Задача A. Разреженные таблицы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Дан массив из n чисел. Требуется написать программу, которая будет отвечать на запросы следующего вида: найти минимум на отрезке между u и v включительно.

Формат входных данных

В первой строке входного файла даны три натуральных числа n, m ($1 \leqslant n \leqslant 10^5$, $1 \leqslant m \leqslant 10^7$) и a_1 ($0 \leqslant a_1 < 16\,714\,589$) — количество элементов в массиве, количество запросов и первый элемент массива соответственно. Вторая строка содержит два натуральных числа u_1 и v_1 ($1 \leqslant u_1, v_1 \leqslant n$) — первый запрос.

Элементы $a_2, a_3, \ldots, a_n$ задаются следующей формулой:

$$a_{i+1} = (23 \cdot a_i + 21563) \bmod 16714589.$$

Например, при $n = 10$, $a_1 = 12345$ получается следующий массив: $a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095)$.

Запросы генерируются следующим образом:

$$u_{i+1} = \big((17 \cdot u_i + 751 + ans_i + 2i) \bmod n\big) + 1,$$
$$v_{i+1} = \big((13 \cdot v_i + 593 + ans_i + 5i) \bmod n\big) + 1,$$

где ans_i — ответ на запрос номер i.

Обратите внимание, что u_i может быть больше, чем v_i.

Формат выходных данных

В выходной файл выведите u_m, v_m и ans_m (последний запрос и ответ на него).

Пример

стандартный ввод	стандартный вывод
10 8 12345 3 9	5 3 1565158

Задача B. Прямоугольники

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

Когда-то тут была легенда про вёдра, но её съели.

Жюри олимпиады

Есть таблица T размера $N \times M$. Элементами таблицы являются прямоугольники T_{ij}, где $0 \leqslant i < N$ и $0 \leqslant j < M$. Прямоугольник T_{ij} задаётся четвёркой чисел $(x_1^{ij},\ y_1^{ij},\ x_2^{ij},\ y_2^{ij})$, где (x_1^{ij}, y_1^{ij}) и (x_2^{ij}, y_2^{ij}) — координаты противоположных углов прямоугльника. Стороны прямоугольника параллельны осям координат.

Далее вам поступают запросы. Каждый запрос состоит из четырёх чисел: (r_1, c_1, r_2, c_2). Ответом на такой запрос является площадь фигуры, являющейся пересечением всех прямоугольников T_{ij} таких, что $\min(r_1, r_2) \leqslant i \leqslant \max(r_1, r_2)$ и $\min(c_1, c_2) \leqslant j \leqslant \max(c_1, c_2)$. Запросов очень много, поэтому мы просим вас вывести сумму ответов на все запросы по модулю $10^9 + 7$.

Формат входных данных

В первой строке записаны два целых числа N и M — размеры таблицы T ($1 \leqslant N, M \leqslant 127$). Далее в N строках описывается таблица T: в $(i+1)$-й строке $(j+1)$-я четвёрка чисел $x_1^{ij}\ y_1^{ij}\ x_2^{ij}\ y_2^{ij}$ описывает прямоугольник T_{ij}. Гаранируется, что $|x_k^{ij}|, |y_k^{ij}| \leqslant 10^6$.

Дальше в отдельной строке записано четыре числа. Первое из них, число Q — количество запросов ($1 \leqslant Q \leqslant 5 \cdot 10^6$). Следующие три числа — это A, B, v_0 ($0 \leqslant A, B, v_0 < 10^9 + 7$). При помощи этих чисел генерируется бесконечная последовательность $\{v_i\}$ по правилу $v_i = (A \cdot v_{i-1} + B) \bmod (10^9 + 7)$.

После этого k-й запрос (запросы нумеруются с единицы) задаётся следующей четвёркой чисел: $(v_{4k-3} \bmod N, v_{4k-2} \bmod M, v_{4k-1} \bmod N, v_{4k} \bmod M)$.

Формат выходных данных

Выведите сумму ответов на все запросы по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
2 2 0 0 2 2 1 1 3 3 0 3 2 1 1 2 3 0 1 500000003 4 2	1
3 2 8 -1 -7 6 6 8 9 10 -4 -10 4 9 -3 -8 6 9 -2 -9 3 8 -5 7 7 3 5 303164476 273973578 65779139	85

Замечание

В первом примере запрос имеет вид `(1,0,0,1)`, то есть это запрос ко всей таблице. Пересечением всех прямоугольников является квадрат с углами в точках $(1, 1)$ и $(2, 2)$. Его площадь равна 1.

Во втором примере запросы имеют вид `(0,1,1,1)`, `(1,0,2,0)`, `(0,0,2,1)`, `(0,1,1,1)`, `(0,1,0,0)`. На второй запрос ответ — 85, на остальные — 0.

Задача C. Звезды

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

Вася любит наблюдать за звездами. Но следить за всем небом сразу ему тяжело. Поэтому он наблюдает только за частью пространства, ограниченной кубом размером $n \times n \times n$. Этот куб поделен на маленькие кубики размером $1 \times 1 \times 1$. Во время его наблюдений могут происходить следующие события:

1. В каком-то кубике появляются или исчезают несколько звезд.
2. К нему может заглянуть его друг Петя и поинтересоваться, сколько видно звезд в части пространства, состоящей из нескольких кубиков.

Формат входных данных

Первая строка входного файла содержит натуральное число $1 \leqslant n \leqslant 128$. Координаты кубиков — целые числа от 0 до $n - 1$. Далее следуют записи о происходивших событиях по одной в строке. В начале строки записано число m. Если m равно:

1, то за ним следуют 4 числа — x, y, z ($0 \leqslant x, y, z < N$) и k ($-20000 \leqslant k \leqslant 20000$) — координаты кубика и величина, на которую в нем изменилось количество видимых звезд;

2, то за ним следуют 6 чисел — x_1, y_1, z_1, x_2, y_2, z_2 ($0 \leqslant x_1 \leqslant x_2 < N$, $0 \leqslant y_1 \leqslant y_2 < N$, $0 \leqslant z_1 \leqslant z_2 < N$), которые означают, что Петя попросил подсчитать количество звезд в кубиках (x, y, z) из области: $x_1 \leqslant x \leqslant x_2$, $y_1 \leqslant y \leqslant y_2$, $z_1 \leqslant z \leqslant z_2$;

3, то это означает, что Васе надоело наблюдать за звездами и отвечать на вопросы Пети. Эта запись встречается во входном файле только один раз и будет последней.

Количество записей во входном файле не больше 100 002.

Формат выходных данных

Для каждого Петиного вопроса выведите искомое количество звезд.

Пример

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

Задача D. Катый ноль

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

Реализуйте эффективную структуру данных, позволяющую изменять элементы массива и вычислять индекс k-го слева нуля на данном отрезке в массиве.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leqslant N \leqslant 200\,000$) — количество чисел в массиве. Во второй строке вводятся N чисел от 0 до 100 000 — элементы массива. В третьей строке вводится одно натуральное число M ($1 \leqslant M \leqslant 200\,000$) — количество запросов. Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (`s` — вычислить индекс k-го нуля, `u` — обновить значение элемента). Следом за `s` вводится три числа — левый и правый концы отрезка и число k ($1 \leqslant k \leqslant N$). Следом за `u` вводятся два числа — номер элемента и его новое значение.

Формат выходных данных

Для каждого запроса s выведите результат. Все числа выводите в одну строку через пробел. Если нужного числа нулей на запрашиваемом отрезке нет, выводите -1 для данного запроса.

Пример

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

Задача E. Перестановки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.75 секунд
Ограничение по памяти:	512 мегабайт

Вася выписал на доске в каком-то порядке все числа от 1 по N, каждое число ровно по одному разу. Количество чисел оказалось довольно большим, поэтому Вася не может окинуть взглядом все числа. Однако ему надо всё-таки представлять эту последовательность, поэтому он написал программу, которая отвечает на вопрос — сколько среди чисел, стоящих на позициях с x по y, по величине лежат в интервале от k до l. Сделайте то же самое.

Формат входных данных

В первой строке лежит два натуральных числа — $1 \leqslant N \leqslant 10^5$ — количество чисел, которые выписал Вася и $1 \leqslant M \leqslant 10^5$ — количество вопросов, которые Вася хочет задать программе. Во второй строке дано N чисел — последовательность чисел, выписанных Васей. Далее в M строках находятся описания вопросов. Каждая строка содержит четыре целых числа $1 \leqslant x \leqslant y \leqslant N$ и $1 \leqslant k \leqslant l \leqslant N$.

Формат выходных данных

Выведите M строк, каждая должна содержать единственное число — ответ на Васин вопрос.

Пример

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

Замечание

Напишите merge-sort-tree. Другие решения будут забанены.

Задача F. После карантина

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

И вот, наконец, оно — первое очное занятие параллели A'! Ребята поднимаются на пятый этаж и дружно заходят в аудиторию. Но не тут-то было — из-за противовирусных мер в аудитории теперь специальная рассадка. А именно — в каждом ряду ровно один стул, а каждый ученик заранее знает свой номер ряда A_i.

Дети заходят в кабинет по очереди, в порядке очереди садятся на свои места и достают ноутбуки.

Как вы все знаете, основное мерило крутости программиста — количество наклеек на ноутбуке. Поэтому каждый раз, когда школьник по пути до своего места проходит мимо человека, у которого больше стикеров на ноуте, он расстраивается.

Посчитайте для каждого школьника, сколько раз он расстроится.

Формат входных данных

Обратите внимание — в этой задаче несколько наборов тестовых данных, не более 5. Каждый набор начинается с числа учеников n ($1 \leqslant n \leqslant 10^5$). После этого в следующих n строках идут числа A_i, B_i ($1 \leqslant A_i \leqslant n$, $1 \leqslant B_i \leqslant 10^9$) — номер места i-го школьника и количество наклеек на его ноутбуке. Гарантируется, что A_i образуют перестановку.

Формат выходных данных

Для каждого набора тестовых данных в отдельной строке выведите n чисел $p_1 \dots p_i \dots p_n$ — число огорчений i-го школьника.

Пример

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

Задача G. И снова сумма...

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте структуру данных, которая поддерживает множество S целых чисел, с котором разрешается производить следующие операции:

- $add(i)$ — добавить в множество S число i (если он там уже есть, то множество не меняется);
- $sum(l, r)$ — вывести сумму всех элементов x из S, которые удовлетворяют неравенству $l \leqslant x \leqslant r$.

Формат входных данных

Исходно множество S пусто. Первая строка входного файла содержит n — количество операций ($1 \leqslant n \leqslant 300\,000$). Следующие n строк содержат операции. Каждая операция имеет вид либо «`+` i», либо «`?` l r». Операция «`?` l r» задает запрос $sum(l, r)$.

Если операция «`+` i» идет во входном файле в начале или после другой операции «`+`», то она задает операцию $add(i)$. Если же она идет после запроса «`?`», и результат этого запроса был y, то выполняется операция $add((i + y) \bmod 10^9)$.

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до 10^9.

Формат выходных данных

Для каждого запроса выведите одно число — ответ на запрос.

Пример

стандартный ввод	стандартный вывод
6 + 1 + 3 + 3 ? 2 4 + 1 ? 2 4	3 7

Задача H. Переворот

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Дан массив. Надо научиться обрабатывать два типа запросов.

- `1 L R` - перевернуть отрезок [L, R]
- `2 L R` - найти минимум на отрезке [L, R]

Формат входных данных

Первая строка файла содержит два числа n, m. ($1 \leqslant n, m \leqslant 10^5$) Во второй строке находится n чисел a_i ($1 \leqslant a_i \leqslant 10^9$)- исходный массив. Остальные m строк содержат запросы, в формате описанном в условии. Для чисел L,R выполняется ограничение ($1 \leqslant L \leqslant R \leqslant n$).

Формат выходных данных

На каждый запрос типа 2, во входной файл выведите ответ на него, в отдельной строке.

Пример

стандартный ввод	стандартный вывод
10 7 5 3 2 3 12 6 7 5 10 12 2 4 9 1 4 6 2 1 8 1 1 8 1 8 9 2 1 7 2 3 6	3 2 2 2

Задача I. Логирование

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

Лиза пишет программу для анализа логов дистрибутивных компьютерных систем. Когда узел становится онлайн, это может протолкнуть партию событий логов в прошлом. И наоборот, когда он переходит в автономный режим некоторые записи логов могут исчезнуть.

Чтобы обеспечить стабильность и доступность приложения Лизе необходимо контролировать число различных событий в сегментах лога. Она будет разбираться с распределительной частью, в то время как вы должны реализовать локальный.

Изначально список логов пуст, и вы должны поддерживать следующие операции:

- **insert** `<index>` `<number>` `<type>` — добавить `<number>` событий типа `<type>` перед событием с индексом `<index>`. Все события, у которых индекс больше или равен `<index>` нумеруются заново.
- **remove** `<index>` `<number>` — удалить `<number>` элементов, начиная с элемента под индексом `<index>`.
- **query** `<index_1>` `<index_2>` — вывести количество различных типов событий на отрезке с `<index_1>` до `<index_2>` включительно.

События нумеруются с 1. Тип событий представляется одним латинским символом.

Формат входных данных

В первой строке входного файла содержится единственное целое число n — количество операций ($1 \leqslant n \leqslant 30\,000$). Следующие по n строк содержат описание запросов.

Описание операции начинается с типа операции: '`+`' для добавления, '`-`' для удаления и '`?`' для запроса. Дальше следует аргументы запроса, описанные в условиях выше.

Все запросы валидны, элементы с такими индексами существуют, нет запросов на удаление несуществующих элементов.

Количество запросов добавления и удаления не превышает 10 000.

Типы событий представлены в виде строчной буквы латинского алфавита.

Формат выходных данных

Для каждого запроса **query** выведите одно целое число — количество различных типов событий на отрезке `<index_1>`, `<index_2>` включительно.

Пример

стандартный ввод	стандартный вывод
8 + 1 4 w + 3 3 o ? 2 3 - 2 2 ? 2 3 + 2 2 t ? 1 6 - 1 6	2 1 3

Замечание

Пояснение к примеру:

1. `wwww`

2. `wwoooww`
3. `w[wo]ooww` : 2 различных события
4. `wooww`
5. `w[oo]ww` : 1 событие
6. `wttooww`
7. `[wttoow]w` : 3 различных события
8. `w`

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

В следующих m строках задано по два числа l_i и r_i ($1 \leqslant l_i \leqslant r_i \leqslant n$) — 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 2 1 5 3 1 1 5 1 3 1 1 2 4 2 5	13 4 1 2 11

Задача K. Автоматизация склада

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

Компания занимается автоматизацией склада. На складе хранятся n видов товаров, пронумерованных от 1 до n, каждый вид товара хранится в своём помещении. Товар вида i хранится в помещении с номером i.

Специальный робот обслуживает запросы по получению товаров со склада. Для доступа в помещения склада робот использует специальные электронные карты. Карты у робота хранятся в специальном отсеке, из которого он может вынуть верхнюю карту. Вынутую карту робот может вернуть в отсек на любое место: на верхнюю позицию, между любыми двумя картами или на самую нижнюю позицию.

Чтобы открыть помещение, робот действует следующим образом. Он вынимает карты из отсека для их хранения и возвращает их обратно в отсек, пока на верхней позиции не окажется карта от помещения, которое ему необходимо открыть. После этого, вынув эту карту, робот использует её, чтобы открыть помещение, и затем также возвращает в отсек для хранения карт. Если суммарно роботу потребовалось вынуть из отсека x карт, включая ту, которой он в итоге открыл помещение, будем говорить, что для открытия помещения робот совершил x действий.

В начале рабочего дня роботу поступил заказ на выдачу m товаров: $a_1, a_2, \ldots, a_m$. Робот должен выдать товары именно в этом порядке. Для этого он последовательно выполняет следующие действия: открывает помещение, в котором лежит очередной товар, берет товар, закрывает помещение и выдаёт товар клиенту. После этого робот переходит к выдаче следующего товара.

Исходно электронные карты лежат в отсеке в следующем порядке, от верхней к нижней: $b_1, b_2, \ldots, b_n$. Для каждого помещения в отсеке лежит ровно одна карта.

Время выдачи товаров со склада зависит от того, сколько раз суммарно роботу придётся вынимать верхнюю карту из отсека для их хранения, чтобы найти карту от очередного помещения. Необходимо таким образом выбрать места, куда робот должен возвращать вынутые карты, чтобы минимизировать суммарное количество действий робота для открытия помещений.

Требуется написать программу, которая по заданным целым числам n и m, последовательности выдаваемых товаров $a_1, a_2, \ldots, a_m$ и начальному положению карт в отсеке для хранения $b_1, b_2, \ldots, b_n$ определяет, какое минимальное количество действий придется совершить роботу, чтобы открыть все помещения в необходимом порядке. Для каждой вынутой карты необходимо также указать позицию, на которую её необходимо вернуть, чтобы добиться оптимального количества действий.

Формат входных данных

Первая строка входных данных содержит два целых числа n и m ($1 \leqslant n, m \leqslant 3 \cdot 10^5$) — количество видов товаров и количество товаров, которые необходимо выдать со склада.

Вторая строка содержит m целых чисел $a_1, a_2, \ldots, a_m$ ($1 \leqslant a_i \leqslant n$) — типы товаров, которые необходимо выдать со склада, перечисленные в том порядке, в котором это необходимо сделать.

Третья строка содержит n различных целых чисел $b_1, b_2, \ldots, b_n$ ($1 \leqslant b_i \leqslant n$) — порядок, в котором карты исходно находятся в отсеке для их хранения, перечисленные от верхней к нижней.

Формат выходных данных

Первая строка должна содержать число k — минимальное количество действий, которое потребуется совершить роботу, чтобы выдать товары в заданном порядке.

Далее выведите k чисел. Для каждого действия робота выведите одно число: позицию, на которую ему следует вернуть вынутую карту в отсек для хранения. Если карта возвращается на самую верхнюю позицию, следует вывести 1, если после одной карты, 2, и так далее, для последней позиции следует вывести n.

Если существует несколько способов минимизировать суммарное число действий, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
1 1 1 1	1 1
4 5 4 1 2 4 4 4 3 2 1	7 4 4 2 4 4 1 4
2 2 1 2 2 1	3 2 2 2

Задача L. Магазин

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

Наташа уже хотела полететь на Землю, как вспомнила, что ей нужно пойти в марсианский магазин, чтобы купить друзьям марсианские сувениры.

Известно, что марсианский год длится x_{max} месяцев, месяц — y_{max} дней, сутки — z_{max} секунд. Ещё Наташа знает, что этот магазин работает по такому расписанию: выбираются 2 месяца в году: x_l и x_r ($1 \leqslant x_l \leqslant x_r \leqslant x_{max}$), 2 дня в месяце: y_l и y_r ($1 \leqslant y_l \leqslant y_r \leqslant y_{max}$) и 2 секунды в сутках: z_l и z_r ($1 \leqslant z_l \leqslant z_r \leqslant z_{max}$). Магазин работает во все такие моменты времени (месяц x, день y, секунда z), когда одновременно $x_l \leqslant x \leqslant x_r$, $y_l \leqslant y \leqslant y_r$ и $z_l \leqslant z \leqslant z_r$.

К сожалению, Наташе неизвестны числа $x_l, x_r, y_l, y_r, z_l, z_r$.

Один марсианин сказал Наташе: "Я ходил в этот магазин $(n + m)$ раз. Из них n раз он был открыт, а m раз — закрыт."Также он рассказал про каждый свой поход в магазин: месяц, день, секунду этого похода и был ли магазин в этот момент открыт или закрыт.

Наташа может пойти в магазин k раз. Для каждого из них определите, будет ли магазин в момент похода в него открыт, закрыт или эта информация неизвестна.

Формат входных данных

Первая строка содержит 6 целых чисел x_{max}, y_{max}, z_{max}, n, m, k ($1 \leqslant x_{max}, y_{max}, z_{max} \leqslant 10^5$, $1 \leqslant n \leqslant 10^5$, $0 \leqslant m \leqslant 10^5$, $1 \leqslant k \leqslant 10^5$) — количество месяцев в году, дней в месяце, секунд в сутках, раз, когда магазин (по словам марсианина) был открыт, когда он был закрыт, запросов Наташи.

i-я из следующих n строк содержит 3 целых числа x_i, y_i, z_i ($1 \leqslant x_i \leqslant x_{max}$, $1 \leqslant y_i \leqslant y_{max}$, $1 \leqslant z_i \leqslant z_{max}$) — месяц, день и секунда i-го раза, когда магазин, со слов марсианина, был открыт.

i-я из следующих m строк содержит 3 целых числа x_i, y_i, z_i ($1 \leqslant x_i \leqslant x_{max}$, $1 \leqslant y_i \leqslant y_{max}$, $1 \leqslant z_i \leqslant z_{max}$) — месяц, день и секунда i-го раза, когда магазин, со слов марсианина, был закрыт.

i-я из следующих k строк содержит 3 целых числа x_i, y_i, z_i ($1 \leqslant x_i \leqslant x_{max}$, $1 \leqslant y_i \leqslant y_{max}$, $1 \leqslant z_i \leqslant z_{max}$) — месяц, день и секунда i-го запроса Наташи.

Формат выходных данных

Если марсианин ошибся и его сведения о том, когда магазин открыт и когда закрыт, противоречивы, то в единственной строке выведите "`INCORRECT`" (без кавычек).

В противном случае в первой строке выведите "`CORRECT`" (без кавычек). Далее выведите k строк: в i-й из них выведите ответ на i-й запрос Наташи: "`OPEN`" (без кавычек), если магазин в момент этого запроса был открыт, "`CLOSED`" (без кавычек), если был закрыт, или "`UNKNOWN`" (без кавычек), если эту информацию на основе имеющихся данных определить невозможно.

Примеры

стандартный ввод	стандартный вывод
10 10 10 3 1 3 2 6 2 4 2 4 6 4 6 9 9 9 3 3 3 10 10 10 8 8 8	CORRECT OPEN CLOSED UNKNOWN
10 10 10 1 1 1 2 5 7 2 5 7 8 9 10	INCORRECT

Замечание

Рассмотрим первый тестовый пример.
В году 10 месяцев, в месяце 10 дней, в сутках 10 секунд.
Магазин был открыт в 3 момента времени:

- месяц 2, день 6, секунда 2;
- месяц 4, день 2, секунда 4;
- месяц 6, день 4, секунда 6.

Магазин был закрыт в момент времени: месяц 9, день 9, секунда 9.
Запросы:

- месяц 3, день 3, секунда 3 — открыто ("`OPEN`") (поскольку магазин открывается не позже, чем в месяц 2, день 2, секунду 2 и закрывается не раньше, чем в месяц 6, день 6, секунду 6);
- месяц 10, день 10, секунда 10 — закрыто ("`CLOSED`") (поскольку закрыто даже в месяц 9, день 9, секунда 9);
- месяц 8, день 8, секунда 8 — неизвестно ("`UNKNOWN`") (поскольку возможно и расписание, в котором магазин открыт в этот момент, и расписание, в котором магазин закрыт в этот момент).

Во втором тестовом примере магазин был закрыт и открыт в один и тот же момент времени — противоречие ("`INCORRECT`").

Задача M. Сумма минимумов

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	512 мегабайт

Дан массив a длины n. Для запроса $[l, r]$ требуется найти сумму минимумов по всем подотрезкам $[i, j]$, где $l \leqslant i \leqslant j \leqslant r$:

$$\sum_{i=l}^{r} \sum_{j=i}^{r} \min(a_i, a_{i+1}, \ldots, a_j).$$

Необходимо ответить на q запросов.

Формат входных данных

Первая строка содержит одно целое число t ($1 \leqslant t \leqslant 10^4$) — количество наборов входных данных. Далее следуют описания наборов. В каждом наборе:

- первая строка содержит два целых числа n и q ($1 \leqslant n \leqslant 4 \cdot 10^5$, $1 \leqslant q \leqslant 4 \cdot 10^5$);
- вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 10^6$);
- затем идут q строк, i-я из которых содержит два целых числа l_i и r_i ($1 \leqslant l_i \leqslant r_i \leqslant n$).

Гарантируется, что сумма n по всем наборам не превосходит $4 \cdot 10^5$, а сумма q по всем наборам не превосходит $4 \cdot 10^5$.

Формат выходных данных

Для каждого запроса выведите одно целое число — значение суммы минимумов по всем подотрезкам, целиком лежащим внутри отрезка запроса.

Пример

стандартный ввод	стандартный вывод
6 7 4 1 2 3 2 1 2 3 4 6 1 7 2 7 3 7 8 2 4 3 1 1 2 4 3 2 3 6 1 5 5 2 4 5 1 2 3 1 4 1 5 10 8 2 3 6 5 8 9 8 10 10 1 2 7 6 10 1 9 1 6 3 9 4 10 2 10 1 8 10 4 3 4 5 3 4 5 9 10 8 9 1 9 2 10 1 10 2 9 1 1 0 1 1	8 39 32 23 15 23 21 29 113 92 235 88 193 164 226 176 186 210 240 159 0