_i \leq 5000$)，表示 i 号同学收受 b_i 个原石的贿赂之后可以改变主意投票给 DoubleHappy。

Output

输出一个整数，表示 DoubleHappy 想要必定当选优秀成员至少需要多少原石。

Examples

standard input	standard output
4 1 2 3 4 7 6 7 8	6
5 1 5 5 5 5 5 6 7 8 5	0

Note

对于第一个样例，DoubleHappy 可以贿赂 2 号同学，这样 DoubleHappy 有两票，其他同学最多只有一票。可以证明没有更优解。

对于第二个样例，只有 DoubleHappy 和 5 号同学得票，所以 DoubleHappy 不进行任何操作也可以进入前二并当选。

Problem M. 一元四次方程

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

众所周知, Asoul 是一个四个人组成的虚拟偶像团体, 其成员分别是向晚大魔王、贝拉Kira、嘉然今天吃什么和乃琳Queen。因为是虚拟偶像, 所以她们每个人都代表一个虚数。可以知道必定存在一个一元四次方程, 其四个根恰好分别是她们所代表的数。作为“4u”, jxt 正在不断寻找这个方程。

现在, 你的任务是, 帮助 jxt 找到五个整数系数 a, b, c, d, e ($-10 \leq a, b, c, d, e \leq 10, a \neq 0$), 使得一元四次方程 $ax^4 + bx^3 + cx^2 + dx + e = 0$ 没有实数解。

请输出任意一组满足条件的系数。

Input

本题无输入。

Output

输出一行五个整数, 分别代表 a, b, c, d, e ($-10 \leq a, b, c, d, e \leq 10$ 且 $a \neq 0$), 每两个整数之间用一个空格隔开。你的输出需要保证一元四次方程 $ax^4 + bx^3 + cx^2 + dx + e = 0$ 没有实数解。

请不要输出多余的空格; 请不要输出任何回车。

Note

对于实数系四次方程 $ax^4 + 4bx^3 + 6cx^2 + 4dx + e = 0$, 我们记

- $G = a^2d - 3abc + 2b^3, H = b^2 - ac, I = ae - 4bd + 3c^2$
- $J = \frac{4H^3 - a^2HI - G^2}{a^3}$
- $\Delta = I^3 - 27J^2$
- $\delta = 12H^2 - a^2I$

, 则我们可以按顺序使用以下根的判别法则:

1. 若 $G \neq 0, I = J = 0$, 则方程有一个三重实根和一个单重实根;
2. 若 $G = \delta = 0, H > 0$, 则方程有两个互异的两重实根;
3. 若 $G = \delta = 0, H < 0$, 则方程有一对两重共轭虚根;
4. 若 $H = I = G = 0$, 则方程有一个四重实根;
5. 若 $\Delta < 0$, 则方程有两个互异实根和一对共轭虚根;
6. 若 $\Delta > 0, \delta > 0, H > 0$, 则方程有四个互异实根;
7. 若 $\Delta > 0, H \leq 0, \delta < 0$, 则方程有两对互异的共轭虚根;
8. 若 $\Delta = 0, IJ \neq 0, \delta > 0, H > 0$, 则方程有一个两重实根和两个单重实根;
9. 若 $\Delta = 0, IJ \neq 0$, 且 $H \leq 0$ 与 $\delta < 0$ 之中至少有一者成立, 则方程有一个两重实根和一对共轭虚根。

请注意题目描述和参考资料中方程的系数的区别!

为什么题目不是一元五次方程捏? 因为一元五次方程没有一般根式解捏。