

## 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: 0.3 секунд  
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

Предполагается, что из всех структур для этой задачи вы выберете дерево Фенвика. Скорее всего, любая другая структура данных получит ТЛ.

## Задача C. Simple 2D

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

Дан набор точек на плоскости. Напишите структуру данных, отвечающую на запросы «количество точек в прямоугольнике».

### Формат входных данных

В первой строке идёт целое число  $N$  ( $N \leq 10^5$ ) — количество точек.

В последующих  $N$  строках идут координаты точек в формате  $(x_i, y_i)$  — числа, не превосходящие по модулю  $10^9$ .

Далее идёт целое число  $M$  ( $M \leq 10^5$ ) — количество запросов.

В последующих  $M$  строках идут запросы в формате  $(x_{1_j}, y_{1_j}, x_{2_j}, y_{2_j})$  — числа, не превосходящие по модулю  $10^9$ , задающие координаты нижнего-левого и верхнего-правого угла  $j$ -ого прямоугольника-запроса. Стороны считаем принадлежащими многоугольнику.

### Формат выходных данных

Для каждого запроса выведите единственное целое число — количество точек, попавших внутрь прямоугольника.

### Пример

стандартный ввод	стандартный вывод
5	3
-1 -1	5
-1 0	0
0 1	1
0 -1	1
1 -1	
5	
0 -3 3 1	
-1 -1 1 1	
3 4 5 6	
-1 -1 -1 -1	
0 0 0 5	

## Задача D. Окна

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

На экране расположены прямоугольные окна, каким-то образом перекрывающиеся (со сторонами, параллельными осям координат). Вам необходимо найти точку, которая покрыта наибольшим числом из них.

### Формат входных данных

В первой строке входного файла записано число окон  $n$  ( $1 \leq n \leq 50\,000$ ).

Следующие  $n$  строк содержат координаты окон  $x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2}$ , где  $(x_{i,1}, y_{i,1})$  — координаты левого верхнего угла  $i$ -го окна, а  $(x_{i,2}, y_{i,2})$  — правого нижнего (на экране компьютера  $y$  растет сверху вниз, а  $x$  — слева направо). Все координаты — целые числа, по модулю не превосходящие  $10^6$ .

### Формат выходных данных

В первой строке выходного файла выведите максимальное число окон, покрывающих какую-либо из точек в данной конфигурации. Во второй строке выведите два целых числа, разделенных пробелом — координаты точки, покрытой максимальным числом окон. Окна считаются замкнутыми, т. е. покрывающими свои граничные точки.

### Примеры

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

## Задача E. Ор выше гор

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

Рик и Морти очень любят ходить на горный хребет Высокоорный для того, чтобы поорать — это там просто невероятное. Не так давно они нашли интересное акустическое свойство этого хребта: если Рик и Морти начнут одновременно орать с разных гор, то их ор будет слышен между этими горами вплоть до высоты, равной побитовому ИЛИ высот гор, на которые они взошли, и всех гор между ними.

Помогите Рикку и Морти посчитать, сколькими способами они могут выбрать две различные горы так, что если они начнут орать с этих гор, ор их будет слышен выше этих гор и всех гор между ними. Формально говоря, требуется вычислить, сколько существует таких пар  $l$  и  $r$  ( $1 \leq l \leq r \leq n$ ), что побитовое ИЛИ всех высот гор на отрезке от  $l$  до  $r$  включительно строго больше высоты любой горы на этом отрезке.

### Формат входных данных

В первой строке содержится целое число  $n$  ( $1 \leq n \leq 200\,000$ ) — количество гор в хребте. В следующей строке содержатся  $n$  целых чисел  $a_i$  ( $0 \leq a_i \leq 10^9$ ) — высоты гор в порядке, в котором они следуют в хребте.

### Формат выходных данных

Выведите одно число — искомое количество способов выбрать две различные горы.

### Примеры

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

### Замечание

В первом примере все искомые способы — это пары гор со следующими номерами (горы нумеруются с единицы):

$[(1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5)]$

Во втором примере искомым пар не существует, поскольку для любой пары гор высота ора с них равна 3, и эта высота равна высоте любой из гор, следовательно она не выше их.

## 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 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 G. Вудист Мирко

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

Юный Мирко решил купить куклу вуду. Учитывая что он крайне заинтересован в том, чтобы купить ее как можно дешевле, он начал отслеживать цены на кукол вуду каждый день. Его список состоит из цен на куклы в последние  $N$  дней, где  $a_i$  обозначает цену куклы  $i$  дней назад.

Мирко думает, что нашел связь между средней ценой кукол в течении нескольких последовательных дней и ценой куклы в следующий день. Он хочет проверить свою догадку и задался вопросом: "Для данного числа  $P$ , сколько существует наборов последовательных дней в течении последних  $N$  дней, для которых средняя цена куклы в эти дни составляет не менее  $P$ ".

Два набора последовательных дней считаются различными, если у них отличается первый или последний день.

### Формат входных данных

Первая строка содержит одно целое число  $N$  ( $1 \leq N \leq 10^6$ ), количество дней в списке Мирко.  
Вторая строка содержит  $N$  целых чисел  $a_i$  ( $0 \leq a_i \leq 10^9$ ) — цены кукол в соответствующие дни.  
Третья строка содержит одно целое число  $P$  ( $0 \leq P \leq 10^9$ ).

### Формат выходных данных

Выведите одно целое число - ответ на вопрос Мирко для данного  $P$ .

### Примеры

стандартный ввод	стандартный вывод
3 1 2 3 3	1
3 1 3 2 2	5
3 1 3 2 3	1

## Задача N. Профессиональный декоратор заборов

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

Вася работает подмастерьем в известной студии. Недавно ему поручили помогать молодому, но подающему большие надежды художественному декоратору заборов и изгородей Витезславу Смолокурову. Миссия эта очень ответственная, и от ее выполнения зависит Васино будущее.

Стиль Смолокурова очень необычен, а его работы пользуются большим спросом. Процесс работы разделен на два этапа. На первом этапе Вася делает заготовку — длинный забор, который состоит из набора цветных вертикальных планок. На втором этапе Витезслав приступает к работе.

Для того, чтобы придать забору более спокойный и гармоничный вид, он несколько раз производит следующую операцию: выбирает некоторый цвет и отрезок, после чего перекрашивает этот отрезок забора в выбранный цвет. По своей творческой натуре, Смолокуров может в корне менять концепцию узора по несколько раз за час, поэтому иногда он перекрашивает одну и ту же планку несколько раз. Кроме того, Витезслав не хочет, чтобы какой-то узор повторялся слишком часто. Для того, чтобы избежать этого, он иногда проверяет, не совпадает ли один отрезок забора с другим.

Несложно догадаться, что и перекрашивание, и проверки осуществляет Вася. Работа эта не самая простая, поэтому Вася просит ему помочь хотя бы с проверками на совпадение.

### Формат входных данных

Первая строка входного файла содержит одно целое число  $n$  — количество планок в заборе ( $1 \leq n \leq 100\,000$ ). Вторая строка содержит  $n$  целых чисел, разделенных пробелами — цвета соответствующих планок.

Третья строка входного файла содержит одно целое число  $m$  — количество сравнений и перекрашиваний ( $1 \leq m \leq 100\,000$ ). Следующие  $m$  строк содержат описания заданий, который Вася получает от Витезслава: четыре целых числа  $q, l, r$  и  $k$ .

В случае перекрашивания  $q = 0$ . Эта запись означает перекрашивание всех планок с  $l$  по  $r$  включительно в цвет  $k$  ( $1 \leq l \leq r \leq n$ ). В запросе на сравнение  $q = 1$ . Эта запись означает сравнение кусков забора длины  $k$  начиная с позиций  $l$  и  $r$  соответственно ( $1 \leq l, r \leq n - k + 1, k > 0$ ).

Все числа во входном файле положительные и не превышают 100 000.

### Формат выходных данных

Выведите одну строку: для каждого запроса на сравнение выведите '+' в случае совпадения соответствующих кусков забора и '-' в противном случае.

### Примеры

стандартный ввод	стандартный вывод
7 1 2 1 3 1 2 1 3 0 4 5 2 1 3 1 2 1 3 1 3	+ -
2 1 2 5 1 1 2 1 0 2 2 1 1 1 2 1 0 1 2 3 1 1 1 2	- + +



## Задача I. 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 J. 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 К. Эффективное тестирование

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

Начиная с 20xx года все организаторы всех школьных олимпиад по программированию договорились проводить соревнования исключительно по интернету, для чего было создано общество с ограниченной ответственностью «Организация онлайн-олимпиад» (ООО «ООО»). Разумеется, такая серьёзная организация не может обойтись без собственной тестирующей системы, поэтому для её создания были наняты эффективные менеджеры, закуплены доски и подготовлена синяя изолента.

Для повышения эффективности процесса тестирования была разработана следующая архитектура. Сначала все  $m$  тестов задачи располагаются в порядке от 1 к  $m$  в очереди тестирования. Затем модуль планирования последовательно выполняет  $n$  действий. Действие  $i$  состоит в том, чтобы выбрать отрезок очереди с позиции  $l_i$  по  $r_i$  включительно (в нумерации с единицы) и проверить решение на каждом втором тесте на этом отрезке, а именно на тестах на позициях  $l_i, l_i+2, l_i+4, \dots, r_i$  очереди (при этом гарантируется, что  $l_i$  и  $r_i$  имеют одинаковую чётность). После этого те тесты, на которых было проведено тестирование, удаляются из очереди, а все оставшиеся тесты сдвигаются по очереди таким образом, чтобы пустых мест не осталось. Например, если в очереди находились тесты с исходными номерами 2, 3, 4, 5, 10, 12, 13, 20 и была применена операция с  $l_i = 3, r_i = 7$ , то посылка будет протестирована на тестах с позиций 3, 5 и 7, которые исходно имели номера 4, 10 и 13. После выполнения данной операции очередь тестирования будет состоять из тестов с исходными номерами 2, 3, 5, 12, 20.

Вам поручено реализовать модуль, который для каждого из  $n$  описанных выше действий будет определять минимальный и максимальный номер теста в изначальной нумерации из тех, на которых на этом шаге проверялось решение.

### Формат входных данных

В первой строке входных данных находятся два числа  $n$  и  $m$  ( $1 \leq n \leq 100\,000, 1 \leq m \leq 10^{18}$ ) — количество действий модуля планирования и количество тестов в задаче.

В каждой из последующих  $n$  строк записаны два целых числа  $l_i$  и  $r_i$  ( $1 \leq l_i \leq r_i \leq m$ ) — параметры  $i$ -го действия модуля планирования. Гарантируется, что перед началом выполнения действия  $i$  в очереди тестирования находятся хотя бы  $r_i$  тестов и что числа  $l_i$  и  $r_i$  имеют одинаковую чётность.

### Формат выходных данных

Для каждого из  $n$  действий модуля планирования выведите два целых числа — минимальный и максимальный номер теста в исходной нумерации из тех, на которых проверялось решение на соответствующем шаге.

### Примеры

стандартный ввод	стандартный вывод
2 10	2 8
2 8	1 5
1 3	
4 6	1 1
1 1	2 2
1 1	3 3
1 1	5 5
2 2	

### Замечание

Рассмотрим, как изменяется очередь тестирования в первом примере.

1. Изначально в очереди тестирования находятся все тесты от 1 до 10, то есть очередь имеет вид 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

2. При выполнении первого запроса будут удалены тесты 2, 4, 6, 8, и очередь примет вид 1, 3, 5, 7, 9, 10.
3. При выполнении второго запроса будут удалены тесты 1 и 5, очередь примет вид 3, 7, 9, 10.

## Задача Lbonus. Приказы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.4 секунд
Ограничение по памяти:	64 мегабайта

Вася работает в НИИГСД (НИИ Государственных Структур Данных). Он изучает приказы правительства далёкого государства.

В том государстве все города расположены вдоль одной дороги. Они пронумерованы в порядке обхода. Изначально качество жизни в каждом из них равно нулю.

Далее последовательно издаются указы вида «уровень жизни в городах с  $i$  по  $j$  должен стать не меньше  $x$ ».

Также есть некоторые официальные заявления. Они имеют следующую форму: «средний уровень жизни в городах с  $i$  по  $j$  равен  $x$ ». Вася нуждается в помощи с проверкой этих утверждений: для каждого из них известны  $i$  и  $j$ , требуется подсчитать верное значение  $x$ .

Можете считать, что каждый приказ исполняется, а также в каждый момент времени каждый город имеет минимальный неотрицательный уровень жизни, удовлетворяющий всем приказам.

### Формат входных данных

Ввод состоит из одного или более тестов. Каждый тест начинается строкой с двумя целыми числами  $n$  и  $k$  — числом городов и событий, соответственно. Следующие  $k$  строк содержат по одному описанию события:

- $\sim i j x$  означает приказ: после этого, все города с номерами от  $i$  до  $j$  включительно должны иметь уровень жизни не менее  $x$  ( $1 \leq x \leq 10^9$ ,  $1 \leq i \leq j \leq n$ ).
- $? i j$  означает официальное заявление: следует подсчитать средний уровень жизни в городах с  $i$  по  $j$  включительно ( $1 \leq i \leq j \leq n$ ).

В конце ввода будет помещён тест с  $n = k = 0$ , который не требуется обрабатывать.

Сумма  $n$  по всему вводу не превысит 100 000. Сумма  $k$  по всему вводу не превысит 100 000.

### Формат выходных данных

Для каждого официального заявления выведите на отдельной строке искомый средний уровень жизни в виде несократимой дроби с наименьшим возможным натуральным знаменателем. Если знаменатель равен 1, выведите вместо дроби целое число. Следуйте формату вывода, как это показано в примере.

### Пример

стандартный ввод	стандартный вывод
10 10	0
? 1 10	1
$\sim$ 1 10 1	10
? 1 10	10
$\sim$ 2 3 10	5
$\sim$ 3 4 5	27/5
? 2 2	16/5
? 3 3	
? 4 4	
? 1 5	
? 1 10	
0 0	