

## Задача А. 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 —  $1 \leq N \leq 10^5$  — количество чисел, которые выписал Вася и  $1 \leq M \leq 10^5$  — 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  $1 \leq x \leq y \leq N$  и  $1 \leq k \leq l \leq N$ .

### Формат выходных данных

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

### Пример

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

### Замечание

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

## Задача В. 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  $N$  ( $1 \leq N \leq 200\,000$ ) — количество чисел в массиве. Во второй строке вводятся  $N$  чисел от 0 до 100 000 — элементы массива. В третьей строке вводится одно натуральное число  $M$  ( $1 \leq M \leq 200\,000$ ) — 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  $k$  ( $1 \leq k \leq N$ ). 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	4
0 0 3 0 2	
3	
u 1 5	
u 1 0	
s 1 5 3	

### Замечание

TL для Python 8 секунд

## Задача С. 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 А'! 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  $n$  ( $1 \leq n \leq 10^5$ ). После этого в следующих  $n$  строках идут числа  $A_i, B_i$  ( $1 \leq A_i \leq n, 1 \leq B_i \leq 10^9$ ) — 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 D. 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 E. 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:  $a_1, a_2, \dots, a_m$ . 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:  $b_1, b_2, \dots, b_n$ . 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  $a_1, a_2, \dots, a_m$  и начальному положению карт в отсеке для хранения  $b_1, b_2, \dots, b_n$  определяет, какое минимальное количество действий придётся 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  $m$  ( $1 \leq n, m \leq 3 \cdot 10^5$ ) — количество видов товаров и количество товаров, которые необходимо выдать со склада.

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

Третья строка содержит  $n$  различных целых чисел  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq n$ ) — 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 требуется 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 F. 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  $x_r$  ( $1 \leq x_l \leq x_r \leq x_{max}$ ), 2 дня в месяце:  $y_l$  и  $y_r$  ( $1 \leq y_l \leq y_r \leq y_{max}$ ) и 2 секунды в сутках:  $z_l$  и  $z_r$  ( $1 \leq z_l \leq z_r \leq z_{max}$ ). Магазин работает во все такие моменты времени (месяц  $x$ , день  $y$ , секунда  $z$ ), когда одновременно  $x_l \leq x \leq x_r$ ,  $y_l \leq y \leq y_r$  и  $z_l \leq z \leq z_r$ .

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  $x_{max}, y_{max}, z_{max}, n, m, k$  ( $1 \leq x_{max}, y_{max}, z_{max} \leq 10^5$ ,  $1 \leq n \leq 10^5$ ,  $0 \leq m \leq 10^5$ ,  $1 \leq k \leq 10^5$ ) — 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  $x_i, y_i, z_i$  ( $1 \leq x_i \leq x_{max}$ ,  $1 \leq y_i \leq y_{max}$ ,  $1 \leq z_i \leq z_{max}$ ) — месяц, день и секунда  $i$ -го раза, когда магазин, со слов марсианина, был открыт.

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

$i$ -я из следующих  $k$  строк содержит 3 целых числа  $x_i, y_i, z_i$  ( $1 \leq x_i \leq x_{max}$ ,  $1 \leq y_i \leq y_{max}$ ,  $1 \leq z_i \leq z_{max}$ ) — 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 G. Динамический Лес

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

Вам нужно научиться обрабатывать 3 типа запросов:

1. Добавить ребро в граф («link»).
2. Удалить ребро из графа («cut»).
3. По двум вершинам  $a$  и  $b$  вернуть длину пути между ними (или  $-1$ , если они лежат в разных компонентах связности) («get»).

Изначально граф пустой (содержит  $N$  вершин, не содержит ребер). Гарантируется, что в любой момент времени граф является лесом. При добавлении ребра гарантируется, что его сейчас в графе нет. При удалении ребра гарантируется, что оно уже добавлено.

### Формат входных данных

Числа  $N$  и  $M$  ( $1 \leq N \leq 10^5 + 1$ ,  $1 \leq M \leq 10^5$ ) — количество вершин в дереве и, соответственно, запросов.

Далее  $M$  строк, в каждой строке команда («link» или «cut», или «get») и 2 числа от 1 до  $N$  — номера вершин в запросе.

### Формат выходных данных

В выходной файл для каждого запроса «get» выведите одно число — расстояние между вершинами, или  $-1$ , если они лежат в разных компонентах связности.

### Примеры

стандартный ввод	стандартный вывод
3 7 get 1 2 link 1 2 get 1 2 cut 1 2 get 1 2 link 1 2 get 1 2	-1 1 -1 1
5 10 link 1 2 link 2 3 link 4 3 cut 3 4 get 1 2 get 1 3 get 1 4 get 2 3 get 2 4 get 3 4	1 2 -1 1 -1 -1