

Problem A. 简单的加法乘法计算题

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

JokerShaco 有一个数字 x ，最开始 $x = 0$ ，他想要把 x 变成 y 。为了达到这个目标，他可以利用两个集合 A 和 B 。其中集合 A 包含 n 个元素，分别是 1 到 n 的所有正整数；集合 B 包含 m 个元素。每次它可以对 x 进行如下任意次操作：

- 选择 A 中的一个元素 a ，令 x 加上 a 。
- 选择 B 中的一个元素 b ，令 x 乘以 b 。

已知 y ， n ， m 和 B 中 m 个元素的具体值，JokerShaco 想知道让 x 变成 y 的最少操作次数。

Input

第一行包含三个整数 y ($1 \leq y \leq 5 \cdot 10^6$)， n ($1 \leq n \leq 5 \cdot 10^6$) 和 m ($1 \leq m \leq 10$)，其含义如题目所述。

第二行包含 m 个正整数，其中第 i 个表示 B 中的第 i 个元素 b_i ($1 \leq b_i \leq 5 \cdot 10^6$)。

Output

输出一个整数，表示让 x 变成 y 的最少操作次数。在题目条件下可知一定能将 x 变成 y 。

Examples

standard input	standard output
10 3 1 2	3
100 6 3 2 3 5	3

Problem B. 逆 KMP

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 256 megabytes

Walk Alone 是一个字符串大师，但是他已经对传统的字符串算法感到无聊，如 KMP 算法，所以他最近在思考逆向的 KMP。下面是他提出的问题：

给你一个长度为 n 的整数序列 a ，对于任意的整数 i ($1 \leq i \leq n$)，满足 $0 \leq a_i < i$ 。你需要构造另一个整数序列 s ，满足以下条件：

- 序列 s 的长度为 n ，并且其中任意元素 s_i 满足 $1 \leq s_i \leq n$ ；
- 对于所有的整数 i ($1 \leq i \leq n$) 和 j ($1 \leq j \leq a_i$)，满足 $s_j = s_{i-a_i+j}$ ；
- 满足上述条件的前提下，序列 s 中出现的不同元素的数量**最多**。

当然，Walk Alone 可以很轻松地解决这道题，但他想把这道题当作对你的考验。

Input

第一行包含一个整数 n ($1 \leq n \leq 2 \cdot 10^5$)，表示序列 a 的长度。
第二行包含 n 个整数，其中第 i 个整数定义为 a_i ($0 \leq a_i < i$)。

Output

输出用空格间隔的 n 个整数，其中第 i 个整数定义为 s_i 。如果存在多个合法答案，请输出字典序最小的那个。

Examples

standard input	standard output
5 0 0 1 2 3	1 2 1 2 1
11 0 0 0 0 2 1 0 0 3 0 1	1 2 3 1 2 1 1 2 3 4 1

Problem C. 排列排序问题

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

JokerShaco 有一个长度为 n 的排列 p ，他认为一个排列必须是有序的，所以他打算将其进行排序。
他可以对这个排列进行如下操作：

- 将这个排列切割成若干个序列（也可以不切割，保持原样），每个序列至少含有一个元素；
- 选择其中一些序列并将它们翻转；
- 将这些序列按照他的意愿重新组合拼接得到一个新的排列。

JokerShaco 认为切割操作非常累，他想知道如果必须把这个排列变得有序，至少需要切割多少次。
一个长度为 n 的排列的定义为，包含从 1 到 n 这 n 个不同的整数的序列，每个整数恰好出现一次。
序列翻转的定义为，假设存在一个长度为 m 的序列 $[a_1, a_2, \dots, a_{m-1}, a_m]$ ，那么将这个序列翻转后将会得到 $[a_m, a_{m-1}, \dots, a_2, a_1]$ 。

Input

第一行包含一个整数 n ($1 \leq n \leq 10^6$)，表示排列 p 的长度。
第二行包含 n 个整数，其中第 i 个整数定义为 p_i 。保证输入的 p 一定是长度为 n 的排列。

Output

输出一个整数，表示将排列 p 变得有序所需的最少切割次数。

Examples

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

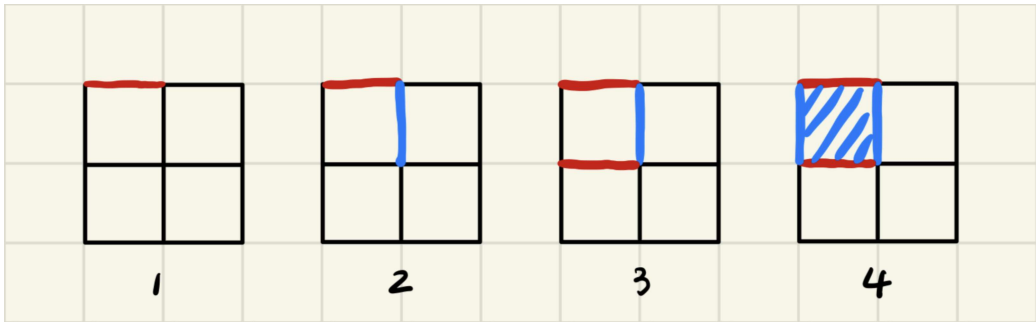
Problem D. 网格染色

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

有一个由 $n \times n$ 个小正方形组成的网格图，其中每个小正方形的边长为 1，Walk Alone 和 Kelin 在这上面进行一场染色游戏，游戏的规则如下：

- Walk Alone 和 Kelin 轮流进行操作，Walk Alone 先手。
- 当轮到 Walk Alone 操作时，他选择一条尚未被染色的正方形边框并将其染成红色。操作结束时，如果这条边框是一个（或两个）正方形最后一条被染色的边框，则自动将该正方形也同样染成红色。
- 当轮到 Kelin 操作时，他选择一条尚未被染色的正方形边框并将其染成蓝色。操作结束时，如果这条边框是一个（或两个）正方形最后一条被染色的边框，则自动将该正方形也同样染成蓝色。
- 当所有的边框都被染色后，游戏结束。此时染成自己所属颜色的正方形数量更多的一方获胜；或者当数量相等时，达成平局。

例如，在 2×2 的网格图中，一种可能的游戏过程如下：



已知网格图的边长 n ，如果两人都积极进行游戏（按照最优策略，尽可能地让自己获胜，或者在自己无法获胜的前提下尽可能地达成平局），请你推断哪一方会获胜或达成平局。

Input

一行包含一个整数 n ($1 \leq n \leq 10^9$)，表示网格图的边长。

Output

如果 Walk Alone 获胜，输出 ‘Walk Alone’（不带引号，下同）；否则如果 Kelin 获胜，输出 ‘Kelin’；否则平局输出 ‘Draw’。

Example

standard input	standard output
1	Kelin

Problem E. 序列配对

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 512 megabytes

你有一个长度为 n 的序列 $a_1, a_2, \dots, a_n$ ，初始序列内任意元素 $a_i = 0$ 。
之后会告诉你 n 组配对信息，每组配对信息形如整数对 (l, r) ，表示将 a_l 和 a_r 进行配对。在配对之后，你必须执行下面两种操作之一（不可全选）：

- 令 a_l 加 1，随后 a_r 减 1。
- 令 a_r 加 1，随后 a_l 减 1。

你得知这些配对信息遵循着一个奇妙的规定：在 n 组整数对内的 $2n$ 个整数中，每个序列的下标都恰好出现 2 次！

此时你想知道，在所有操作方案中，使 $\sum_{i=1}^n a_i^2 = k$ 的方案数，由于答案可能会很大，你只需求出其对 998 244 353 取模后的结果。

Input

第一行包含一个整数 n ($1 \leq n \leq 2 \cdot 10^5$)，表示序列的长度。
接下来 n 行，其中第 i 行包含两个整数 l_i, r_i ($1 \leq l \leq r \leq n$)，表示第 i 组配对信息。保证输入的这 $2n$ 个整数符合题目要求。
最后一行包含一个整数 k ($0 \leq k \leq 10^9$)，其含义如题目所述。

Output

输出一个整数，表示使 $\sum_{i=1}^n a_i^2 = k$ 的方案数对 998 244 353 取模后的结果。

Examples

standard input	standard output
3 1 3 2 3 1 2 0	2
6 2 5 3 6 2 5 4 6 1 3 1 4 8	28

Problem F. 新取模运算

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

在这道题中，我们定义一个新的运算符号 $\oplus$ 并将其称为新取模运算。

当计算 $x \oplus y$ 时，如果 x 不是 y 的倍数，则得到 x 除以 y 的余数；否则令 x 不断除以 y 直到 x 不再是 y 的倍数，假设它为 x' ，然后得到 x' 除以 y 的余数。例如， $4 \oplus 5 = 4$ ， $20 \oplus 5 = 4$ ， $100 \oplus 5 = 4$ 。

给定一个质数 p ，接下来会有多组询问，对于每次询问会给出一个整数 n ，你需要计算出 $n! \oplus p$ 的值。其中 $n!$ 是 n 的阶乘，即所有小于等于 n 的正整数的乘积。

Input

第一行包含两个整数 T ($1 \leq T \leq 10^5$) 和 p ($2 \leq p \leq 10^6$)，分别表示询问的次数和给定的质数。

接下来 T 行，每行包含一个整数 n ($1 \leq n \leq 10^{18}$)，含义如题目所述。

Output

对于每次询问，输出一行包含一个整数，即 $n \oplus p$ 的值。

Examples

standard input	standard output
3 7	4
11	1
45	2
14	
2 10007	3152
1919	3679
810	

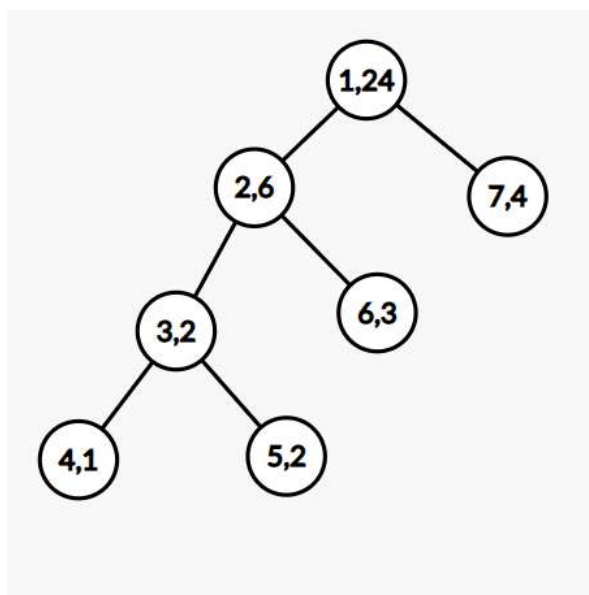
Problem G. 广义线段树

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

对于任意长度为 n 的序列 a ，可以基于 a 建立一个广义线段树。广义线段树是一个有 $(2n - 1)$ 个节点的带权二叉树，对于每个编号为 p 的节点，都有两个属性 L_p 和 R_p ，表示其维护的区间为 $[L_p, R_p]$ ，同时其权值 $M_p = \prod_{i=L_p}^{R_p} a_i$ 。另外，广义线段树还满足如下性质：

- 所有编号为 $p \in [n, 2n - 1]$ 的节点是叶节点，同时 $L_p = R_p = p + 1 - n$ 。
- 所有编号为 $p \in [1, n - 1]$ 的节点是非叶节点，其必定有左儿子 X_p 和右儿子 Y_p ，且 $p < X_p < Y_p$ 。节点 p 维护的区间为左右儿子维护区间之并，且保证维护区间连续。形式化地，有 $R_{X_p} = L_{Y_p} - 1$ ，且 $L_p = L_{X_p}$ ， $R_p = R_{Y_p}$ 。

例如，下面是一个基于 $n = 4$ 的序列 $a = \{1, 2, 3, 4\}$ 建立的广义线段树（节点内整数对 (p, M_p) 分别表示编号和权值）。可以发现，广义线段树的形态并不唯一。



对这个广义线段树而言：

- $[L_7, R_7] = [4, 4]$ ，故 $M_7 = a_4$
- $[L_6, R_6] = [3, 3]$ ，故 $M_6 = a_3$
- $[L_5, R_5] = [2, 2]$ ，故 $M_5 = a_2$
- $[L_4, R_4] = [1, 1]$ ，故 $M_4 = a_1$
- $[L_3, R_3] = [L_4, R_5] = [1, 2]$ ，故 $M_3 = a_1 \times a_2$
- $[L_2, R_2] = [L_3, R_6] = [1, 3]$ ，故 $M_2 = a_1 \times a_2 \times a_3$

- $[L_1, R_1] = [L_2, R_7] = [1, 4]$, 故 $M_1 = a_1 \times a_2 \times a_3 \times a_4$

分别给定长度为 n 的序列 a, b 以及节点数为 $(2n - 1)$ 的广义线段树 T 的形态（即每个节点的左右儿子编号），然后你需要执行 n 次操作，第 i 次操作为将 a_i 变成 $a_i \times b_i$ 。

每次操作结束后，你需要基于修改后的序列 a 建立与 T 形态相同的广义线段树，并求出所有节点的权值和，即 $\sum_{i=1}^{2n-1} M_i$ 。由于结果可能会非常大，你只需要求出其对 998 244 353 取模后的值。

Input

第一行包含一个整数 n ($1 \leq n \leq 5 \cdot 10^5$)，表示序列 a 和 b 的长度。

第二行包含 n 个整数，其中第 i 个整数定义为 a_i ($1 \leq a_i < 998\,244\,353$)。

第三行包含 n 个整数，其中第 i 个整数定义为 b_i ($1 \leq b_i < 998\,244\,353$)。

接下来 $n - 1$ 行，其中第 i 行包含两个整数 X_i, Y_i ($i < X_i < Y_i \leq 2n - 1$)，分别表示节点 i 的左右儿子编号。保证输入的广义线段树形态符合题目要求。

Output

输出一行用空格间隔的 n 个整数，其中第 i 个整数表示第 i 次修改后的答案对 998 244 353 取模后的值。

Example

standard input	standard output
4	75 207 390 974
1 2 3 4	
2 3 2 3	
2 7	
3 6	
4 5	

Note

样例中广义线段树的形态和题面中的例子相同。

第一次修改后， a_1 变为 $a_1 \times b_1 = 1 \times 2 = 2$ ，因而新的 $a = \{2, 2, 3, 4\}$ 。可以计算出：

- $M_7 = a_4 = 4$
- $M_6 = a_3 = 3$
- $M_5 = a_2 = 2$
- $M_4 = a_1 = 2$
- $M_3 = a_1 \times a_2 = 2 \times 2 = 4$
- $M_2 = a_1 \times a_2 \times a_3 = 2 \times 2 \times 3 = 12$
- $M_1 = a_1 \times a_2 \times a_3 \times a_4 = 2 \times 2 \times 3 \times 4 = 48$

故权值之和为 $M_1 + M_2 + \dots + M_7 = 75$ 。

第二次修改后， a_2 变为 $a_2 \times b_2 = 2 \times 3 = 6$ 。后续的操作与第一次操作类似，此处不再赘述。

Problem H. 狭义线段树

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

你打算建立一棵有 $(2n - 1)$ 个节点，其中有 n 个叶子节点的二叉树。具体地，建立这棵树的伪代码如图所示。

```

cnt ← 0
build(1, n, 1)
Function build(l, r, p):
  cnt ← cnt + 1
  id ← cnt
  fatherid ← p
  // fatherid is the father node of node id
  if l = r then
    | return
  end
  mid ← ⌊(l + r) / 2⌋
  build(l, mid, id)
  build(mid + 1, r, id)
end
  
```

不难发现，节点 1 是根节点，并且所有节点的编号与其 DFS 序相同。另外，你认为叶子节点十分重要，因此你按照节点编号从小到大又把这 n 个叶子节点分别称为 1 号叶子，2 号叶子， $\dots$ ， n 号叶子。叶子节点会被其上方的节点管辖，具体来说，如果 i 号叶子在节点 j 的子树内，则称节点 j 管辖 i 号叶子，不妨用 $g(j, i) = 1$ 表示；否则若节点 j 不管辖 i 号叶子，则 $g(j, i) = 0$ 。注意，叶子节点同时也管辖自己本身。

同时，你认为一个好的二叉树要有点权，于是对于节点 i ，定义其点权为 v_i 。初始所有节点的点权都为 0。

在一次梦中，你正在改造这棵二叉树。你将会对这棵二叉树依次执行 q 次操作，每次操作的格式和描述如下：

- 1 $s\ t\ v$: 对于所有的整数 i ($i \in [s, t]$)，将节点 i 的点权加 v 。
- 2 $s\ t\ v$: 令 $\mathcal{S} = \bigcup_{i \in [s, t]} S_i$ ，其中 S_i 表示节点 i 管辖的叶子节点的集合。然后对于 $\mathcal{S}$ 中所有叶子节点，将其点权加 v 。注意 $\mathcal{S}$ 是不重复集合，即在本次操作中，每个叶子节点最多被修改一次。
- 3 $l\ r$: 计算 $\sum_{i=l}^r f(i) \bmod 998\,244\,353$ ，其中 $f(i)$ 表示管辖 i 号叶子的所有节点的点权之和，即 $f(i) = \sum_{g(j, i)=1} v_j$ 。

你还想再加点操作，但是早八的铃声把你吵醒了，不过你还是决定实现一下这个奇思妙想。

Input

第一行包含一个整数 n ($3 \leq n \leq 10^5$)，表示二叉树中叶子节点的数量。

第二行包含 $(2n - 2)$ 个整数，其中第 i 个整数表示节点 $i + 1$ 的父亲节点编号。保证给出的树的形态与节点编号和题目所描述的相同。

第三行包含一个整数 q ($3 \leq q \leq 10^5$)，表示操作次数。

接下来 q 行，每行第一个整数 opt ($opt \in [1, 3]$) 表示操作类型，若 $opt = 1$ 或 $opt = 2$ ，则紧跟三个整数 s, t, v ($1 \leq s \leq t \leq 2n - 1, 0 \leq v < 998\,244\,353$)，表示一次修改操作；若 $opt = 3$ ，则紧跟两个整数 l, r ($1 \leq l \leq r \leq n$)，表示一次询问操作。所有参数的含义如题目所述。

Output

对于每个 $opt = 3$ 的操作，输出一行一个整数，表示本次询问的结果。

Example

standard input	standard output
5	18
1 2 3 3 2 1 7 7	29
5	38
1 2 4 3	
3 1 5	
2 5 7 5	
3 2 5	
3 1 5	

Problem I. Fujisaki 讨厌数学

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

众所周知，Fujisaki 在微积分（一）（上）、微积分（一）（下）、线性代数、离散数学（一）、概率论与数理统计这些课程中，只取得了非常低的分数，这导致他十分痛恨数学，因此他给大家带来了一道数学题。

已知 $x + x^{-1} = k$ ，其中 k 是一个整数且 $k \geq 2$ ，Fujisaki 想让你帮他求出 $x^n + x^{-n}$ 的值。可以证明对于任意的整数 $n \geq 0$ ，这个值都是一个整数。由于结果可能会非常大，你只要求出其对 M 取模后的值。

Input

一行包含三个整数 M ($3 \leq M \leq 998\,244\,353$)， k ($2 \leq k < M$) 和 n ($0 \leq n \leq 10^{18}$)，其含义如题目所述。

Output

输出一个整数，表示答案对 M 取模后的值。

Examples

standard input	standard output
998244353 10 1	10
998244353 2 3	2
100 4 5	24

Problem J. 基因编辑

Input file: standard input
Output file: standard output
Time limit: 1.5 seconds
Memory limit: 512 megabytes

绮月有 n 条 DNA 碱基序列 $S_1, S_2, \dots, S_n$ ，每条碱基序列可以用一个仅包含 ‘A’、‘C’、‘G’ 和 ‘T’ 这四种大写字母的字符串表示。

绮月可以拼合两条 DNA 碱基序列，具体操作为将前一条碱基序列的一个前缀（可以为空）和后一条的一个后缀（可以为空）结合，如 ‘ACGC’ 与 ‘CTAT’ 拼合就有可能得到 ‘ACGCTAT’、‘ACGCCTAT’、‘ACAT’ 或 ‘T’。

绮月据此定义，一个三元组 (i, j, k) 是好的，当且仅当 $1 \leq i, j, k \leq n$ ， $i \neq k$ ， $j \neq k$ ，且 S_i 与 S_j 拼合可以得到 S_k 。

绮月想知道好的三元组的数量。

Input

第一行包含一个整数 n ($3 \leq n \leq 2 \times 10^5$)，表示碱基序列的数量。

接下来 n 行，每行包含一个字符串，其中第 i 个字符串定义为碱基序列 S_i ($1 \leq |S_i| \leq 2 \times 10^6$)。保证 $\sum |S_i| \leq 2 \times 10^6$ 。

Output

输出一个整数，表示好的三元组的数量。

Examples

standard input	standard output
3 AAA AA AA	12
3 ACGC CTAT ACAT	1
4 A C T G	0

Problem K. 不定项选择题

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

有一道共 n 个选项的不定项选择题，它的答案至少包含一个选项，由于题目与选项的内容晦涩难懂，你打算通过尝试每一种可能的答案来通过这道题。

初始时所有选项都没有被勾选，你可以执行任意次下述操作：

- 勾选一个当前未被勾选的选项。
- 取消勾选一个当前已被勾选的选项。

当你勾选中的选项与答案一致时，你将会立即通过此题。在开始尝试前，你产生了一个疑问：在最坏的情况下，至少需要操作多少次才可以通过这道题？

以下关于软件配置项之间的关系，哪些陈述是正确的。

- ☐ A、任何软件配置项都不应是孤立的，否则这样的软件配置项没有意义和价值
- ☐ B、软件项目中可能会存在孤立的、与任何其他软件配置项没有关联关系的软件配置项
- ☐ C、如果二个软件配置项之间存在关联关系，那么对一个软件配置项的修改可能会影响另一个软件配置项。
- ☐ D、软件配置项及其之间的关系可以用树来加以表示。

Input

一行包含一个整数 n ($1 \leq n \leq 20$)，表示选项的数量。

Output

输出一个整数，表示在最坏情况下通过题目的最少操作次数。

Examples

standard input	standard output
1	1
2	3
3	7
15	32767

Note

图片与题目无关。

Problem L. Azur Lane

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

你是港区的一位指挥官，你可以通过培养指挥喵来增强舰队的实力，指挥喵由对应的喵箱培育。喵箱有 k 个等级，等级越高稀有度越高。你每天会获得若干个喵箱（至少一个），然后再使用“一键放入”将这些喵箱放到喵窝里进行培育。“一键放入”会优先放入稀有度更高的喵箱（即按照稀有度从高到低排序后依次放入），放入的喵箱会排在已放入喵箱的后面，在**每天结束时**系统会自动扣除与喵窝中喵箱数量相等的钱。最开始时你的喵窝里没有喵箱。

因为你非常懒，所以你会等 n 天后才把喵箱一起打开。在你 n 天后进入喵窝时，你看到了 m 个喵箱，用一个长度为 m 的序列 a 表示喵窝里按放入先后顺序排列的喵箱的等级。但是你已经忘了这 n 天里你每天获得了多少喵箱了，你甚至忘了 n 的值是多少，显然这有许多情况。你想知道，对于满足 $1 \leq n \leq m$ 的所有整数 n ，在所有情况中，这 n 天总共扣除的钱数至少为多少？

Input

第一行包含两个整数 m ($1 \leq m \leq 10^6$), k ($1 \leq k \leq 10^6$), 分别表示喵窝中喵箱的数量和喵箱的最高等级。

第二行包含 m 个整数 $a_1, a_2, \dots, a_m$ ($1 \leq a_i \leq k$), 表示喵窝中排列着的喵箱的等级。

Output

输出一行用空格间隔的 m 个整数，其中第 i 个整数表示当 $n = i$ 时在所有可能的情况下至少会扣除的钱数。如果没有合法的情况，请输出 -1 。

Examples

standard input	standard output
3 3 2 3 1	-1 4 6
8 4 3 2 4 2 1 2 3 2	-1 -1 -1 21 22 25 29 36

Note

样例一解释：

当 $n = 1$ 时，无法在一天内获得这 3 个喵箱（第二个喵箱的等级大于第一个喵箱的等级，不符合“一键放入”从高到低放入的原则），因此输出 -1 。

当 $n = 2$ 时，第一天获得第一个喵箱，此时喵窝内有 1 个喵箱，当天花费为 1；第二天获得后两个喵箱，此时喵窝内有 3 个喵箱，当天花费为 3。因此总钱数为 4。

当 $n = 3$ 时，每天依次获得一个喵箱，总钱数为 $1 + 2 + 3 = 6$ 。

Problem M. 近似递增序列

Input file: standard input
Output file: standard output
Time limit: 6 seconds
Memory limit: 512 megabytes

对于一个长度为 m ($m \geq 1$) 的整数序列 $a_1, a_2, \dots, a_m$ ($a_i > 0$)，如果最多只存在一个整数 p ($1 \leq p < m$) 满足 $a_p \geq a_{p+1}$ ，则可以称这样的序列为近似递增序列，同时我们定义这个近似递增序列的权值为 $\prod_{i=1}^m a_i$ 。

设 $f(i)$ 表示权值为 i 的近似递增序列的数量，duoluoluo 想知道 $\sum_{i=1}^n f(i)$ 的值，但是他连 $f(2)$ 都不会计算，你可以帮帮他吗？由于答案可能会非常大，你只需要求出其对 998 244 353 取模后的值。

Input

一行包含一个整数 n ($1 \leq n \leq 10^8$)，其含义如题目所述。

Output

输出一个整数，表示 $\sum_{i=1}^n f(i)$ 对 998 244 353 取模后的值。

Examples

standard input	standard output
2	7
5	26

Note

样例一中 7 个近似递增序列为：{1}，{1, 1}，{1, 1, 2}，{1, 2}，{1, 2, 1}，{2}，{2, 1}。

Problem N. A+B problem

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

Walk Alone 是一个字符串大师，在他的眼中每个字符串都表示一个 26 进制的数，具体地说，按照字母表顺序从 A 到 Z 分别对应着十进制下 0 到 25，那么 BA 对应十进制下 $26 (1 \times 26 + 0)$ ， BB 对应十进制下 $27 (1 \times 26 + 1)$ ，以此类推。

Walk Alone 想考考已经习惯了十进制加法的你，给你两个字符串 s 和 t ，请你计算出他们所表示的 26 进制数的和，即 $s + t$ ，并同样以字符串形式告诉他，例如 $A + A = A$ ， $B + C = D$ 。为了简化问题， s 和 t 分别只包含一个大写字母，但请注意，答案不一定只包含一个大写字母。

Input

一行用空格分隔的两个大写字母，分别表示 s 和 t 。

Output

输出一行只包含大写字母的字符串，表示 $s + t$ 。输出的答案不包含前导 0，即类似 AA 、 AB 的输出都是不合法的。

Examples

standard input	standard output
A A	A
B C	D
Z B	BA