

# Problem A. A

Time limit 1000 ms  
Mem limit 524288 kB

## 题目描述

在一个生物实验室中进行了一项实验。起初，科学家们有  $n$  个冷冻细菌，编号从 1 到  $n$ 。

根据实验计划，编号为  $i$  的冷冻细菌将在实验开始后  $a_i$  秒进入培养皿。如果有多个细菌同时进入，它们会同时进入。

一旦冷冻细菌进入培养皿，它会解冻并开始成熟。编号为  $i$  的细菌成熟需要  $t_i$  秒。一旦细菌成熟，它会立即变成两个成熟细菌，然后每个成熟细菌在每秒末再次分裂成两个成熟细菌。

菌落的大小是指培养皿中细菌的总数。实验的目标是确定经过多少秒后，菌落的大小恰好等于  $m$ 。

请帮助科学家们确定所需的秒数，或者确定菌落的大小永远不会恰好等于  $m$ 。

## 输入格式

第一行包含两个整数  $n$  和  $m$  ( $1 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq 10^9$ )，分别表示冷冻细菌的数量和期望的菌落大小。

第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ )，表示冷冻细菌进入培养皿的时间。

第三行包含  $n$  个整数  $t_1, t_2, \dots, t_n$  ( $1 \leq t_i \leq 10^9$ )，表示冷冻细菌的成熟时间。

## 输出格式

如果菌落的大小永远不会等于  $m$ ，输出 **-1**。

否则，输出实验开始后经过的秒数，使得菌落的大小恰好等于  $m$ 。

## 样例 1

Input	Output
4 11 3 5 1 10 2 9 2 13	5

样例 2

Input	Output
13 124 5 6 8 8 1 6 4 6 4 7 10 3 9 5 2 10 5 2 1 1 4 8 3 4 1 9	8

数据范围与提示

详细子任务附加限制及分值如下表所示。

子任务	分值	附加限制	子任务依赖
1	13	$m \leq n, a_i \leq 10^5, t_i = 10^9$	
2	14	$a_i = i, t_i$ 相同	
3	17	$n, a_i, t_i \leq 3000$	
4	23	$a_i$ 都等于 1	
5	33	无附加限制	1 ~ 4

# Problem B. B

Time limit 1000 ms  
Mem limit 524288 kB

## 题目描述

在玛莎的生日那天，她像往常一样收到一个包含  $n$  个自然数的数组  $a$ ，其中每个数字都在 1 到  $m$  之间。玛莎非常喜欢数字 3，因此数组的长度是 3 的倍数。

玛莎决定将数字组合成**三元组**：每个三元组要么由三个相同的数字组成，要么由三个连续的数字组成。换句话说，每个三元组的形式要么是  $(x, x, x)$ ，要么是  $(x, x + 1, x + 2)$ ，其中  $x$  是某个自然数。

玛莎想要玩这个礼物数组，她想知道有多少种方法可以将数组中的数字划分为这样的三元组。如果两个划分方式不能通过一一对应的方式使得每个三元组中的数字相同，则认为这两种划分方式不同。由于划分的数量可能非常大，玛莎只需要知道其对  $10^9 + 7$  取模的结果。

请帮助玛莎计算将数组划分为三元组的方法数，并对  $10^9 + 7$  取模。

## 输入格式

第一行输入包含两个整数  $n$  和  $m$  ( $1 \leq n \leq 5000, 1 \leq m \leq 5000, n = 3 \cdot k$  对于某个自然数  $k$ )。

第二行包含  $n$  个整数  $a_i$  ( $1 \leq a_i \leq m$ )，表示数组中的数字。

## 输出格式

输出一个整数，表示将数组划分为三元组的方法数，并对  $10^9 + 7$  取模。

## 样例 1

Input	Output
9 4 3 4 2 4 4 2 3 3 2	2

在第一个样例中，有两种方法可以将数字划分为三元组： $\{(2, 2, 2), (3, 3, 3), (4, 4, 4)\}$  和  $\{(2, 3, 4), (2, 3, 4), (2, 3, 4)\}$ 。

样例 2

Input	Output
6 3 1 2 3 1 2 1	0

数据范围与提示

详细子任务附加限制及分值如下表所示。

子任务	分值	附加限制	子任务依赖
1	10	$m \leq 3$	
2	8	$m \leq 4$	1
3	10	每个数字从 1 到 $m$ 最多出现两次	
4	12	数组 $a$ 不包含能被 4 整除的数字	1
5	29	$n \leq 500m \leq 500$	
6	31	无附加限制	1, 2, 3, 4, 5

# Problem C. C

Time limit 2000 ms  
Mem limit 524288 kB

## 题目描述

我们称两个字符串  $s$  和  $t$  是等价的，如果对于任意长度为 2 的字符串  $u$ ， $u$  在  $s$  中出现的次数与在  $t$  中出现的次数相同。例如，字符串 `aaaba`、`abaaa` 和 `baaab` 彼此等价（`aa` 出现两次，`ab` 出现一次，`ba` 出现一次，`bb` 不出现），而字符串 `abb` 和 `bba` 则不等价。

在这个问题中，给定  $Q$  个由字符 `a`、`b` 和 `c` 组成的字符串，对于每个字符串，需要计算有多少个与其等价的非空字符串，这些字符串也由 `a`、`b` 和 `c` 组成。由于这个数量可能非常大，需要输出其对  $10^9 + 7$  取模的结果。

## 输入格式

第一行输入一个整数  $G$ ，表示当前测试点所属的子任务编号。对于样例， $G = 0$ 。  
第二行输入一个整数  $q$  ( $1 \leq q \leq 10^5$ )，接下来是  $q$  行，每行是一个由字符 `a`、`b` 和 `c` 组成的字符串。这些字符串的总长度不超过  $10^6$ 。

## 输出格式

输出  $q$  个整数，对于每个字符串，输出与其等价的字符串数量对  $10^9 + 7$  取模的结果。

## 样例

Input	Output
0	3
4	3
abaa	2
abca	1
ccbca	
bacc	

字符串 ``abaa`` 等价的字符串有 ``abaa``、``aaba``、``baab``；字符串 ``abca`` 等价的字符串有 ``abca``、``bcab``、``cab``；字符串 ``ccbca`` 等价的字符串有 ``ccbca`` 和 ``cbcca``；字符串 ``bacc`` 仅等价于 ``bacc``。

数据范围与提示

详细子任务附加限制及分值如下表所示。

子任务	分值	附加限制	子任务依赖
1	11	字符串 $s$ 不包含字符 $c$	
2	13	字符串 $s$ 中 $a$ 和 $c$ 不相邻	1
3	11	$n_i \leq 13$	
4	10	$L \leq 40$	3
5	9	$L \leq 60$	3, 4
6	13	$w \leq 100; L \leq 10^5$	
7	33	无附加限制	1 ~ 6

## Problem D. D

**Time limit** 2000 ms

**Mem limit** 524288 kB

### 题目描述

二叉树是一组顶点，每个顶点可以有左孩子和右孩子。其中一个顶点是树的根，它不是任何其他顶点的孩子。从根开始，每次移动到一个孩子，可以到达任何顶点。可以从给定顶点到达的所有顶点的集合称为其子树。

二叉树有三种主要遍历方式：前序遍历（pre-order）、中序遍历（in-order）和后序遍历（post-order）。

前序遍历的顺序是通过以下递归算法获得的：

1. 将树的根添加到遍历序列中。
2. 如果根有左孩子，记录其子树的前序遍历。
3. 如果根有右孩子，记录其子树的前序遍历。

在中序遍历中，根在其孩子子树的遍历之间被记录，而在后序遍历中，根在其孩子子树的遍历之后被记录。

我们可以将这三种遍历方式进行概括：在每个顶点记录一个从  $-1$  到  $1$  的整数  $x$ ，表示我们在何时记录这个顶点：

- $x = -1$ ：在遍历其孩子子树之前；
- $x = 0$ ：在遍历其孩子子树之间；
- $x = 1$ ：在遍历其孩子子树之后。

因此，如果所有顶点记录的是  $-1$ ，则遍历是前序遍历；如果是  $0$ ，则是中序遍历；如果是  $1$ ，则是后序遍历。

考虑一个有  $n$  个顶点的树，顶点编号从  $1$  到  $n$ 。树的根是顶点  $1$ 。最初，所有顶点记录的数字都是  $-1$ 。

在研究中，需要处理  $q$  个以下类型的查询：

1. 将顶点  $l, l+1, \dots, r$  的数字更改为  $x$  ( $x$  可以是  $-1, 0$  或  $1$ )。
2. 报告在当前遍历中顶点  $i$  的位置。

需要输出所有第二种类型查询的答案。

输入格式

第一行输入包含两个整数  $n$  和  $q$  ( $1 \leq n, q \leq 10^5$ )。

接下来的  $n$  行中，每行包含两个整数  $L_i$  和  $R_i$  ( $0 \leq L_i, R_i \leq n$ )，分别表示顶点  $i$  的左孩子和右孩子的编号，如果没有对应的孩子则为 0。

保证  $L_i$  和  $R_i$  构成一个有效的二叉树。

接下来的  $q$  行中，每行表示一个查询。第一位是整数  $t$  ( $t \in \{1, 2\}$ )，表示查询的类型。

如果是第一种类型的查询，接下来是三个整数  $l, r$  和  $x$  ( $1 \leq l \leq r \leq n, x$  可以是  $-1, 0$  或  $1$ )，表示更改顶点范围和新值。

如果是第二种类型的查询，接下来是一个整数  $i$  ( $1 \leq i \leq n$ )，表示需要输出顶点  $i$  在遍历中的位置。

输出格式

对于每个第二种类型的查询，输出一个从 1 到  $n$  的整数，表示对应顶点在遍历中的位置。

样例

Input	Output
5 5 3 4 0 0 5 2 0 0 0 0 2 2 1 1 3 1 2 5 1 3 3 0 2 3	4 1 2

在样例中，遍历顺序如下变化：

- [1, 3, 5, 2, 4]
- [5, 2, 3, 4, 1]
- [5, 3, 2, 4, 1]

数据范围与提示

详细子任务附加限制及分值如下表所示。



子任务	分值	附加限制	子任务依赖
1	10	$n, q \leq 5000$	
2	5	$q_1 \leq 10$	
3	10	所有第一种类型的查询都在所有第二种类型的查询之前	
4	10	所有叶子（没有孩子的顶点）在距离根相同的距离处，没有只有一个孩子的顶点	
5	10	所有第一种类型的查询中 $l = r$	
6	20	所有第一种类型的查询中 $x \in \{-1, 1\}$ ，每个顶点最多有一个孩子	
7	10	所有第一种类型的查询中 $x \in \{-1, 1\}$	6
8	10	每个顶点最多有一个孩子	6
9	15	无附加限制	1 ~ 8