Задача A. Элементарные частицы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Марсиане активно занимаются межпланетной торговлей. А город Олимп-Сити, известный своим космодромом, давно стал местом, куда поступают товары со всех уголков Галактики. Чтобы перевозить еще больше грузов с далеких планет, марсианам необходимы быстрые космические корабли.

Группа ученых как раз проводит эксперименты, чтобы построить быстрый двигатель для нового космического корабля. В текущем эксперименте участвует последовательность из n элементарных частиц, где i-я частица имеет тип a_i.

Назовем подотрезком ($1 \leqslant l \leqslant r \leqslant n$) последовательности частиц ($a_1, a_2, \ldots, a_n$) последовательность ($a_l, a_{l+1}, \ldots, a_r$) для некоторой левой границы l и правой границы r ($1 \leqslant l \leqslant r \leqslant n$). Например, у последовательности (1 4 2 8 5 7) ее подотрезком при $l = 2$ и $r = 4$ является последовательность (4 2 8). Также скажем, что два подотрезка являются различными, если хотя бы одна из границ этих подотрезков не совпадает.

Обратите внимание, что подотрезки могут быть одинаковыми как последовательности, но при этом считаться различными. Например, возьмем последовательность (1 1 1 1 1) и два подотрезка: один с $l = 1$ и $r = 3$, другой — с $l = 2$ и $r = 4$. Оба подотрезка равны (1 1 1), но при этом считаются различными, поскольку их левая и правая границы не совпадают.

Ученые хотят провести реакцию и получить два различных подотрезка одинаковой длины. Обозначим эту длину k. При этом полученная пара подотрезков должна быть *гармоничной*, т. е. для некоторого i ($1 \leqslant i \leqslant k$) должно оказаться так, что типы частиц на i-й позиции совпадают. Например, пара подотрезков (1 7 3) и (4 7 8) гармонична, т. к. у обоих подотрезков на второй позиции стоит 7, а пара подотрезков (1 2 3) и (3 1 2) — нет.

Чем длиннее получатся гармоничные подотрезки, тем больше шансов, что ученые смогут сконструировать быстрый двигатель. Поэтому они попросили Вас помочь рассчитать максимальную возможную длину гармоничной пары из различных отрезков.

Формат входных данных

Первая строка входных данных содержит целое число n ($2 \leqslant n \leqslant 3 \cdot 10^5$) — количество элементарных частиц в последовательности.

Вторая строка входных данных содержит n целых чисел a_i ($1 \leqslant a_i \leqslant 3 \cdot 10^5$) — тип i-й элементарной частицы.

Формат выходных данных

Выведите одно целое число — максимальную возможную длину гармоничной пары из различных отрезков. Если такой пары не существует, требуется вместо этого вывести число -1.

Система оценки

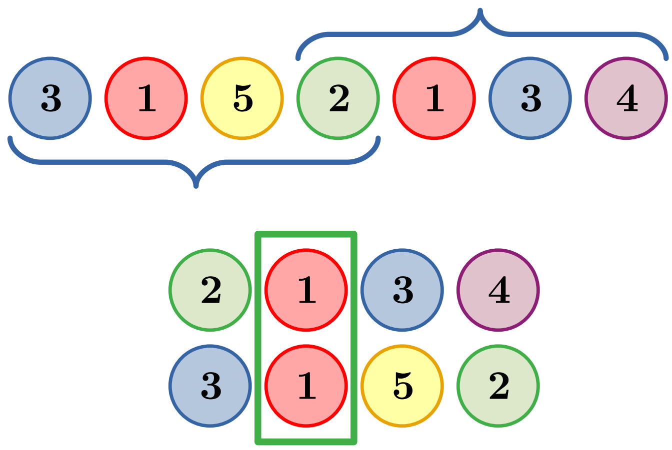
№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n \leqslant 200$	19	
2	$n \leqslant 3000$, $a_i \leqslant 2$	7	
3	$n \leqslant 3000$, $a_i \leqslant 100$	15	2
4	$n \leqslant 3000$	23	1 - 3
5	Нет дополнительных ограничений	36	1 - 4

Примеры

стандартный ввод	стандартный вывод
7 3 1 5 2 1 3 4	4
6 1 1 1 1 1 1	5
6 1 4 2 8 5 7	-1

Замечание

Первый пример изображен на рисунке ниже:



Как видно на рисунке, можно выбрать подотрезки (2 1 3 4) и (3 1 5 2), которые образуют гармоничную пару. Их длина равна 4, поэтому ответ равен 4.

Во втором примере можно взять два отрезка: первый — с $l = 1$ и $r = 5$, а второй — с $l = 2$ и $r = 6$. Как нетрудно видеть, эти отрезки образуют гармоничную пару и при этом считаются различными несмотря на то, что оба равны (1 1 1 1 1).

В третьем примере можно заметить, что гармоничный отрезок построить нельзя, и ответ равен -1.

Задача B. Оптимизация дорог

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	128 мегабайт

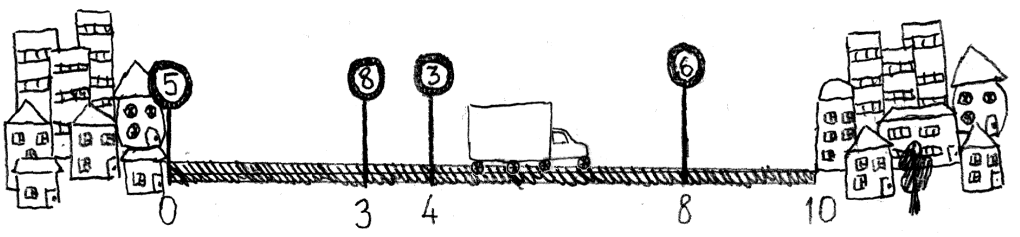
Правительство Марса заинтересовано не только в оптимизации космических перелетов, но и в улучшении дорожной системы планеты.

Одна из самых важных магистралей Марса соединяет Олимп-Сити и Кцолоп, столицу Кидонии. В рамках этой задачи будем рассматривать только путь от Кцолопа до Олимп-Сити, но не обратный путь (от Олимп-Сити до Кцолопа).

Дорога от Кцолопа до Олимп-Сити имеет длину ℓ километров. Каждая точка дороги имеет координату x ($0 \leqslant x \leqslant \ell$) — расстояние до Кцолопа в километрах. Таким образом, Кцолоп находится в точке с координатой 0, а Олимп-Сити — в точке с координатой ℓ.

На дороге стоит n дорожных знаков, на i-м из которых написано ограничение скорости a_i. Это ограничение означает, что следующий километр следует проехать за a_i минут и действует до тех пор, пока по пути не встретится следующий дорожный знак. Один из дорожных знаков стоит в самом начале дороги (т. е. в точке с координатой 0) и задает начальную скорость.

Зная расположение всех дорожных знаков, нетрудно вычислить время поездки от Кцолопа до Олимп-Сити. Рассмотрим пример:



В данном случае необходимо проехать первые три километра за пять минут каждый, затем — один километр за восемь минут, затем — еще четыре километра за три минуты каждый и, наконец, последние два километра за шесть минут каждый. Итого время в пути составит $3 \cdot 5 + 1 \cdot 8 + 4 \cdot 3 + 2 \cdot 6 = 47$ минут.

С целью оптимизации движения правительство Марса решило убрать не более k дорожных знаков. При этом знак в начале дороги убирать нельзя, иначе на начальном отрезке пути не будет никаких ограничений. Убирать знаки необходимо таким образом, чтобы минимизировать время в пути от Кцолопа до Олимп-Сити.

В Кидонии сосредоточены крупные промышленные предприятия Марса, поэтому оптимизация дороги до Олимп-Сити является приоритетной задачей. По этой причине правительство Марса поручило Вам решить эту задачу и выяснить, какие знаки необходимо убрать.

Формат входных данных

В первой строке входных данных находится два целых числа n, ℓ и k ($1 \leqslant n \leqslant 500$, $1 \leqslant \ell \leqslant 10^5$, $0 \leqslant k \leqslant n-1$) — количество знаков на дороге от Кцолопа до Олимп-Сити, расстояние между этими городами и максимальное количество знаков, которые разрешается убрать.

Во второй строке входных данных находится n целых чисел d_i ($d_1 = 0$, $d_i < d_{i+1}$, $0 \leqslant d_i \leqslant \ell - 1$) — координата i-го знака.

В третьей строке входных данных находится n целых чисел a_i ($1 \leqslant a_i \leqslant 10^4$) — ограничение скорости, записанное на i-м знаке.

Формат выходных данных

Выведите одно целое число — минимально возможное время в пути от Кцолопа до Олимп-Сити (в минутах), если убрать не более k дорожных знаков.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$k = 0$	4	
2	$k \leqslant 2$	5	1
3	$k \leqslant 3$	8	1 - 2
4	$k \leqslant 4$	10	1 - 3
5	$k = n - 1$	13	
6	$n \leqslant 20$	14	
7	$n \leqslant 80$	11	6
8	$n \leqslant 200$	17	6 - 7
9	Нет дополнительных ограничений	18	1 - 8

Примеры

стандартный ввод	стандартный вывод
4 10 0 0 3 4 8 5 8 3 6	47
4 10 2 0 3 4 8 5 8 3 6	38

Замечание

В первом примере знаки удалять нельзя, и ответ равен 47, как сказано в условии задачи выше.

Во втором примере следует удалить второй и четвертый знаки. Тогда придется проехать первые четыре километра за $4 \cdot 5 = 20$ минут, а последние шесть километров – за $6 \cdot 3 = 18$ минут. Итого получаем $4 \cdot 5 + 6 \cdot 3 = 38$ минут.

Задача C. Двоичные пауки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

На Марсе обитает необычный вид пауков — Двоичные пауки. Они плетут паутину, чтобы защищаться от врагов.

Чтобы сплести паутину, пауки объединяются в пары. При этом, если у первого паука в паре x лапок, а у второго — y лапок, то у них выходит паутина *прочностью* $x \oplus y$. Здесь $\oplus$ обозначает операцию побитового ИЛИ.

Двоичные пауки живут большими группами. Сейчас Вы наблюдаете за группой из n пауков, причем i-й паук имеет a_i лапок.

Когда группе пауков грозит опасность, то некоторые из них становятся *защитниками*. Защитники выбираются следующим образом. Во-первых, должно быть как минимум два паука-защитника. Во-вторых, любая пара из пауков-защитников должна уметь сплести паутину прочностью хотя бы k. В-третьих, защитников должно быть как можно больше.

Ученые долго исследовали поведение Двоичных пауков и выдвинули гипотезу, что они всегда могут выбрать защитников оптимальным образом, удовлетворяя при этом условиям выше. Вам предстоит проверить эту гипотезу на группе пауков. Для этого надо понять, сколько пауков должны стать защитниками. А поскольку Вы не являетесь Двоичным пауком, то Вы решили прибегнуть к помощи компьютера и написать программу, которая решает эту непростую задачу.

Формат входных данных

В первой строке входных данных находится два целых числа n и k ($2 \leqslant n \leqslant 3 \cdot 10^5$, $0 \leqslant k \leqslant 2^{30} - 1$) — количество пауков в группе и минимальная допустимая прочность паутины.

Во второй строке входных данных находится n целых чисел a_i ($0 \leqslant a_i \leqslant 2^{30} - 1$) — количество лапок у i-го паука.

Формат выходных данных

В первой строке выведите целое число ℓ ($2 \leqslant \ell \leqslant n$) — максимально возможное количество пауков-защитников.

Во второй строке выведите через пробел ℓ различных целых чисел b_i ($1 \leqslant b_i \leqslant n$) — номера пауков, которые станут защитниками.

Если существует несколько способов выбрать защитников, то выведите любой из них.

Увы, может получиться и так, что собрать защитников невозможно. В таком случае требуется вывести единственное число -1.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n \leqslant 20$	8	
2	$n \leqslant 300$	13	1
3	$n \leqslant 3000$	12	1 - 2
4	$a_i \leqslant 15$	5	
5	$a_i \leqslant 255$	9	4
6	$k = 2^m$ для некоторого m	24	
7	Нет дополнительных ограничений	29	1 - 6

Примеры

стандартный ввод	стандартный вывод
`6 8` `2 8 4 16 10 14`	`3` `1 4 5`
`6 1024` `1 2 3 1 4 0`	`-1`

Замечание

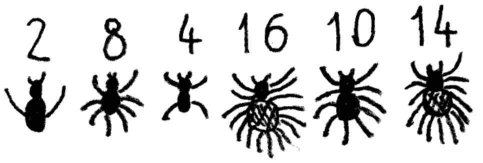
Исключающее ИЛИ — операция, которая на вход принимает два бита и возвращает бит 0, если биты на входе равны и 1 в противном случае. Например: $1 \oplus 0 = 1$, $1 \oplus 1 = 0$. Для применения побитового исключающего ИЛИ двух чисел эти числа сначала переводят в двоичную систему счисления, а затем применяют исключающее ИЛИ к каждому из разрядов. Например, $6 \oplus 3 = 5$, поскольку $6 = 110_2$, $3 = 11_2$. Применив исключающее ИЛИ поразрядно, получаем $5 = 101_2$:

$$\begin{array}{rr} & 110_2 \\ \oplus & 11_2 \\ \hline & 101_2 \end{array}$$

В языке программирования Pascal побитовое исключающее ИЛИ чисел a и b обозначается `a xor b`, а в языках Python и C++ — `a ^ b`.

Теперь рассмотрим пример из условия.

В первом примере группа пауков выглядит следующим образом:



Возьмем трех пауков: с двумя, десятью и 16-ю лапками. Тогда нетрудно видеть, что каждая пара сможет сплести достаточно прочную паутину, т. к. $2 \oplus 10 = 8 \geqslant 8$, $2 \oplus 16 = 18 \geqslant 8$ и $10 \oplus 16 = 26 \geqslant 8$.

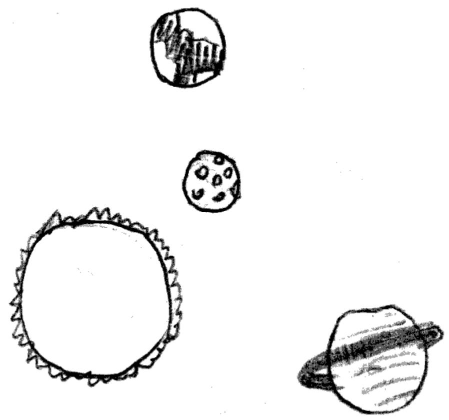
Данный вариант выбора не единственный: можно, к примеру, выбрать трех пауков с номерами 3, 4 и 6.

Во втором примере никакая пара пауков не сможет сплести паутину прочностью 1024 или больше, поэтому требуется вывести -1.

Задача D. Покорители Андромеды

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Мальчик Петя и его друг, робот Petya++, играют в компьютерную игру под названием «Покорители Андромеды». Суть игры заключается в исследовании Туманности Андромеды — одной из многочисленных галактик во Вселенной.



Ребята уже обосновались на одной из звездных систем в этой галактике. Звездная система состоит из n планет. Будем считать, что планеты пронумерованы целыми числами от 1 до n в порядке их удаления от звезды. Каждая планета имеет некоторую характеристику a_i, которая называется *привлекательность*.

В игре происходят события двух типов:

1. Петя изменяет привлекательность каждой из планет с номерами l до r включительно таким образом, что она становится равной x;

2. Petya++ облетает все планеты с номерами от l до r включительно и получает золото. Количество единиц золота, полученного Petya++, равно

$$1^2 \cdot a_l + 2^2 \cdot a_{l+1} + \cdots + (r-l)^2 \cdot a_{r-1} + (r-l+1)^2 \cdot a_r.$$

Petya++ хочет научиться быстро находить количество единиц золота, которое он получит, для каждого события второго типа. Но планет оказалось настолько много, что он просто не может посчитать ответы сам, даже несмотря на то, что Petya++ — робот. Поэтому он решил посоветоваться с Петей, как написать программу для решения этой задачи. Ребята долго думали, но так и не придумали быстрое решение, поэтому они обратились за помощью к Вам.

Поскольку планет много, то и количество единиц золота, собранного с них, может оказаться очень большим. Поэтому ребятам нужно знать лишь остаток от деления ответа на 10^9+7.

Формат входных данных

В первой строке входных данных находится два целых числа n и q ($1 \leqslant n \leqslant 5 \cdot 10^5$, $1 \leqslant q \leqslant 5 \cdot 10^5$) — количество планет в звездной системе и количество событий в игре.

Во второй строке входных данных находится n целых чисел a_i ($1 \leqslant a_i \leqslant 10^9$) — изначальная привлекательность каждой из планет.

В каждой из следующих q строк находится описание очередного события. Описание события состоит из нескольких чисел, разделенных пробелами и начинается с целого числа t ($t \in \{1, 2\}$) — тип события:

- если $t = 1$, то далее следуют три целых числа l, r и x ($1 \leqslant l \leqslant r \leqslant n$, $1 \leqslant x \leqslant 10^9$). Это означает, что Петя сделал привлекательность всех планет от l до r равной x;

- если $t = 2$, то далее следуют два целых числа l и r ($1 \leqslant l \leqslant r \leqslant n$). Это означает, что Petya++ хочет узнать, сколько золота он соберет, если облетит все планеты от l до r включительно.

Формат выходных данных

Для каждого события второго типа выведите по одному целому числу в отдельной строке — количество единиц золота, собранное Petya++, по модулю $10^9 + 7$.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n \leqslant 3000$, $q \leqslant 3000$	8	
2	$l = r$, если $t = 2$	9	
3	$a_i = 1$, $t = 2$	4	
4	$t = 2$	16	3
5	$l = r$, если $t = 1$	14	3 - 4
6	$n \leqslant 2 \cdot 10^5$, $q \leqslant 2 \cdot 10^5$	23	1
7	Нет дополнительных ограничений	26	1 - 6

Пример

стандартный ввод	стандартный вывод
8 7 3 8 2 4 6 1 3 2 1 3 2 2 5 1 1 3 7 2 2 5 1 3 4 5 2 1 3 2 1 1	53 148 167 80 7

Замечание

Рассмотрим пример из условия.

После первого события Petya++ соберет $1^2 \cdot 3 + 2^2 \cdot 8 + 3^2 \cdot 2 = 53$ единиц золота.

После второго события Petya++ соберет $1^2 \cdot 8 + 2^2 \cdot 2 + 3^2 \cdot 4 + 4^2 \cdot 6 = 148$ единиц золота.

После третьего события привлекательность планет станет равной (7 7 7 4 6 1 3).

После четвертого события Petya++ соберет $1^2 \cdot 7 + 2^2 \cdot 7 + 3^2 \cdot 4 + 4^2 \cdot 6 = 167$ единиц золота.

После пятого события привлекательность планет станет равной (7 7 5 5 6 1 3).

После шестого события Petya++ соберет $1^2 \cdot 7 + 2^2 \cdot 7 + 3^2 \cdot 5 = 80$ единиц золота.

Наконец, после седьмого события Petya++ соберет 7 единиц золота.