Задача 5. Разбиение массива

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дан массив $A = [a_1, a_2, \ldots, a_n]$, содержащий n натуральных чисел.

Требуется раскрасить элементы массива в два цвета таким образом, чтобы не существовало двух элементов x и y одного цвета, таких, что x нацело делился на y и выполнялось равенство $\frac{x}{y} = p$, где p — *простое* число. Гарантируется, что такая раскраска существует.

Напомним, что целое число $p > 1$ называется простым, если оно имеет ровно два делителя: 1 и p.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 100\,000$) — количество элементов в массиве.

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^6$) — элементы массива.

Формат выходных данных

Выведите описание разбиения массива на два множества в следующем формате.

Выведите n целых чисел, i-е из которых равняется 1, если элемент a_i надо раскрасить в первый цвет, и 2, если элемент a_i надо раскрасить во второй цвет.

Если существует несколько подходящих раскрасок, вы можете вывести любую из них.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	9	$a_i \leqslant 2$ для всех i		первая ошибка
2	19	Гарантируется, что все a_i являются степенями некоторого простого числа p		первая ошибка
3	12	$a_i \leqslant 3$ для всех i	1	первая ошибка
4	13	$a_i \leqslant 4$ для всех i	1, 3	первая ошибка
5	21	$n \leqslant 10$		первая ошибка
6	26	нет	1–5	первая ошибка

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4	2 1 1 2
1 20	1

Замечание

В первом примере есть два элемента первого цвета: 2 и 3, и два элемента второго цвета: 1 и 4. Элементы первого цвета не делятся нацело друг на друга. 4 нацело делится на 1, но их отношение не является простым числом.

Задача 6. Бактерии

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В биологической лаборатории проводят эксперимент. В начале у ученых есть n *замороженных* бактерий, пронумерованных от 1 до n.

Согласно плану эксперимента замороженная бактерия с номером i попадёт в чашку Петри через a_i секунд после начала эксперимента. Если таких бактерий несколько, они все попадают туда одновременно.

Как только замороженная бактерия оказывается в чашке Петри, она размораживается и начинает *созревать*. Созревание бактерии с номером i занимает t_i секунд. Как только бактерия созрела, она начинает размножаться: немедлено превращается в две *созревшие* бактерии, и затем каждая созревшая бактерия в конце каждой секунды снова делится на две созревшие бактерии.

Размером колонии называется общее количество бактерий в чашке Петри. Цель эксперимента — определить, через сколько секунд размер колонии будет в точности равен m.

Помогите ученым определить искомое число секунд или выясните, что размер колонии никогда не будет в точности равен m.

Формат входных данных

В первой строке даны целые числа n, m ($1 \leqslant n \leqslant 2 \cdot 10^5$, $1 \leqslant m \leqslant 10^9$) — количество замороженных бактерий и желаемый размер колонии.

Во второй строке даны n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^9$) — времена перемещения замороженных бактерий в чашку Петри.

В третьей строке даны n целых чисел $t_1, t_2, \ldots, t_n$ ($1 \leqslant t_i \leqslant 10^9$) — продолжительность созревания замороженных бактерий.

Формат выходных данных

Если размер колонии никогда не будет равен m, выведите -1.

В противном случае выведите число секунд после начала эксперимента, через которое размер колонии будет в точности равен m.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	13	$m \leqslant n$, $a_i \leqslant 10^5$, $t_i = 10^9$		первая ошибка
2	14	$a_i = i$, t_i равны		первая ошибка
3	17	$n, a_i, t_i \leqslant 3000$		первая ошибка
4	23	a_i равны 1		первая ошибка
5	33	—	1–4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
4 11 3 5 1 10 2 9 2 13	5
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

Замечание

Рассмотрим, как развивается эксперимент в первом примере.

Время	**Бактерия 1**	**Бактерия 2**	**Бактерия 3**	**Бактерия 4**	**Всего**
0	заморожена	заморожена	заморожена	заморожена	0
1	заморожена	заморожена	в чашке Петри, созревает	заморожена	1
2	заморожена	заморожена	в чашке Петри, созревает	заморожена	1
3	в чашке Петри, созревает	заморожена	в чашке Петри, созрела, 2 бактерии	заморожена	3
4	в чашке Петри, созревает	заморожена	в чашке Петри, созрела, 4 бактерии	заморожена	5
5	в чашке Петри, созрела, 2 бактерии	в чашке Петри, созревает	в чашке Петри, созрела, 8 бактерий	заморожена	11

Задача 7. Разбиение на тройки

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

На день рождения Маше как обычно подарили массив a из n натуральных чисел, в котором каждое число находится в пределах от 1 до m включительно. Маша очень любит число три, поэтому длина массива делится на три.

Маша решила объединять числа в *тройки*: каждая тройка чисел должна состоять или из трех одинаковых чисел, или из трех последовательных чисел. Другими словами, каждая тройка имеет или вид (x, x, x), или $(x, x+1, x+2)$, где x — какое-то натуральное число.

Маша хочет поиграть с подаренным массивом, и ее интересует количество способов разбить числа этого массива на такие тройки. Два способа разбиения считаются различными, если нельзя установить взаимно-однозначное соответствие между тройками первого разбиения и тройками второго разбиения, что числа внутри соответствующих троек равны. Так как количество разбиений может быть большим, Маше достаточно знать его остаток по модулю $10^9 + 7$.

Помогите Маше посчитать количество способов разбить числа подаренного ей массива на тройки по модулю $10^9 + 7$.

Формат входных данных

Первая строка входных данных содержит два целых числа n и m ($1 \leqslant n \leqslant 5000$, $1 \leqslant m \leqslant 5000$, $n = 3 \cdot k$ для какого-то натурального k).

Вторая строка содержит n целых чисел a_i — числа массива ($1 \leqslant a_i \leqslant m$).

Формат выходных данных

В единственной строке одно число — количество способов разбить числа массива на тройки по модулю $10^9 + 7$.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	10	$m \leqslant 3$		первая ошибка
2	8	$m \leqslant 4$	1	первая ошибка
3	10	каждое число от 1 до m встречается не более двух раз		первая ошибка
4	12	массив a не содержит чисел, которые делятся на 4	1	первая ошибка
5	29	$n \leqslant 500$, $m \leqslant 500$		первая ошибка
6	31	—	1, 2, 3, 4, 5	первая ошибка

Примеры

стандартный ввод	стандартный вывод
9 4 3 4 2 4 4 2 3 3 2	2
6 3 1 2 3 1 2 1	0

Замечание

В первом примере числа можно разбить на тройки двумя способами: $\{(2, 2, 2), (3, 3, 3), (4, 4, 4)\}$ и $\{(2, 3, 4), (2, 3, 4), (2, 3, 4)\}$.

Задача 8. Обходы бинарного дерева

Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Бинарное дерево — это набор вершин, у каждой из которых может быть левый и правый ребёнок. Одна из вершин является корнем дерева, она не является ребёнком какой-то другой. Начав в корне и каждый раз переходя в одного из детей, можно дойти до любой вершины. Множество вершин, до которых можно дойти из заданной, называется её поддеревом.

У бинарного дерева есть три основных обхода: прямой (*pre-order*), центрированный (*in-order*) и обратный (*post-order*).

Прямой обход дерева — это порядок его вершин, полученный следующим рекурсивным алгоритмом:

1. Добавить корень дерева в обход.
2. Если у корня есть левый ребёнок, выписать прямой обход его поддерева.
3. Если у корня есть правый ребёнок, выписать прямой обход его поддерева.

В центрированном обходе корень дерева выписывается между обходами поддеревьев его детей, в обратном — после обходов поддеревьев его детей.

Обобщим эти три варианта обхода: пусть в каждой вершине записано целое число x от -1 до 1, обозначающее, в какой момент мы выписываем эту вершину, а именно:

- $x = -1$: до обходов поддеревьев её детей;
- $x = 0$: между обходами поддеревьев её детей;
- $x = 1$: после обходов поддеревьев её детей.

Таким образом, если во всех вершинах записано -1, обход является прямым, если 0 — центрированным, если 1 — обратным.

Рассмотрим дерево с n вершинами, пронумерованных от 1 до n. Корень дерева — вершина 1. Изначально во всех вершинах записано число -1.

В рамках исследования необходимо обработать q запросов одного из следующих типов:

1. Поменять числа в вершинах $l, l+1, \ldots, r$ на x (x равен -1, 0 или 1).
2. Сообщить, на какой позиции в текущем обходе будет стоять вершина i.

Необходимо вывести ответы на все запросы второго типа.

Формат входных данных

В первой строке входных данных даны два целых числа n и q ($1 \leqslant n, q \leqslant 100\,000$).

В следующих n строках даны по два целых числа L_i и R_i ($0 \leqslant L_i, R_i \leqslant n$) — номер левого и правого ребёнка вершины i соответственно, либо 0, если соответствующий ребёнок отсутствует.

Гарантируется, что L_i и R_i задают корректное бинарное дерево.

В следующих q строках даны запросы. Первое число в строке t ($t \in \{1, 2\}$) — тип запроса.

В случае запроса первого типа далее даны целые числа l, r и x ($1 \leqslant l \leqslant r \leqslant n$, x равен -1, 0 или 1) — границы отрезка вершин, в которых меняются числа, и новое значение.

В случае запроса второго типа далее дано число i ($1 \leqslant i \leqslant n$) — номер вершины, позицию которой в обходе необходимо вывести.

Формат выходных данных

На каждый запрос второго типа выведите единственное число от 1 до n — позицию соответствующей вершины в обходе.

Система оценки

Пусть q_1 — количество запросов первого типа.

Подзадача	Баллы	Дополнительные ограничения	Необх. подзадачи	Информация о проверке
1	10	$n, q \leqslant 5000$		первая ошибка
2	5	$q_1 \leqslant 10$		первая ошибка
3	10	все запросы первого типа идут до всех запросов второго типа		первая ошибка
4	10	все листья (вершины без детей) находятся на одном расстоянии от корня, нет вершин с ровно одним ребёнком		первая ошибка
5	10	$l = r$ для всех запросов первого типа		первая ошибка
6	20	$x \in \{-1, 1\}$ для всех запросов первого типа, у каждой вершины не более одного ребёнка		первая ошибка
7	10	$x \in \{-1, 1\}$ для всех запросов первого типа	6	первая ошибка
8	10	у каждой вершины не более одного ребёнка	6	первая ошибка
9	15	нет	1–8	первая ошибка

Пример

стандартный ввод	стандартный вывод
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

Замечание

В примере обход меняется следующим образом:

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