2026 年牛客寒假算法基础集训营 1

主办方：牛客竞赛

命题人： yeVegeTable（**苯环**）

比赛日期：2026 年 2 月 3 日 14:00 ～ 18:00

寒假基础集训营

试题汇总

题号	题目名称
A	A+B Problem
B	Card Game
C	Array Covering
D	Sequence Coloring
E	Block Game
F	Permutation Counting
G	Digital Folding
H	Blackboard
I	AND vs MEX
J	MST Problem
K	Constructive
L	Need Zero

若存在疑问请通过比赛页面答疑区提问，存在题面变更以官方通知为准

Problem A. A+B Problem

Time limit: 2 seconds
Memory limit: 256 megabytes

小苯正在学习 A + B Problem，为此他从家中翻出了恰好八个”七段码数位显示器”（以下简称显示器）。

如下图所示，显示器共有 7 个灯管，图中已标明编号。点亮其中的一些灯管就可以形成合法的数字，0 – 9 的对应的点亮结果如下图二，其中红色灯管是被点亮的，灰色则是未被点亮**（其余的结果均是不合法的数字）**。

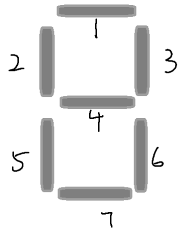

图 1: 灯管编号

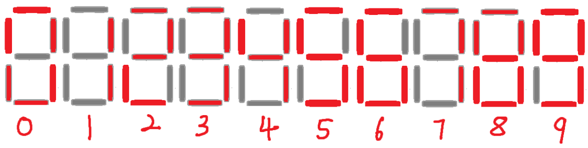

图 2: 0 ∼ 9 对应的状态

遗憾的是，放置时间太久导致所有的显示器都发生了**相同**的故障，具体来说，在点亮他们的编号为 i 的灯管时，灯管都是仅有 $\mathrm{p_i}\%$ 的概率会被点亮，而还会有 $1 - \mathrm{p_i}\%$ 的概率不会被点亮。（各根灯管的点亮尝试相互独立；不同显示器之间、同一显示器内不同编号灯管之间的点亮结果均互不影响。）

但小苯的学习还得进行下去，现在他会让小红指定一个整数 C $(0 \leqq \mathrm{C} \leqq 2026)$，接着小苯会将其中的四个排成一排，另外四个排成另一排，并对其 7 根灯管（共 $7 \times 8 = 56$ 根）**均各**尝试一次点亮操作。（由于所有显示器参数相同，具体选哪 4 台放在第一排与第二排均等价，可视为任意固定分配。）

现在请你计算出如下事件的概率（需全部满足）：

- 最终所有显示器均有灯管被点亮（也就是说显示器的灯管不能全灭）。
- 最终所有显示器显示的结果均为合法数字。
- 第一排的显示器前后拼接形成的十进制数记作 A，第二排的显示器前后拼接形成的十进制数记作 B 的话，满足：$\mathrm{A} + \mathrm{B} = \mathrm{C}$。

（需要注意的是：我们认为 A 和 B 都可以存在前导 0。）

Input

每个测试文件均包含多组测试数据。第一行输入一个整数 T $(1 \leqq T \leqq 1000)$ 代表数据组数，每组测试数据描述如下：

第一行一个整数 C $(0 \leqq \mathrm{C} \leqq 2026)$ 表示小红指定的整数。

第二行输入七个整数 $\mathrm{p_1}, \mathrm{p_2}, \ldots, \mathrm{p_7}$ $(0 \leqq \mathrm{p_i} \leqq 100)$，分别表示显示器中，每一根灯管在尝试点亮时，被点亮的概率百分比。（概率为 $\mathrm{p_i}\%$。）

Output

对于每组测试数据：

在单独的一行输出一个整数表示最终的概率（对 998244353 取模）。

可以证明答案可以表示为一个不可约分数 $\frac{p}{q}$，为了避免精度问题，请直接输出整数 $(p \times q^{-1} \bmod M)$ 作为答案，其中 $M = (998244353)$，q^{-1} 是满足 $q \times q^{-1} \equiv 1 \pmod{M}$ 的整数。

Examples

standard input	standard output
3	1
0	994344961
100 100 100 0 100 100 100	20591129
0	
100 100 100 0 50 100 100	
34	
24 54 10 12 33 1 99 98	

Explanation

对于第一组测试数据，所有的显示器都恰好只能显示出 0，因此显示出 A 的四个显示器显示 0 的概率为 1，显示出 B 的也为 1，因此 A + B = 0 一定成立，即概率为 1。

对于第二组测试数据，所有的显示器都恰好有 50% 的概率显示出 0，而另 50% 的概率显示非数字的不合法结果，因此四个显示 A 的显示器显示出 0 的概率为：$(\frac{1}{2})^4$，显示 B 的也同理。因此 A + B = 0 的概率为 $(\frac{1}{2}^4)^2 = \frac{1}{256}$，对 998244353 取模的值为 994344961。

Note

【分数取模教程】

以下文中的 mod 表示取模，也就是大家平时写的 % 符号。

我们以任意分数：$\frac{a}{b}$ 举例。

直接给出结论：根据费马小定理，在模数 m 为质数，且 b 不是 m 的倍数的情况下有：

$$\frac{a}{b} \bmod m = a \times (b^{m-2}) \bmod m$$

也就是说在 mod m 的意义下，$\frac{1}{b} = b^{m-2} \bmod m$。

例如：在 mod 998244353 的情况下，$\frac{2}{3} = 2 \times 3^{998244351} (\mathrm{mod} 998244353)$，这是因为 998244353 是一个质数，且 3 不是 998244353 的倍数。

上式中，$b^{m-2} \bmod m$ 实际上就是我们常说的" 乘法逆元"，即把" a 除以 b" 转化为" a 乘上 b 的逆元"，这样一来结果算下来是正确的，同时我们将分数（也就是小数）域下的运算转为了整数域下的运算，这样一来就避免了小数可能产生的精度问题，同时不影响答案的正确性。

证明需要大量计算，这里不再展开，我们只需要使用结论即可。

Problem B. Card Game

Time limit: 2 seconds
Memory limit: 256 megabytes

小苯正在和小红玩卡牌游戏。游戏中有 $2 \times n$ 张牌，每张牌上都有一个数字，所有牌的数字恰好构成了一个长度为 $2 \times n$ 的排列。

游戏最初时，$2 \times n$ 张卡牌被恰好平均分成了两组 n 张牌，并分别发给了两人，小苯第 i 张牌上的数字是 a_i，而小红第 i 张牌上的数字是 b_i，具体的游戏过程如下：

- 如果两人之中有一人已经没有牌了，则游戏结束。
- 两人取出自己的牌堆里的第一张牌（编号为 1）并比较大小，对应数字大的那一方得一分，同时弃掉这张更大的牌；而数字小的一方则没有任何变化。（即既不弃牌，也不得分。）

而现在小苯希望自己的得分尽可能多，为此他在游戏开始前可以**任意**地重新排列自己的牌，以得到更高的游戏分数。

现在你的任务则是求出，有多少种重新排列（选择不进行重排也是一种方案）的方式，能使得小苯得到他能得到的最高分（对 998244353 取模）。

Input

每个测试文件均包含多组测试数据。第一行输入一个整数 T $(1 \leqq T \leqq 10^5)$ 代表数据组数，每组测试数据描述如下：

第一行一个正整数 n $(1 \leqq n \leqq 2 \times 10^5)$，表示两人的卡牌数量。

第二行 n 个正整数 a_i $(1 \leqq a_i \leqq 2 \times n)$，表示小苯的卡牌上的数字。

第三行 n 个正整数 b_i $(1 \leqq b_i \leqq 2 \times n)$，表示小红的卡牌上的数字。

（保证 a 数组和 b 数组共同构成一个长度为 $2 \times n$ 的排列。）

除此之外，保证单个测试文件的 n 之和不超过 2×10^5。

Output

对于每组测试数据：

输出小苯有多少种重新排列自己卡牌的方案，以得到最高的得分。（由于结果可能很大，因此输出结果对 998244353 取模的值。）

Examples

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

Explanation

对于第一组测试数据，无论小苯如何重排他的卡牌，他的得分总是 0，而有：$\{1, 2\}$ 和 $\{2, 1\}$ 两种重排方式，因此输出 2。

Problem C. Array Covering

Time limit:	1 second
Memory limit:	256 megabytes

给定长度为 n 的数组，其中第 i 个数的值为 a_i。

小苯希望数组中所有数字的总和尽可能大，为此他可以做**任意次**如下操作：

- 选择一对下标 l, r $(1 \leqq l < r \leqq n)$，接着将 (l, r) 区间（注意是开区间）内的所有数都变为区间端点值的较大者。
- 形式化的即：对所有 j $(l < j < r)$，均执行：$a_j := \max(a_l, a_r)$。（其中 := 表示赋值操作）。

你的任务就是求出数组总和的最大值。

Input

每个测试文件均包含多组测试数据。第一行输入一个整数 T $(1 \leqq T \leqq 10^5)$ 代表数据组数，每组测试数据描述如下：

第一行一个正整数 n $(1 \leqq n \leqq 5 \times 10^5)$ 表示数组 a 的长度。

第二行 n 个正整数 a_i $(1 \leqq a_i \leqq 10^9)$ 表示最初时所有数字的总和。

除此之外，保证同一个测试文件的所有测试数据中 n 的总和不超过 5×10^5。

Output

对于每组测试数据：

输出一行一个正整数表示在可以进行任意次操作的情况下，所有数字之和的最大值。

Examples

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

Explanation

对于第一组测试数据，可以选择 $l = 1, r = 3$ 操作一次，操作完后数组为：$\{5, 5, 5\}$，求和为 15 最大。

Problem D. Sequence Coloring

Time limit: 2 seconds
Memory limit: 256 megabytes

给定一个长度为 n 的数字序列 $a_1, a_2, a_3, \cdots a_n$，每个元素有一个颜色，且初始时要么为白色，要么为黑色，使用其值的大小表示其颜色：若 $a_i > 0$，则第 i 个元素为白色；否则为黑色。

小苯可以在第 0 秒时将其中至多 k 个**白色的**数字染红，接下来从第 1 秒开始，每秒都会发生如下事件：

- 所有**红色**的数字会将其右侧 a_i 个数字里的（最多到 n）**白色**数字染红。
- 形式化的：对于所有红色元素所在的位置 i，将第 $i+1$ 到 $\min(n, i+a_i)$ 个元素中的**白色**元素也染红。
- 所有的红色元素同步进行这一操作（即每一秒开始时已经是红色的元素，会在该秒内尝试染红其它元素）；
- **黑色数字并不会、也无需被染红。**

请你帮助小苯以最优策略染色最初的数字，使得所有的白色元素都被染红的时间尽可能短（可以为 0 秒）。求出这个最短时间，或报告无论如何都无法将所有的白色元素都染红。

Input

每个测试文件均包含多组测试数据。第一行输入一个整数 T $(1 \leqq T \leqq 10^5)$ 代表数据组数，每组测试数据描述如下：

第一行两个整数 n, k $(1 \leqq k \leqq n \leqq 5 \times 10^5)$，分别表示数字序列的长度，以及最开始可以染红的数字个数。

第二行 n 个整数 a_i $(0 \leqq a_i \leqq n)$ 表示数字序列。（其中 $a_i > 0$ 为白色数字，$a_i = 0$ 为黑色数字。）

除此之外，保证单个测试文件的 n 之和不超过 5×10^5。

Output

对于每组测试数据：

如果可以将所有的白色数字都染红，则输出一个整数表示最短的全染红时间；否则输出一个 -1 即可。

Examples

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

Explanation

对于第一组测试数据，我们一开始选择染红：$i = 1$ 和 $i = 6$ 即可，即：$\{2, 0, 1, 1, 0, 1\}$。

第一秒后，序列变为：$\{2, 0, 1, 1, 0, 1\}$；

第二秒后，序列变为：$\{2, 0, 1, 1, 0, 1\}$；

此时，序列中所有的白色数字均已被染红，耗时 2 秒，可以证明不存在更优的方案，因此输出 2。

Problem E. Block Game

Time limit: 1 second
Memory limit: 256 megabytes

小苯正在玩方块小游戏，游戏中有一排 n 个方块，每个方块上都有一个数字，此外，他还有一个写着数字 k 的万能方块，游戏过程如下：

小苯可以进行任意次以下操作：

- 将万能方块从方块序列的最左侧插入，同时最右侧的方块 (a_n) 会被挤出这一行，成为新的万能方块。
- 形式化地，记开始时的序列为 $a'_1, a'_2, \ldots, a'_n$（初始时 $a'_i = a_i$），万能方块值为 k'（初始时 $k' = k$），操作完成后序列变为 $k', a'_1, \ldots, a'_{n-1}$，万能方块上的数字变为 a'_n，其他方块保持相对顺序整体右移一位。

你的任务是计算出，按照最优方式经过若干次操作后，从左往右数**第一个**方块上的数字加上最终的**万能方块**上的数字总和的最大值。

Input

每个测试文件包含多组测试数据。第一行输入一个整数 T $(1 \leqq T \leqq 10^5)$ 代表数据组数，每组测试数据描述如下：

第一行两个整数 n, k $(1 \leqq n \leqq 2 \times 10^5, -10^6 \leqq k \leqq 10^6)$，表示方块个数、初始的万能方块上的数字。

第二行 n 个整数 a_i $(-10^6 \leqq a_i \leqq 10^6)$，表示从左往右数第 i 个方块上写的数字。

保证同一个测试文件中，所有测试数据的 n 之和不超过 2×10^5。

Output

对于每组数据

在单独的一行输出一个整数表示最终：从左往右数第一个方块上的数字 + 万能方块上的数字之和的最大值。

Examples

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

Explanation

对于第一组测试数据，我们操作一次，方块序列变为：$\{5, 1, 2, 3, 3, 2\}$，此时万能方块变为：$k = 1$，总和为 6 最大，可以证明不存在更优的答案。

Problem F. Permutation Counting

Time limit: 2 seconds
Memory limit: 1024 megabytes

这天小红给了小苯一个长度为 n 的排列 p，但她把 p 隐藏了起来，只告诉了小苯 m 条有关 p 的信息，具体的：

- 第 i $(1 \leqq i \leqq m)$ 条信息包含 3 个参数 l_i, r_i, k_i，表示 p 在 $[l_i, r_i]$ 中的最大值等于 k_i。
- 形式化的即：$\max(p_{l_i}, p_{l_i+1}, ..., p_{r_i}) = k_i$。

你的任务则是帮助小苯确定，有多少种可能的 p 都符合上述的所有信息（当然也有可能不存在）。

Input

每个测试文件均包含多组测试数据。第一行输入一个整数 T $(1 \leqq T \leqq 10^5)$ 代表数据组数，每组测试数据描述如下：

第一行两个正整数 n, m $(1 \leqq n, m \leqq 2 \times 10^6)$，分别表示排列 p 的长度 n，以及有关 p 的信息条数。

接下来 m 行，每行三个正整数 l_i, r_i, k_i $(1 \leqq l_i \leqq r_i \leqq n, 1 \leqq k_i \leqq n)$，描述每条信息。

除此之外，保证单个测试文件的 n 之和不超过 2×10^6，m 之和不超过 2×10^6。

Output

对于每组测试数据：

在单独的一行输出一个整数，表示可能的排列个数。（由于结果可能很大，因此输出结果对 998244353 取模的值。）

Examples

standard input	standard output
4 3 1 1 2 2 5 2 2 3 4 2 5 5 5 1 2 3 4 5 1 1 5 4	2 24 36 0

Explanation

对于第一组测试数据，长度为 3 的排列中，满足在区间 $[1, 2]$ 中的最大值等于 2 的排列有：$\{1, 2, 3\}, \{2, 1, 3\}$ 2 个。

对于第四组测试数据，显然长度为 5 的排列，在区间 $[1, 5]$ 中的最大值一定是 5；而信息中说是 4 显然不可能，因此不存在这样的排列，输出 0。

Problem G. Digital Folding

Time limit: 1 second
Memory limit: 256 megabytes

小苯发现了一种特殊的数字运算，称为”数字折叠”。对于一个正整数 x，定义其”折叠数”为：

将 x 的十进制数位翻转并去除前导 0，x 的值更改为翻转后得到的新数。（例如 123 操作后会变为 321，而 120 会变为 21。）

现在小苯拿到了一个区间 $[\mathrm{L}, \mathrm{R}]$，他想知道如果将区间中所有的数字 i $(\mathrm{L} \leqq \mathrm{i} \leqq \mathrm{R})$ 的折叠数都求出，那么其中的最大值是多少，你的任务就是求出这个最大值。

Input

每个测试文件包含多组测试数据。第一行输入一个整数 T $(1 \leqq T \leqq 10^4)$ 代表数据组数，每组测试数据描述如下：

第一行两个整数 L, R $(1 \leqq \mathrm{L} \leqq \mathrm{R} \leqq 10^{15})$ 表示题中所述的区间。

Output

对于每组数据：在单独的一行输出一个整数，表示区间中所有数的”折叠数”的最大值。

Examples

standard input	standard output
3 1 20 1000 10000 1 999	91 9999 999

Explanation

对于第一组测试数据，折叠数最大的数字是 19，其折叠数是 91。

Problem H. Blackboard

Time limit: 1 second
Memory limit: 256 megabytes

小红在黑板上写了一个计算式，具体来讲是 n 个数字 a_i，其中间由 $n-1$ 个加号（+）连接组成。

$$a_1 + a_2 + \cdots + a_n$$

现在小苯想去擦去黑板上的一些 + 运算符，但他擦得很不干净，只擦去了 + 中的 −，剩下的部分就是一个 | 了。巧合的是 | 恰好也是一个运算符。(按位或 or 运算符)

小苯想知道，有多少种不同的擦黑板方式，能使得按照新算式进行计算，结果和擦黑板前的算式计算结果相同，请你帮他算一算。(不擦黑板也是一种方案。)

【注】:

特别的，在本题中我们认为 or 运算符的优先级大于 +。

两种擦黑板方式不同当且仅当存在至少一个运算符位置，其在其中一个方式中为 +，而在另一个方式中为 |（即按位或 or)。

Input

每个测试文件均包含多组测试数据。第一行输入一个整数 T $(1 \leqq T \leqq 100)$ 代表数据组数，每组测试数据描述如下:

第一行包含一个整数 n $(1 \leqq n \leqq 2 \times 10^5)$

第二行包含 n 个正整数 $a_1, a_2, \ldots, a_n$ $(0 \leqq a_i < 2^{31})$

除此之外，保证单个测试文件的 n 之和不超过 2×10^5。

Output

对于每组测试数据:

输出一行一个整数表示不同的擦黑板方案个数，由于结果可能很大，因此输出答案对 998244353 取模的值。

Examples

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

Explanation

对于第一组测试数据:

- $1+2$ 这个计算式，我们可以选择擦中间的 +，就能得到：$1 \mid 2 = 3$，原式运算结果也为 3，因此是一种方案，不替换也是一种方案，共两种，因此输出 2。

Problem I. AND vs MEX

Time limit: 1 second
Memory limit: 256 megabytes

小苯有一个可重数字集合 S，初始为空。现在他可以进行任意次以下操作，给 S 中加入一些元素，具体的：

- 他可以从区间 $[l, r]$ 中选择若干个（至少一个）**不同的**数字，将这些数字的 AND（按位与）加入集合 S。

他可以做任意次上述操作，请问 S 的 mex 最大可以达到多少。

【MEX】整数数组的 mex 定义为没有出现在集合中的最小非负整数。例如 $\text{mex}\{1, 2, 3\} = 0$、$\text{mex}\{0, 2, 5\} = 1$。

Input

每个测试文件均包含多组测试数据。第一行输入一个整数 T $(1 \leqq T \leqq 10^5)$ 代表数据组数，每组测试数据描述如下：

输入一行两个整数 l, r $(0 \leqq l \leqq r < 2^{30})$。

Output

在单独的一行输出一个整数表示最大的 MEX。

Examples

standard input	standard output
3 3 4 1 2 0 10	1 3 11

Explanation

对于第一组测试数据，我们只能选择 3 和 4 将他们的 AND（即 0）加入 S，此时 S 的 $\text{MEX} = 1$。

Problem J. MST Problem

Time limit: 2 seconds
Memory limit: 256 megabytes

小苯拿到了一个 n 个点、$\frac{n\times(n-1)}{2}$ 条边组成的**无向完全图**，没有重边和自环，其中第 i 个点的点权为 a_i。u, v 两点之间的边权为 $a_u + a_v$。

而调皮的小红删除掉了其中一些边，导致图不再是一个完全图，具体来讲有 m 条删除记录，每条记录都有一个点对 (u_j, v_j) 组成，表示小红删除了这条边（注意，记录可能有重复，即可能存在两个记录删除的边是一样的。）

现在你的任务就是求出这张图的最小生成树 的权重（树中所有边的权重之和最小）。如果此时图不存在最小生成树，则输出 -1。

【名词解释】

最小生成树：对于一张由 n 个节点构成的连通图，选出 $n-1$ 条边将所有节点连通，且使得这 $n-1$ 条边的权重之和最小，这样的结构称为最小生成树。

Input

每个测试文件均包含多组测试数据。第一行输入一个整数 T $(1 \leqq T \leqq 10^5)$ 代表数据组数，每组测试数据描述如下：

第一行两个整数 n, m $(2 \leqq n \leqq 3 \times 10^5, 1 \leqq m \leqq 3 \times 10^5)$，分别表示图的节点个数和小红删除边的记录条数。

第二行 n 个整数 a_i $(1 \leqq a_i \leqq 10^9)$，表示每个节点的点权。

接下来 m 行，每行两个正整数 u_j, v_j，描述第 j 条删除记录。

除此之外，保证单个测试文件的 n 之和不超过 3×10^5，m 之和不超过 3×10^5。

Output

对于每组测试数据：

在单独的一行输出一个整数：

如果此时图存在最小生成树，则输出整张图的最小生成树的权重；否则输出一个 -1。

Examples

standard input	standard output
3 3 1 1 2 3 2 3 4 4 2 2 2 2 1 2 2 3 3 4 1 4 3 2 1 2 3 1 2 2 3	7 12 -1

Explanation

对于第一组测试数据，整个图已经被删成一棵树，因此图自己就是自己的唯一生成树，当然也就是最小生成树，仅剩的两条边 $(1,2)$ 和 $(1,3)$ 的权分别为：$1+2=2, 2+3=5$，因此最小生成树权重为 7。

Problem K. Constructive

Time limit: 1 second
Memory limit: 256 megabytes

给定一个正整数 n，你需要构造一个长度为 n 的数组 a，满足以下条件：

1. 数组中的每个元素 a_i 都是正整数
2. 所有元素的乘积等于所有元素的和
3. 数组中的元素**互不相同**

小苯想知道，对于给定的 n，是否存在这样的数组。如果存在，请构造字典序 最小的解；如果不存在报告无解。

【名词解释】

数组的字典序比较：从数组的第一个数字开始逐个比较，直至发现第一个不同的位置，比较这个位置数字的大小关系，数字较小的数组字典序也较小；如果比较到其中一个数组的结尾时依旧全部相同，则较短的数组字典序更小。

例如 $\{2, 3, 4, 5\}$ 的字典序小于 $\{2, 4, 3, 5\}$，也小于 $\{3, 2, 4, 5\}$。

Input

每个测试文件均包含多组测试数据。第一行输入一个整数 T $(1 \leqq T \leqq 100)$ 代表数据组数，每组测试数据描述如下：

一行一个整数 n $(1 \leqq n \leqq 100)$

除此之外，保证单个测试文件中所有 n 之和不超过 1000。

Output

对于每组测试数据：

如果存在满足条件的数组，第一行输出”YES” (不带双引号)，第二行输出**字典序最小的**满足条件的数组，元素间用单个空格分隔；

如果不存在，输出一行”NO” (不带双引号)。

Examples

standard input	standard output
2 1 2	YES 1 NO

Explanation

对于第一组测试数据，数组 $\{1\}$ 显然是唯一符合条件的解。

对于第二组测试数据，可以证明不存在解。

Problem L. Need Zero

Time limit: 1 second
Memory limit: 256 megabytes

小苯拿到了一个正整数 n，现在他希望 n 的个位数是 0，为此他可以执行以下操作**恰好**一次：

- 选择一个正整数 x $(1 \leqq x \leqq 10^5)$，并执行：$n := n \times x$。（其中 := 表示赋值操作。）

你的任务就是帮助小苯找出**最小**的 x。

（可以证明一定存在解。）

Input

第一行一个正整数 n $(1 \leqq n \leqq 10^5)$。

Output

输出一行一个正整数，表示最小的合法解 x。（可以证明在题目的限定范围内一定有解。）

Examples

standard input	standard output
125	2
10	1

Explanation

对于 $n = 125$，我们只需要选择 $x = 2$，就可以将 n 变为 $125 \times 2 = 250$，满足其个位数为 0。显然 2 是最小的正整数解。

对于 $n = 10$，我们选择 $x = 1$ 即可，操作后 n 不变，满足条件。显然 1 是最小的正整数解。