

## Задача А. 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  $1 \leq n \leq 128$ . Координаты кубиков — целые числа от 0 до  $n - 1$ . Далее следуют записи о происшедших событиях по одной в строке. В начале строки записано число  $m$ . Если  $m$  равно:

- 1, то за ним следуют 4 числа —  $x, y, z$  ( $0 \leq x, y, z < N$ ) и  $k$  ( $-20000 \leq k \leq 20000$ ) — координаты кубика и величина, на которую в нем изменилось количество видимых звезд;
- 2, то за ним следуют 6 чисел —  $x_1, y_1, z_1, x_2, y_2, z_2$  ( $0 \leq x_1 \leq x_2 < N, 0 \leq y_1 \leq y_2 < N, 0 \leq z_1 \leq z_2 < N$ ), которые означают, что Петя попросил подсчитать количество звезд в кубиках  $(x, y, z)$  из области:  $x_1 \leq x \leq x_2, y_1 \leq y \leq y_2, z_1 \leq z \leq z_2$ ;
- 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	0
2 1 1 1 1 1 1	1
1 0 0 0 1	4
1 0 1 0 3	2
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	

## Задача В. Прямоугольники

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
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 С. 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 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. King and ICPC

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

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 F. 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:

- $\wedge i j x$  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
$\wedge$ 1 10 1	10
? 1 10	10
$\wedge$ 2 3 10	5
$\wedge$ 3 4 5	27/5
? 2 2	16/5
? 3 3	
? 4 4	
? 1 5	
? 1 10	
0 0	

## Задача G. Застройка мегаполиса

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

В 2000 году Москва оказалась застроена до такой степени, что в черте города совсем не осталось территории, пригодной для постройки новых зданий. В поисках новых источников дохода правительство города приняло план, согласно которому все железнодорожные пути в черте города реконструируются и заменяются на подземные, а высвобожденная поверхность используется для коммерческой аренды.

Планирование будущей застройки было начато с участка Октябрьской железной дороги длиной в  $k$  метров. Так как строить здания поверх образованного туннеля — долгий и сложный процесс, было принято решение закрепить новый участок за наиболее популярными передвижными точками общепита, продающими мороженное, хот-доги, кофе и тому подобное.

Участок для строительства разделён на  $k$  сегментов одинаковой длины, последовательно пронумерованных целыми числами от 1 до  $k$ . Из  $n$  поступивших в правительство заявок на получение территории  $i$ -я претендует на сегменты с  $l_i$  по  $r_i$ , причём соответствующая точка общепита будет оказывать давление величиной  $p_i$  на соответствующий отрезок поверхности. Каждую заявку правительство либо отклонит, либо полностью удовлетворит, предоставив точке все запрошенные сегменты.

Правительство города заинтересовано в том, чтобы сдать каждый сегмент новообразованной территории в аренду хотя бы одной точке общепита. При этом, чтобы уменьшить риск обвала туннеля, было принято решение минимизировать максимальное из оказываемых давлений на каком-либо из сегментов. Обратите внимание, что не запрещается сдавать в аренду один сегмент сразу нескольким точкам общепита, но в таком случае давление, оказываемое ими на данный сегмент поверхности, суммируется.

Помогите правительству одобрить такой набор заявок, чтобы каждый сегмент был сдан в аренду хотя бы одной передвижной точке, но максимальное давление, оказываемое на туннель, было как можно меньше.

### Формат входных данных

В первой строке входных данных находятся два целых числа  $n$  и  $k$  ( $1 \leq n \leq 100\,000$ ,  $1 \leq k \leq 10^9$ ) — количество заявок на открытие точек общепита и количество сегментов поверхности.

В последующих  $n$  строках описаны заявки, каждая из которых задаётся тремя целыми числами  $l_i$ ,  $r_i$ ,  $p_i$  ( $1 \leq l_i \leq r_i \leq 10^9$ ,  $1 \leq p_i \leq 10^9$ ), соответственно границами предприятия и давлением, которое оно оказывает на поверхность туннеля.

### Формат выходных данных

Выведите наименьшее возможное максимальное давление, оказываемое на поверхность туннеля точками общественного питания, при условии, что все сегменты поверхности туннеля сданы в аренду. Если не существует способа покрыть весь участок, выведите  $-1$ .



## Примеры

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

## Замечание

В первом тесте из условия оптимальное решение — принять первые две заявки, тогда максимальное давление, равное 3, будет достигаться на третьем сегменте.

Во втором тесте из условия невозможно сдать в аренду третий сегмент.

В третьем тесте из условия одним из оптимальных решений будет удовлетворить все заявки, тогда максимальное давление, равное 8, будет достигаться на четвёртом сегменте. Обратите внимание, что минимизировать или максимизировать количество удовлетворённых заявок не требуется.

В четвёртом тесте из условия оптимальное решение — удовлетворить первую и четвёртую заявки, тогда на все сегменты будет оказываться одинаковое давление, равное 1.

## Задача Н. 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 I. Квиддич и футбол

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1.5 секунд  
Ограничение по памяти: 256 мегабайт

Гермиона хочет посетить несколько матчей по квиддичу в Хогвартсе, а затем отправиться в Лондон и посетить несколько футбольных матчей там. Гермиона выбрала  $n$  дней в порядке времени и для каждого дня  $i$  она знает, что:

- В Хогвартсе в этот день будет матч с интересностью  $a_i$ .
- В Лондоне в этот день будет матч с интересностью  $b_i$ .

Гермиона перемещается из Хогвартса в Лондон мгновенно. (Она ведь волшебница!) Она может переместиться в Лондон в первый день, а может и не перемещаться туда совсем. Гермиона не может посетить два матча в один день.

Пусть  $q_k$  – это максимально возможная интересность матча с минимальной интересностью среди посещённых матчей в случае, если Гермиона посетит ровно  $k$  матчей. Найдите значения  $q_1, q_2, \dots, q_n$ .

### Формат входных данных

Первая строка содержит целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ).

Вторая строка содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

Третья строка содержит  $n$  целых чисел  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq 10^9$ ).

### Формат выходных данных

Выведите  $n$  чисел  $q_1, q_2, \dots, q_n$ .

### Примеры

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

## Задача J. Простыни

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

В один прекрасный день маленький Дональд решил вымыть  $N$  своих чистых белых простыней. После мытья он положил их на землю во дворе, чтобы их высушить. Дональд помещал простыни так, чтобы никакие две простыни не касались ни сторонами, ни углами, и чтобы их стороны не пересекались, но возможно, что он разместил меньшие простыни поверх более крупных или что простыня полностью закрывает какие-то другие простыни. Сделав это, Дональд лег спать.

Друг Дональда Ким как-то узнал о том, что Дональд сушит простыни и решил пообщаться с ним. Он нашел пейнтбольный пистолет своего отца на чердаке. Наряду с пистолетом у него было несколько пейнтбольных мячей, каждый из них имел свой цвет (не обязательно уникальный). Как только Дональд заснул, Ким вошёл во двор к Дональду и начал стрелять пр простыням из пейнтбольного пистолета. Простыни Дональда очень тонкие, поэтому, когда Ким попадает в какую-то простыню, она пропускает краску дальше, на простыню ниже (и та тоже, и так происходит, пока не закончатся простыни и краска не попадет на землю). После того, как Ким использовал все шары, он с радостью покинул двор Дональда. Дональд был очень расстроен, увидев, что случилось с его простынями. Дональд очень заинтересован в правильных данных о преступлении Кима, но он в шоке и не способен думать, поэтому просит вас сказать ему количество цветов на каждой простыне.

Мы можем представлять двор Дональда как бесконечную систему координат, а простыни - как прямоугольники, параллельные осям координат. Выстрелы Кима могут быть представлены как точки в этой системе.

Когда-то в детстве дедушка рассказал Киму, что снаряд никогда не попадает в одну воронку дважды, так что координаты всех выстрелов попарно различны.

### Формат входных данных

Первая строка ввода содержит два числа - количество простынь  $N$  ( $1 \leq N \leq 300000$ ), и количество шаров  $M$  ( $1 \leq M \leq 300000$ ).

$i$ -я из следующих  $N$  строк содержит четыре числа: координаты нижнего левого угла  $A_i, B_i$  ( $1 \leq A_i, B_i \leq 10^9$ ) и верхнего правого угла  $C_i, D_i$ , ( $1 \leq C_i, D_i \leq 10^9$ )  $i$ -й простыни.

$j$ -я из следующих  $M$  строк содержит три числа, где  $X_j, Y_j$  ( $1 \leq X_j, Y_j \leq 10^9$ ) - координаты  $j$ -го выстрела Кима и  $K_j$  ( $1 \leq K_j \leq 10^9$ ) - цвет  $j$ -го пейнтбольного шара.

### Формат выходных данных

$i$ -я из  $N$  выходных строк должна содержать количество цветов на  $i$ -м листе.

## Примеры

стандартный ввод	стандартный вывод
2 2 1 1 3 3 5 6 10 10 3 3 1 5 1 2	1 0
3 3 1 1 7 7 2 2 6 6 3 3 5 5 4 4 1 2 6 2 4 7 3	3 2 1
1 3 1 1 7 7 2 6 2 4 7 3 4 4 1	3

## Задача К. Катый ноль

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 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

Напишите дерево фенвика. Другие решения мы забаним.

## Задача L. База данных

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

Заведующая учебной частью Байтландской школы в целях контроля успеваемости издала приказ, согласно которому каждый школьник обязан в конце каждой недели предоставлять информацию о своих оценках. Требуется предоставлять лишь свой средний балл, который по мнению завуча полностью характеризует успешность школьника в течение недели.

Родителям понравилось это нововведение, ведь теперь они могут посмотреть уровень успеваемости их ребёнка относительно других школьников. Каждую субботу ученики посылают в базу данных школы свои средние оценки, которые там сохраняются, а затем, родители учеников выполняют  $m$  запросов к данной базе данных.

Пусть  $u$  — максимальное число, находящееся в базе в данный момент времени, а  $cnt(x)$  — количество чисел больше либо равных  $x$  (с повторениями), находящихся в базе. Предусмотрены четыре вида запросов к базе данных:

1. Заменить хранимые в базе числа на последовательность  $(cnt(1), cnt(2), \dots, cnt(u))$ .
2. Добавить в базу некоторое число  $x$ .
3. Удалить из базы одно вхождение некоторого числа  $x$ , если такое имеется.
4. Вывести количество чисел, равных данному числу  $x$ .

Родители начинают запрашивать интересующую их информацию и модернизировать данные лишь после того, как все  $n$  школьников отправят свои оценки.

К сожалению, недавно у школы кончилась лицензия на эту замечательную базу данных, поэтому пока её не продлили, на все запросы придётся отвечать вам.

### Формат входных данных

Первая строка входных данных содержит два числа  $n$  и  $m$  ( $1 \leq n, m \leq 200\,000$ ) — количество школьников и количество запросов к базе данных соответственно.

Во второй строке записаны  $n$  целых чисел  $g_i$  ( $1 \leq g_i \leq 200\,000$ ) — значения средних оценок школьников.

Далее следуют  $m$  строк, описывающих запросы в порядке их применения. Каждое описание начинается с одной из букв «t», «a», «r» или «c», означающих запрос первого, второго, третьего или четвертого типа соответственно. Для запросов второго, третьего и четвертого типа далее следует число  $x_i$  ( $1 \leq x_i \leq 200\,000$ ) — параметр запроса.

### Формат выходных данных

Сначала выведите ответы на все запросы четвертого типа. Затем выведите все числа, находящиеся в базе после выполнения всех запросов, в порядке неубывания.

Гарантируется, что в выходных данных должно будет присутствовать хотя бы одно число.



## Пример

стандартный ввод	стандартный вывод
6 8	0
4 3 3 3 6 6	2
t	3 3 5 7 7
c 4	
a 5	
a 3	
r 5	
c 2	
t	
r 3	

## Замечание

Рассмотрим как меняется последовательность из примера:

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

Ниже предоставлены критерии оценки:

№	Баллы	Ограничения	Необх. группы	Комментарий
0	0	—	—	Тесты из условия.
1	15	$n, m, x_i \leq 200$	0	—
2	15	$n, m, x_i \leq 3000$	0 – 1	—
3	35	—	—	Запросы только 1 и 4 типа.
4	35	—	0 – 3	—