

Задача А. Меньше суммы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Дан массив целых чисел $A_1, A_2, \dots, A_N$. Необходимо ответить на Q запросов. Каждый запрос задается тремя числами L_i, R_i, X_i , и в нем нужно найти сумму элементов на подотрезке $A_{L_i}, A_{L_i+1}, \dots, A_{R_i}$, которые не превосходят X_i . При этом запросы даны в зашифрованном виде: на запрос даны числа $\alpha_i, \beta_i, \gamma_i$, и настоящие параметры запроса вычисляются по формулам:

- $L_i = \alpha_i \oplus B_{i-1}$
- $R_i = \beta_i \oplus B_{i-1}$
- $X_i = \gamma_i \oplus B_{i-1}$

Здесь B_i это ответ на запрос i , полагаем, что $B_0 = 0$, $\oplus$ это операция побитового исключающего ИЛИ (XOR).

Формат входных данных

Первая строка содержит целое число N — длина массива A ($1 \leq N \leq 2 \cdot 10^5$).

Вторая строка содержит N целых чисел $A_1, A_2, \dots, A_N$ ($0 \leq A_i \leq 10^9$).

Третья строка содержит целое число Q — количество запросов ($1 \leq Q \leq 2 \cdot 10^5$).

Следующие Q строк содержат по три целых числа $\alpha_i, \beta_i, \gamma_i$ ($0 \leq \alpha_i, \beta_i, \gamma_i \leq 10^{18}$). Гарантируется, что для расшифрованных запросов выполнено $1 \leq L_i \leq R_i \leq N, 0 \leq X_i \leq 10^9$.

Формат выходных данных

Для каждого запроса выведите одно целое число — ответ на запрос.

Пример

стандартный ввод	стандартный вывод
7	5
3 1 4 1 5 9 2	11
5	4
1 4 3	4
7 3 0	16
8 8 15	
5 3 6	
1 3 13	

Замечание

Расшифрованные запросы в примере

1. (1, 4, 3)
2. (2, 6, 5)
3. (3, 3, 4)
4. (1, 7, 2)
5. (5, 7, 9).

Задача В. Размен денег

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

За свою долгую жизнь Боря собрал коллекцию из n монет. Он выложил все эти монеты в ряд. При этом i -я в ряду монета имеет номинал a_i .

Боря собирается в очередное путешествие, но у него осталось очень мало времени на сборы. Поэтому он хочет взять некоторый отрезок лежащих подряд монет и надеется, что ему их хватит.

Боря хочет ответить на несколько запросов. В каждом запросе Боря хочет узнать, какую минимальную сумму он не сможет заплатить без сдачи, если он возьмет все монеты с l_i -й по r_i -ю. Более формально, он хочет найти такое минимальное натуральное число z , что нельзя выбрать подмножество монет с номерами от l_i до r_i , суммарный номинал которых равен z .

Формат входных данных

В первой строке задано два целых числа n и m ($1 \leq n, m \leq 150\,000$) — количество монет у Бори и количество запросов. В следующей строке задано n чисел a_i ($1 \leq a_i \leq 10^9$) — номинал i -й монеты.

В следующих m строках задано по два числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$) — описание запросов.

Формат выходных данных

На каждый из m запросов выведите минимальную сумму, которую нельзя заплатить без сдачи, воспользовавшись монетами с l_i -й по r_i -ю.

Пример

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

Задача С. Прямоугольники

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

В связи с большим количеством покупок дачных участков, два больших, но от этого не менее гордых государства (назовем их условно «первое» и «второе»), установили ряд соглашений, касающихся участков земли около их границы.

Чтобы лучше понять нововведения, рассмотрим границу между этими государствами на карте, которая висит на стене так, что север находится сверху. Введём ортонормированную систему координат, в которой ось OX направлена с запада на восток, а OY — с юга на север. Рассмотрим n равных по величине отрезков на оси OX , i -й из этих отрезков имеет координаты $(i-1, i)$. Каждому из них сопоставим вертикальную полосу, образованную всеми возможными прямыми, параллельными OY и проходящими через сам отрезок. Теперь, чтобы разделить государства, рассмотрим придуманную систему уровней, основанную на введённых вертикальных полосах. Для каждой полосы определим e_j уровень, который задается некоторым числом z_i . Точки, принадлежащие вертикальной полосе соответствующего отрезка, лежащие выше уровня, принадлежат первому государству, а ниже — второму.

Когда коренной житель одного из государств хочет купить прямоугольный участок земли со сторонами, параллельными осям координат (участки другого вида никого не интересуют), он может это сделать, если его родное государство доминирует на выбранном участке. Это происходит, если государство доминирует на большей, чем другое государство, части вертикальных полос, образованных отрезками на оси OX . Для вертикальных полос свойство преобладания определяется следующим образом: если площадь участка на этой полосе, принадлежащего одному из государств, строго больше площади, принадлежащей другому, то первое из них доминирует на этой полосе.

Вас просят написать программу, которая могла бы определять государство, доминирующее на участке, а также изменять границу между государствами.

Формат входных данных

В первой строке входного файла записано n — количество отрезков, на которые разделена ось OX ($1 \leq n \leq 5 \cdot 10^4$).

Во второй строке — n чисел z_i , определяющих границу между государствами ($0 \leq z_i \leq 10^9$).

В третьей строке задано m — число запросов к Вашей программе ($1 \leq m \leq 10^5$).

Далее следует m строк с запросами. Каждый запрос имеет вид « $C \ x \ z$ » или « $Q \ x_1 \ y_1 \ x_2 \ y_2$ ». Запрос вида « $C \ x \ z$ » означает, что уровень вертикальной полосы номер x стал равным z ($1 \leq x \leq n$, $1 \leq z \leq 10^9$). Запрос вида « $Q \ x_1 \ y_1 \ x_2 \ y_2$ » ($1 \leq x_1 < x_2 \leq n$, $0 \leq y_1 < y_2 \leq 10^9$) означает, что требуется вывести государство, доминирующее на участке, левой границей которого является вертикальная полоса номер x_1 (включительно), правой границей — вертикальная полоса номер x_2 (включительно), а с юга и с севера участок ограничен координатами y_1 и y_2 соответственно.

Все числа во входном файле целые.

Формат выходных данных

Для каждого запроса вида « $Q \ x_1 \ y_1 \ x_2 \ y_2$ » выведите «1», если на этом участке доминирует первое государство, «2», если второе, и «0», если ни у одного из государств преимущества нет.

Пример

стандартный ввод	стандартный вывод
2	1
0 0	1
5	0
Q 1 0 2 2	
C 1 1	
Q 1 0 2 2	
C 2 1	
Q 1 0 2 2	

Задача D. Снеговики

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	64 мегабайта

Зима. 2012 год. На фоне грядущего Апокалипсиса и конца света незамеченной прошла новость об очередном прорыве в областях клонирования и снеговиков: клонирования снеговиков. Вы конечно знаете, но мы вам напомним, что снеговик состоит из нуля или более вертикально поставленных друг на друга шаров, а клонирование — это процесс создания идентичной копии (клона).

В местечке Местячково учитель Андрей Сергеевич Учитель купил через интернет-магазин «Интернет-магазин аппаратов клонирования» аппарат для клонирования снеговиков. Теперь дети могут играть и даже играют во дворе в следующую игру. Время от времени один из них выбирает понравившегося снеговика, клонирует его и:

- либо добавляет ему сверху один шар;
- либо удаляет из него верхний шар (если снеговик не пустой).

Учитель Андрей Сергеевич Учитель записал последовательность действий и теперь хочет узнать суммарную массу всех построенных снеговиков.

Формат входных данных

Первая строка содержит количество действий n ($1 \leq n \leq 200\,000$). В строке номер $i + 1$ содержится описание действия i :

- $t\ m$ — клонировать снеговика номер t ($0 \leq t < i$) и добавить сверху шар массой m ($0 < m \leq 1000$);
- $t\ 0$ — клонировать снеговика номер t ($0 \leq t < i$) и удалить верхний шар. Гарантируется, что снеговик t не пустой.

В результате действия i , описанного в строке $i + 1$ создается снеговик номер i . Изначально имеется пустой снеговик с номером ноль.

Все числа во входном файле целые.

Формат выходных данных

Выведите суммарную массу построенных снеговиков.

Пример

стандартный ввод	стандартный вывод
8 0 1 1 5 2 4 3 2 4 3 5 0 6 6 1 0	74

Задача Е. Персистентный массив

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

Дан массив (вернее, первая, начальная его версия).

Нужно уметь отвечать на два запроса:

- $a_i[j] = x$ — создать из i -й версии новую, в которой j -й элемент равен x , а остальные элементы такие же, как в i -й версии.
- $\text{get } a_i[j]$ — сказать, чему равен j -й элемент в i -й версии.

Формат входных данных

Количество чисел в массиве N ($1 \leq N \leq 10^5$) и N элементов массива. Далее количество запросов M ($1 \leq M \leq 10^5$) и M запросов. Формат описания запросов можно посмотреть в примере. Если уже существует K версий, новая версия получает номер $K + 1$. И исходные, и новые элементы массива — целые числа от 0 до 10^9 . Элементы в массиве нумеруются числами от 1 до N .

Формат выходных данных

На каждый запрос типа `get` вывести соответствующий элемент нужного массива.

Примеры

стандартный ввод	стандартный вывод
6 1 2 3 4 5 6 11 create 1 6 10 create 2 5 8 create 1 5 30 get 1 6 get 1 5 get 2 6 get 2 5 get 3 6 get 3 5 get 4 6 get 4 5	6 5 10 5 10 8 6 30
1 1 13 get 1 1 create 1 1 10 get 1 1 get 2 1 create 1 1 20 get 1 1 get 2 1 get 3 1 create 2 1 30 get 1 1 get 2 1 get 3 1 get 4 1	1 1 10 1 10 20 1 10 20 30

Задача F. K -я порядковая статистика на отрезке

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 512 мегабайт

Дан массив из N неотрицательных чисел, строго меньших 10^9 . Вам необходимо ответить на несколько запросов о величине k -й порядковой статистики на отрезке $[l, r]$.

Формат входных данных

Первая строка содержит число N ($1 \leq N \leq 450\,000$) — размер массива.

Вторая строка может быть использована для генерации a_i — начальных значений элементов массива. Она содержит три числа a_1, l и m ($0 \leq a_1, l, m < 10^9$); для i от 2 до N

$$a_i = (a_{i-1} \cdot l + m) \bmod 10^9.$$

В частности, $0 \leq a_i < 10^9$.

Третья строка содержит одно целое число B ($1 \leq B \leq 1000$) — количество групп запросов.

Следующие B строк описывают одну группу запросов. Каждая группа запросов описывается 10 числами. Первое число G обозначает количество запросов в группе. Далее следуют числа x_1, l_x и m_x , затем y_1, l_y и m_y , затем, k_1, l_k и m_k ($1 \leq x_1 \leq y_1 \leq N$, $1 \leq k_1 \leq y_1 - x_1 + 1$, $0 \leq l_x, m_x, l_y, m_y, l_k, m_k < 10^9$). Эти числа используются для генерации вспомогательных последовательностей x_g и y_g , а также параметров запросов i_g, j_g и k_g ($1 \leq g \leq G$)

$$\begin{aligned}x_g &= ((i_{g-1} - 1) \cdot l_x + m_x) \bmod N + 1, & 2 \leq g \leq G \\y_g &= ((j_{g-1} - 1) \cdot l_y + m_y) \bmod N + 1, & 2 \leq g \leq G \\i_g &= \min(x_g, y_g), & 1 \leq g \leq G \\j_g &= \max(x_g, y_g), & 1 \leq g \leq G \\k_g &= (((k_{g-1} - 1) \cdot l_k + m_k) \bmod (j_g - i_g + 1)) + 1, & 2 \leq g \leq G\end{aligned}$$

Сгенерированные последовательности описывают запросы, g -й запрос состоит в поиске k_g -го по величине числа среди элементов отрезка $[i_g, j_g]$.

Суммарное количество запросов не превосходит 600 000.

Формат выходных данных

Выведите единственное число — сумму ответов на запросы.

Пример

стандартный ввод	стандартный вывод
5 1 1 1 5 1 1 0 0 3 0 0 2 0 0 1 2 0 0 5 0 0 3 0 0 1 1 0 0 5 0 0 5 0 0 1 3 0 0 3 0 0 1 0 0 1 1 0 0 4 0 0 1 0 0	15

Задача G. Разнообразные отрезки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 1024 мегабайта

Дан массив $a_0, a_1, \dots, a_{n-1}$. Необходимо обработать запросы, в каждом из которых требуется найти количество различных чисел на подотрезке $[l_i, r_i]$.

Запросы заданы в зашифрованной форме. Пусть p — ответ на предыдущий запрос (для первого запроса $p = 0$). Для i -го запроса даны числа x_i и y_i , а границы l_i и r_i вычисляются следующим образом:

- $l_i = \min((x_i + p) \bmod n, (y_i + p) \bmod n)$,
- $r_i = \max((x_i + p) \bmod n, (y_i + p) \bmod n)$.

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 5 \cdot 10^5$) — длина массива. Во второй строке содержатся n целых чисел $a_0, a_1, \dots, a_{n-1}$ ($0 \leq a_i \leq 10^9$).

В третьей строке дано число q ($1 \leq q \leq 5 \cdot 10^5$) — количество запросов. В каждой из следующих q строк заданы два числа x_i и y_i ($0 \leq x_i, y_i < n$).

Формат выходных данных

На каждый запрос выведите одно число — количество различных чисел на отрезке $[l_i, r_i]$.

Пример

стандартный ввод	стандартный вывод
5	3
0 1 1 0 2	1
5	1
0 4	2
3 3	3
0 1	
4 2	
0 2	

Задача Н. Отсутствующий на отрезках

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 мегабайта

Дан массив $a_0, a_1, \dots, a_{n-1}$. Необходимо обработать запросы, в каждом из которых требуется найти MEX чисел на подотрезке $[l_i, r_i]$.

Запросы заданы в зашифрованной форме. Пусть p — ответ на предыдущий запрос (для первого запроса $p = 0$). Для i -го запроса даны числа x_i и y_i , а границы l_i и r_i вычисляются следующим образом:

- $l_i = \min((x_i + p) \bmod n, (y_i + p) \bmod n)$,
- $r_i = \max((x_i + p) \bmod n, (y_i + p) \bmod n)$.

Напомним, что $\text{MEX}(S)$ набора чисел S это наименьшее целое неотрицательное число, которого нет в S . Так, например, $\text{MEX}(1, 2, 3) = 0$, а $\text{MEX}(0, 1, 2, 4) = 3$.

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 5 \cdot 10^5$) — длина массива. Во второй строке содержатся n целых чисел $a_0, a_1, \dots, a_{n-1}$ ($0 \leq a_i \leq 10^9$).

В третьей строке дано число q ($1 \leq q \leq 5 \cdot 10^5$) — количество запросов. В каждой из следующих q строк заданы два числа x_i и y_i ($0 \leq x_i, y_i < n$).

Формат выходных данных

На каждый запрос выведите одно число — MEX чисел на отрезке $[l_i, r_i]$.

Пример

стандартный ввод	стандартный вывод
5	3
0 1 1 0 2	0
5	1
0 4	3
3 4	2
3 4	
1 3	
3 0	

Замечание

Реальные отрезки запросов в первом примере это $[0, 4]$, $[1, 2]$, $[3, 4]$, $[2, 4]$, $[1, 3]$ и ответы на них это $\text{MEX}(0, 1, 1, 0, 2) = 3$, $\text{MEX}(1, 1) = 0$, $\text{MEX}(0, 2) = 1$, $\text{MEX}(1, 0, 2) = 3$, $\text{MEX}(1, 1, 0) = 2$

Задача I. Intercity Express

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Андрей разрабатывает систему для продажи железнодорожных билетов. Он собирается протестировать ее на Междугородней Экспресс линии, которая соединяет два больших города и имеет $n - 2$ промежуточных станций, то есть в итоге есть n станций, пронумерованных от 1 до n .

В Междугороднем Экспресс поезде есть s мест, пронумерованных с 1 до s . В тестирующем режиме система имеет доступ к базе данных, содержащей проданные билеты в направлении от станции 1 до станции n и должна отвечать на вопросы, можно ли продать билет от станции a до станции b , и если да, нужно найти минимальный номер места, которое свободно на протяжении всего пути между a и b .

Изначально система имеет только доступ на чтение, то есть даже если есть свободное место, она должна сообщить об этом, но не должна изменять данные.

Помогите Андрею протестировать его систему написанием программы, которые будет находить ответы на вопросы.

Формат входных данных

Первая строка содержит число n — количество станций, s — количество мест и m — количество уже проданных билетов ($2 \leq n \leq 10^9$, $1 \leq s \leq 100\,000$, $0 \leq m \leq 100\,000$).

В следующих m строках описаны билеты, описание каждого билета состоит из трех чисел: c_i , a_i и b_i — номер места, которое занимает владелец билета, номер станции, с которой продан билет и номер станции, до которой продан билет ($1 \leq c_i \leq s$, $1 \leq a_i < b_i \leq n$).

Следующая строка содержит число q — количество запросов ($1 \leq q \leq 100\,000$). Специальное значение p должно поддерживаться в течение считывания запросов. Изначально $p = 0$.

Следующие $2q$ целых чисел описывают запросы. Каждый запрос описывается двумя числами: x_i и y_i ($x_i < y_i$).

Чтобы получить города a и b между которыми нужно проверить наличие места, используется следующая формула:

$a = x_i + p$, $b = y_i + p$. Ответ на запрос — число 0, если нет места на каждом отрезке между a и b , или минимальный номер свободного места.

После ответа на запрос, надо приравнять число p полученному ответу на запрос.

Формат выходных данных

Для каждого запроса выведите ответ на него.

Пример

стандартный ввод	стандартный вывод
5 3 5	1
1 2 5	2
2 1 2	2
2 4 5	3
3 2 3	0
3 3 4	2
10	0
1 2	0
1 2	0
1 2	0
2 3	
-2 0	
2 4	
1 3	
1 4	
2 5	
1 5	

Замечание

Обратите внимание, что запросы выглядят так: (1, 2), (2, 3), (3, 4), (4, 5), (1, 3), (2, 4), (3, 5), (1, 4), (2, 5), (1, 5).

Задача J. Комбокамень

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Одна большая компания разрабатывает компьютерную игру «Комбокамень», которая должна мигом перевернуть всю индустрию. Правила игры достаточно сложные, и на реализацию серверного движка, моделирующего ход игры, был объявлен конкурс. От вас требуется реализовать подобный серверный движок.

Суть игры — игроки умеют призывать на арену и усиливать существ, используя заклинания, а также заставлять их сражаться друг с другом. Каждое существо имеет два параметра — численное значение атаки a и численное значение оставшегося здоровья h . Для краткости будем обозначать параметры существа как (a, h) . Исходно на арене нет существ.

Игроку доступны следующие заклинания:

- *Призыв существа*: Призвать новое существо с характеристиками $(1, 1)$. Если уже в игру было введено k существ, то новое существо получает номер $k + 1$.
- *Благословение силы*: Удвоить атаку выбранного существа. Если до применения этого заклинания оно имело характеристики (a, h) , то после этого действия оно будет иметь характеристики $(2a, h)$.
- *Божественный дух*: Удвоить здоровье выбранного существа. Если до применения этого заклинания оно имело характеристики (a, h) , то после этого действия оно будет иметь характеристики $(a, 2h)$.
- *Копия из лавы*: Призвать новое существо, которое будет иметь такие же характеристики, как и выбранное заклинанием существо. Если уже в игру было введено k существ, то новое существо получает номер $k + 1$.
- *Сражайся!*: Заставить двух различных существ сразиться. Во время сражения оба существа одновременно наносят друг другу по одному удару, уменьшая количество здоровья соперника на значение своей атаки. Так, если сражаются два существа с характеристиками (a_1, h_1) и (a_2, h_2) , то после сражения они будут иметь характеристики $(a_1, h_1 - a_2)$ и $(a_2, h_2 - a_1)$, соответственно. Если после сражения у существа остается 0 или меньше единиц здоровья, оно умирает и больше не может участвовать в игре.

От серверного движка, на реализацию которого объявлен конкурс, требуется способность про-моделировать все события и для каждого созданного во время игры существа вывести номер хода, на котором оно погибло, либо определить, что оно осталось живо к концу игры.

Кроме того, движок должен корректно обрабатывать случаи, когда игрок пытается взаимодействовать с мертвыми по мнению сервера существами: если заклинание *Благословение силы*, *Божественный дух* или *Сражайся!* обращено к уже мертвому существу, то не должно произойти ничего. Если заклинание *Копия из лавы* применено к мертвому существу, создается его мертвая копия с такими же характеристиками, но умершая на текущем ходу, в момент копирования.

Формат входных данных

В первой строке дано число n — количество совершенных ходов ($1 \leq n \leq 250\,000$).

В следующих n строках даны ходы, пришедшие к серверному движку, в следующем формате:

- 1 — применить заклинание *Призыв существа*;
- 2 i — применить заклинание *Благословение силы* к существу с номером i ;
- 3 i — применить заклинание *Божественный дух* к существу с номером i ;
- 4 i — применить заклинание *Копия из лавы* к существу с номером i ;
- 5 $i\ j$ — применить заклинание *Сражайся!* к существам с номерами i и j .

Гарантируется, что любые упомянутые в запросах существа к моменту запроса уже были призваны, но, возможно, могут уже быть мертвы.

Формат выходных данных

В первой строке выведите одно целое число k — количество существ, призванных за время игры.

В следующей строке выведите k целых чисел $t_1, t_2, \dots, t_k$ — если существо с номером i осталось живо к концу игры, то t_i должно быть равно -1 , иначе t_i должно быть равно номеру хода, на котором оно погибло.

Пример

стандартный ввод	стандартный вывод
16	5
1	13 5 14 -1 16
2 1	
3 1	
1	
5 1 2	
3 1	
1	
3 3	
3 3	
4 1	
5 1 3	
3 3	
5 1 3	
5 4 3	
5 4 3	
4 1	

Замечание

В таблице можно увидеть, как изменялись характеристики существ в первом примере.

ход	1	2	3	4	5
0	-	-	-	-	-
1	(1, 1)	-	-	-	-
2	(2, 1)	-	-	-	-
3	(2, 2)	-	-	-	-
4	(2, 2)	(1, 1)	-	-	-
5	(2, 1)	мертво	-	-	-
6	(2, 2)	мертво	-	-	-
7	(2, 2)	мертво	(1, 1)	-	-
8	(2, 2)	мертво	(1, 2)	-	-
9	(2, 2)	мертво	(1, 4)	-	-
10	(2, 2)	мертво	(1, 4)	(2, 2)	-
11	(2, 1)	мертво	(1, 2)	(2, 2)	-
12	(2, 1)	мертво	(1, 4)	(2, 2)	-
13	мертво	мертво	(1, 2)	(2, 2)	-
14	мертво	мертво	мертво	(2, 1)	-
15	мертво	мертво	мертво	(2, 1)	-
16	мертво	мертво	мертво	(2, 1)	мертво

Задача К. Операции с таблицей

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам дана таблица $n \times m$, в каждой клетке которой записано число 0. Вам требуется ответить на q запросов трех типов:

- 1 $l\ r\ x$. Добавить число x ко всем числам, лежащим в столбцах $l, l+1, \dots, r$;
- 2 $i\ x$. Заменить все элементы в i -й строке на x ;
- 3 $i\ j$. Вывести значение элемента из i -й строки, j -го столбца.

Формат входных данных

В первой строке указана тройка n, m и q ($1 \leq n, m, q \leq 200\,000$) — размеры таблицы и количество запросов.

Затем следуют описания запросов в следующем формате:

- 1 $p\ q\ y$. ($1 \leq p, q \leq m, 1 \leq y \leq 10^9$)
- 2 $s\ y$. ($1 \leq i \leq n, 1 \leq y \leq 10^9$)
- 3 $s\ c$. ($1 \leq i \leq n, 1 \leq j \leq m$)

Обратите внимание, что запросы зашифрованы! Чтобы их расшифровать, требуется поддерживать переменную α — ответ на последний запрос третьего типа, изначально $\alpha = 0$.

Значения x, i, j, l, r для всех запросов восстанавливаются по формуле:

$$x = 1 + ((\alpha + y) \bmod 10^9)$$

$$i = 1 + ((\alpha + s) \bmod n)$$

$$j = 1 + ((\alpha + c) \bmod m)$$

$$l = 1 + ((\alpha + p) \bmod m)$$

$$r = 1 + ((\alpha + q) \bmod m)$$

Гарантируется, что $l \leq r$ для всех запросов первого типа.

Формат выходных данных

Для каждого запроса третьего типа выведите ответ на него в отдельной строке.

Пример

стандартный ввод	стандартный вывод
3 3 9	1
1 3 1 1000000000	2
3 1 1	2
2 1 1000000000	5
3 1 1	3
3 3 1	4
1 2 3 1000000000	
3 3 2	
3 2 3	
3 3 1	

Задача L. King and ICPC

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Тёмные времена нависли над АСМ-ом. Тысячи команд АСМ-щиков вышли на защиту от нападения СТФ-щиков. Всего имеется N команд, пронумерованных от 1 до N . Каждая команда, по традиции, состоит из трёх человек. Каждому человеку присвоили параметр АСМ-овости, выраженный целым числом.

Король АСМ-щиков хочет создать армию, выбрав из каждой команды ровно по одному человеку так, чтобы суммарная АСМ-овость была как можно больше, но при этом делилась нацело на число D .

Так как АСМ-щики по своей натуре любят усложнять задачи, они решили рассмотреть несколько отрезков $[l_i, r_i]$, составляя армию лишь из команд с номерами из этого отрезка.

Король поручил вам обработку этих запросов. На горизонте уже видны флаги, у вас осталось меньше пяти часов!

Формат входных данных

Первая строка содержит два целых числа N ($1 \leq N \leq 50\,000$) и D ($1 \leq D \leq 50$).

В следующих N строках содержится описание каждой команды в виде трёх целых чисел. АСМ-овость каждого человека является целым числом от 0 до 10^9 .

Следующая строка содержит целое число M — количество отрезков ($1 \leq M \leq 300\,000$).

В следующих M строках идёт описание каждого запроса в виде пары чисел l_i и r_i ($1 \leq l_i \leq r_i \leq N$).

Формат выходных данных

На каждый отрезок выведите в отдельной строке максимальную сумму, которую можно на нём собрать. Если же сумму, делящуюся на D собрать невозможно, выведите «-1» (без кавычек).

Пример

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

Задача М. Настрой в ушах

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4.5 секунд
Ограничение по памяти:	512 мегабайт

С распространением музыкальных стриминговых сервисов сильно изменилось то, как слушатели потребляют музыку. Пользователям доступны огромные каталоги треков, и меломан Илья собрал в своём аккаунте целых N любимых композиций, пронумерованных от 1 до N .

Сейчас у Ильи особенное настроение, и для каждого трека i он знает число A_i *соответствие* этого трека состоянию Ильи. Все значения A_i целые и лежат в диапазоне от 1 до K .

Когда Илья выбирает отрезок треков $[l, r]$ и начинает их слушать, он может пропускать некоторые из них. Для хорошего опыта важно, чтобы последовательность значений A_i у прослушанных треков не убывала. Найдите число способов прослушать треки так, чтобы опыт прослушивания получился хорошим.

Иными словами, от вас требуется найти количество способов выбрать непустую подпоследовательность треков на отрезке $[l, r]$, такую что их соответствия A_i образуют неубывающую последовательность. Поскольку ответ может быть очень большим, выведите его по модулю 998 244 353.

Формат входных данных

В первой строке заданы два целых числа N и K ($1 \leq N \leq 2 \cdot 10^5$, $1 \leq K \leq 100$) — количество треков и диапазон возможных значений.

Во второй строке содержатся N целых чисел $A_1, A_2, \dots, A_N$ ($1 \leq A_i \leq K$) — соответствия треков настроению.

В третьей строке дано целое число Q ($1 \leq Q \leq 4 \cdot 10^5$) — количество запросов.

В каждой из следующих Q строк даны два целых числа l и r ($1 \leq l \leq r \leq N$) — границы отрезка запроса.

Формат выходных данных

Для каждого запроса выведите одно число — количество неубывающих непустых подпоследовательностей на отрезке $[l, r]$ по модулю 998 244 353.

Примеры

стандартный ввод	стандартный вывод
5 3 2 1 2 3 3 4 1 3 2 4 1 5 3 3	5 7 23 1
10 5 4 2 3 3 3 4 2 1 2 1 8 7 8 8 9 10 10 6 9 9 10 1 7 5 6 9 9	2 3 1 6 2 35 3 1

Замечание

В первом примере неубывающие последовательности это

1. В первом запросе: $(2), (1), (2), (2, 2), (1, 2)$.
2. Во втором запросе: $(1), (2), (3), (1, 2), (1, 3), (2, 3), (1, 2, 3)$.
3. В четвёртом запросе:
 $(2), (1), (2), (3), (3), (2, 2), (2, 3), (2, 3), (1, 2), (1, 3),$
 $(1, 3), (2, 3), (2, 3), (3, 3), (2, 2, 3), (2, 2, 3), (2, 3, 3),$
 $(1, 2, 3), (1, 2, 3), (1, 3, 3), (2, 3, 3), (2, 2, 3, 3), (1, 2, 3, 3)$
4. В пятом запросе: (2) .

Задача N. Изменяемое разнообразие

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 512 мегабайт

Дан массив $A_1, A_2, \dots, A_N$. Необходимо обработать запросы двух видов:

- $1 \ l \ r$ — найти количество различных чисел среди $A_l, A_{l+1} \dots A_r$;
- $2 \ i \ x$ — обновить значение $A_i = x$.

Формат входных данных

В первой строке дано целое число N ($1 \leq N \leq 5 \cdot 10^4$) — длина массива. Во второй строке содержатся N целых чисел $A_1, A_2, \dots, A_N$ ($0 \leq A_i \leq 10^9$).

В третьей строке дано число q ($1 \leq q \leq 5 \cdot 10^4$) — количество запросов. В каждой из следующих q строк заданы запросы в одном из двух форматов:

- $1 \ l \ r$ ($1 \leq l \leq r \leq n$);
- $2 \ i \ x$ ($1 \leq i \leq n, 0 \leq x \leq 10^9$).

Формат выходных данных

В ответ на каждый запрос первого типа выведите одно число — количество различных чисел на отрезке.

Пример

стандартный ввод	стандартный вывод
5	3
0 1 1 0 2	2
8	3
1 1 5	2
1 2 4	4
1 2 5	1
2 4 2	
1 2 5	
2 2 3	
1 1 5	
1 4 4	