

Задача D. Прямоугольники

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

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

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

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

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

Формат входных данных

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

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

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

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

Все числа во входном файле целые.

Формат выходных данных

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

Пример

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

Задача F. Минимум в прямоугольнике

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 6 секунд
Ограничение по памяти: 256 мегабайт

Вам дана таблица из $n \cdot m$ целых чисел и последовательность из q запросов $(x_{i_1}, y_{i_1}, x_{i_2}, y_{i_2})$. Для каждого запроса найдите минимальное значение на прямоугольнике, состоящем из чисел a_{xy} по всем (x, y) , удовлетворяющим условиям $x_{i_1} \leq x \leq x_{i_2}$ и $y_{i_1} \leq y \leq y_{i_2}$

Формат входных данных

Первая строка входного файла содержит числа m и n — размеры таблицы ($1 \leq m, n \leq 500$). Следующие m строк содержат по n целых чисел каждая — значения таблицы. Все значения лежат в диапазоне значений типа данных `int` для C++.

Далее следует число q — количество запросов ($1 \leq q \leq 2 \cdot 10^5$). Следующие q строк содержат по четыре целых числа: $x_{i_1}, y_{i_1}, x_{i_2}, y_{i_2}$ ($1 \leq x_{i_1} \leq x_{i_2} \leq m, 1 \leq y_{i_1} \leq y_{i_2} \leq n$).

Формат выходных данных

Выведите q чисел — ответы на запросы в порядке их следования во входном файле.

Пример

стандартный ввод	стандартный вывод
1 1	-2147483648
-2147483648	
1	
1 1 1 1	

Замечание

Решения, не отвечающие в онлайн на запросы, будут баниться.

Задача G. Фэйк Ньюз

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

В Бейтландии зарегистрировано несколько информационных агентств, в том числе новое информационное агентство ООО «неверлай дот ком». Начальник этой организации хочет выпустить сенсационное расследование. Последний месяц он ждет подходящего момента, чтобы новость стала как можно более популярной.

У каждого агентства есть свой показатель авторитетности A , который может отличаться для разных агентств. Процесс публикации новости в Бейтландии устроен так: первым делом агентство ООО «неверлай дот ком» публикует новость. Это запускает цепную реакцию, устроенную следующим образом:

- Каждое агентство из тех, кто еще не опубликовал данную новость, вычисляет число P : количество других агентств, опубликовавших новость.
- Если агентство имеет показатель авторитетности A , а число уже опубликовавших новость агентств P не меньше A , то это агентство публикует данную новость.
- Процесс повторяется до тех пор, пока предыдущее условие выполняется хотя бы для одного из агентств.

ООО «неверлай дот ком» хочет выбрать момент, когда им стоит опубликовать новость, но в Бейтландии часто появляются новые информационные агентства, а старые часто закрываются. От вас требуется обрабатывать запросы двух типов:

1. «+ A » появляется новое агентство с показателем авторитетности A
2. «- A » одно из агентств с показателем авторитетности A прекращает работу.

Для запросов второго типа гарантируется, что существовало хотя бы одно СМИ с показателем авторитетности A . Обратите внимание, что в один момент может существовать несколько агентств с одинаковым показателем авторитетности A .

После каждого запроса вам требуется определить, сколько агентств опубликуют новость, если ООО «неверлай дот ком» опубликует ее прямо сейчас.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 500\,000$) — количество запросов.

В следующих n строках вводятся запросы в формате, описанном в условии задачи: сначала задаётся один из символов «+», «-», затем следует целое число A — показатель авторитетности ($1 \leq A \leq 500\,000$).

Формат выходных данных

Выведите n строк. В каждой строке должно быть одно число — количество агентств, которые опубликуют новость от ООО «неверлай дот ком» после i -го запроса.

Примеры

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

Замечание

Рассмотрим первый пример:

- После первого запроса ни одно другое агенство не будет публиковать новость от ООО «неверлай дот ком».
- После второго запроса появилось агенство, которое готово опубликовать новость.
- После третьего запроса все еще лишь одно агенство готово опубликовать новость.
- После четвёртого запроса процесс публикации будет устроен следующим образом:
 1. ООО «неверлай дот ком» публикует новость.
 2. агенство с $A = 1$ публикует новость.
 3. агенство с $A = 2$ публикует новость.
 4. агенство с $A = 3$ публикует новость.
 5. агенство с $A = 4$ публикует новость.
- После последнего запроса процесс публикации остановится на агенстве с $A = 2$.

Задача Н. 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 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 ($0 \leq x, y, z < N$) 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 I. Беспилотное такси

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

На главной площади Иннополиса планируется демонстрация возможностей беспилотных такси. Площадь имеет форму прямоугольника размером $n \times m$ метров, разделённого на единичные квадраты. Строки прямоугольника пронумерованы от 1 до n , а столбцы — от 1 до m . Таким образом, каждый квадрат характеризуется двумя положительными числами r и c — номерами строки и столбца, на пересечении которых он находится. При движении по площади беспилотное такси перемещается между единичными квадратами, за один *шаг* оно может переместиться из квадрата, в котором оно сейчас находится, в квадрат, имеющий с ним общую сторону.

Зима в этом году выдалась снежная, и для уборки площади в процессе демонстрации планируется использовать беспилотные снегоуборочные машины. При этом система управления беспилотным транспортом находится в режиме тестирования, поэтому одновременно на площади может находиться только одно беспилотное транспортное средство.

Поскольку снег продолжает идти в течение всей демонстрации, инженеры беспилотного такси решили продемонстрировать его возможности по преодолению сугробов. В каждый момент времени *глубина снега* на каждом единичном квадрате площади характеризуется неотрицательным целым числом. *Проходимость* беспилотного такси также задаётся некоторым неотрицательным целым числом. При перемещении по площади такси может находиться только на тех единичных квадратах, глубина снега на которых не превышает его проходимости. Перед каждой поездкой инженеры могут настроить такси, задав при этом значение его проходимости.

Исходно глубина снега на всей площади равна нулю. В начале каждого часа глубина снега на каждом квадрате увеличивается на один. После этого система управления беспилотным транспортом может выполнить одно из трех действий, каждое из которых занимает ровно один час:

- 1) запустить снегоуборочную машину вдоль строки, после этого на всех квадратах этой строки глубина снега становится равной нулю;
- 2) запустить снегоуборочную машину вдоль столбца, после этого на всех квадратах этого столбца глубина снега становится равной нулю;
- 3) продемонстрировать возможности беспилотного такси: задать значение его проходимости, выбрать стартовый и финишный единичные квадраты, и попытаться проехать на такси от стартового квадрата до финишного.

Поездка от стартового до финишного квадрата возможна, если существует последовательность единичных квадратов, начинающаяся в стартовом единичном квадрате и завершающаяся в финишном, в которой любые два соседних квадрата имеют общую сторону, и глубина снега на каждом из квадратов последовательности не превышает проходимости такси. Если выставленное значение проходимости позволяет осуществить поездку, такси должно использовать путь с минимальным количеством шагов.

Для каждой поездки на беспилотном такси требуется определить, возможно ли осуществить поездку, и если да, то какое минимальное количество шагов потребуется для осуществления этой поездки.

Формат входных данных

Первая строка ввода содержит три целых числа n , m и q — размеры площади и количество действий, которые совершаются системой управления беспилотными такси ($1 \leq n, m \leq 10^6$, $1 \leq q \leq 300\,000$).

Следующие q строк содержат описания действий, i -я из них имеет один из следующих трех типов:

- $\langle 1 \ r_i \rangle$ — запустить снегоуборочную машину вдоль строки r_i ($1 \leq r_i \leq n$).

- «2 c_i » — запустить снегоуборочную машину вдоль столбца c_i ($1 \leq c_i \leq m$).
- «3 $r_{i,1}$ $c_{i,1}$ $r_{i,2}$ $c_{i,2}$ k_i » — задать для такси значение проходимости, равное k_i , и попытаться проехать от единичного квадрата $(r_{i,1}, c_{i,1})$ до единичного квадрата $(r_{i,2}, c_{i,2})$ ($1 \leq r_{i,1}, r_{i,2} \leq n$, $1 \leq c_{i,1}, c_{i,2} \leq m$, $0 \leq k_i \leq q$, квадраты $(r_{i,1}, c_{i,1})$ и $(r_{i,2}, c_{i,2})$ различны).

Гарантируется, что во входных данных есть хотя бы одно действие типа 3.

Формат выходных данных

Для каждого действия типа 3 требуется вывести отдельную строку. Если поездка от стартового до финишного квадрата возможна, необходимо вывести минимальное количество шагов, необходимое для осуществления поездки. Если поездка невозможна, необходимо вывести число -1 .

Пример

стандартный ввод	стандартный вывод
4 3 9	3
3 2 1 3 3 1	-1
3 2 1 3 3 1	5
2 1	-1
2 3	3
1 1	
3 2 1 3 3 3	
3 2 1 3 3 3	
2 2	
3 2 1 3 3 6	

Задача J. 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 К. Шахматные баталии

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

Ильдар и Ваня устали постоянно играть в шахматы, поэтому они придумали новую шахматную игру.

Игра происходит на шахматном поле размером $2n \times 2m$. Это поле имеет $2n$ строк и $2m$ столбцов. Клетки этого поля покрашены в черный и белый цвета шахматной раскраской. Более точно, клетка в i -й строке и j -м столбце имеет белый цвет, если $i + j$ чётно, и чёрный цвет в противном случае.

Игра устроена следующим образом. Ильдар помечает некоторые **белые** клетки поля как недоступные. После этого он предлагает Ване попробовать решить следующую задачу: может ли он на доступных **белых** клетках поля расставить $n \times m$ шахматных королей так, что никакие два выставленных короля не бьют друг друга, то есть не стоят на клетках поля, соседних по стороне или углу.

Конечно, Ильдар планирует сделать игру интересной и предложить Ване несколько сложных комбинаций недоступных клеток. Для этого он попросил у вас помощи. Чтобы перед игрой понять, как лучше всего действовать, он хочет потренироваться. Для этого, он берёт пустое поле и хочет q раз либо сделать какую-то доступную клетку недоступной, либо сделать какую-то недоступную клетку доступной. После каждого изменения он бы хотел знать, каким будет ответ на задачу для Вани для текущего множества недоступных клеток.

Помогите Ильдару сделать игру интересной! Напишите программу, которая будет отвечать на его запросы.

Формат входных данных

В первой строке находится три целых числа n, m, q ($1 \leq n, m, q \leq 200\,000$) — количество пар строк шахматной доски, количество пар столбцов шахматной доски и количество запросов.

Следующие q строк описывают запросы Ильдара. Каждая из этих строк содержит два целых числа i, j ($1 \leq i \leq 2n, 1 \leq j \leq 2m, i + j$ чётно). Если клетка (i, j) была недоступна, она становится доступной, в противном случае она становится недоступной.

Формат выходных данных

Выведите q строк. В i -й из этих строк выведите ответ на задачу для доски, полученной после i первых запросов Ильдара.

Выведите «YES», если Ваня может так расставить шахматных королей на незанятые белые клетки поля, что никакие два короля не будут бить друг друга, и «NO», иначе.

Примеры

стандартный ввод	стандартный вывод
1 3 3	YES
1 1	YES
1 5	NO
2 4	
3 2 10	YES
4 2	YES
6 4	NO
1 3	NO
4 2	YES
6 4	YES
2 2	NO
2 4	YES
1 3	YES
4 4	NO
3 1	

Замечание

В первом примере, после второго запроса пешки будут стоять на клетках (1, 1) и (1, 5). Тогда Ваня может поставить три короля на клетки (2, 2), (2, 4) и (2, 6).

После третьего запроса пешки будут стоять на клетках (1, 1), (1, 5) и (2, 4). Тогда отстаетя всего три пустые клетки (2, 2), (1, 3) и (2, 6). Ваня не может поставить трех королей на эти клетки, потому что короли в клетках (2, 2) и (1, 3) бьют друг друга, так как эти клетки соседние по углу.

Задача М. Фокус с подмножествами

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

Ваня придумал интересный фокус с множеством целых чисел.

Пусть у фокусника есть множество положительных целых чисел S . Он называет некоторое положительное целое число x . Зритель должен выбрать, не показывая фокуснику, некоторое подмножество S (возможно пустое). После этого зритель называет фокуснику размер выбранного подмножества. Фокус заключается в том, что после этого фокусник отгадывает: верно ли, что сумма элементов выбранного подмножества не превосходит x . Для пустого подмножества сумма предполагается равной 0.

Ване очень понравился этот фокус, поэтому он начал готовиться к тому, чтобы показать его публике. Для этого он приготовил некоторое множество **различных** положительных целых чисел S . Конечно, Ваня хочет, чтобы фокус обязательно получился. Он называет положительное целое число x *неудачным*, если не может быть точно уверен, что фокус пройдет удачно для любого подмножества, которое выберет зритель.

Чтобы оценить, насколько хорошее множество S он выбрал, он хочет посчитать количество неудачных положительных целых чисел для него.

Также Ваня планирует протестировать различные множества S . Поэтому он просит вас написать программу, которая найдет количество неудачных положительных целых чисел для изначального множества S и для множества S после каждого изменения. Ваня сделает q изменений своего множества, каждое изменение будет одного из двух видов:

- Добавить новое число a в множество S .
- Удалить некоторое число a из множества S .

Формат входных данных

В первой строке находятся два целых числа n, q ($1 \leq n, q \leq 200\,000$) — размер изначального множества S и количество изменений.

В следующей строке находятся n **различных** целых чисел s_1, s_2, \dots, s_n ($1 \leq s_i \leq 10^{13}$) — элементы изначального множества S .

В каждой из следующих q строк находятся два целых числа t_i, a_i ($1 \leq t_i \leq 2, 1 \leq a_i \leq 10^{13}$), описывающих очередное изменение.

- Если $t_i = 1$, то это операция добавления нового числа a_i в множество S . Гарантируется, что этого числа не было в множестве S до выполнения операции.
- Если $t_i = 2$, то это операция удаления числа a_i из множества S . Гарантируется, что это число было в множестве S до выполнения операции.

Формат выходных данных

Выведите $q + 1$ строку.

В первой строке выведите количество неудачных положительных целых чисел для изначального множества S . В следующих q строках выведите количество неудачных положительных чисел для множества S после каждого изменения.

Пример

стандартный ввод	стандартный вывод
3 11	4
1 2 3	1
2 1	6
1 5	12
1 6	19
1 7	13
2 6	8
2 2	2
2 3	10
1 10	3
2 5	0
2 7	0
2 10	

Замечание

В первом тесте изначальное $S = \{1, 2, 3\}$. Для этого множества фокус может не получиться при $x \in \{1, 2, 3, 4\}$. Например, если $x = 4$, то зритель может загадать подмножество $\{1, 2\}$, сумма элементов которого равна $3 \leq x$, а может загадать подмножество $\{2, 3\}$, сумма элементов которого равна $5 > x$. Однако в обоих случаях зритель назовет фокуснику размер подмножества 2, поэтому он не сможет точно сделать правильный ответ. При этом поскольку подмножество размера 3 единственно, а сумма в любом подмножестве меньшего размера не превосходит 5, все $x \geq 5$ не являются неудачными.

Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов всех необходимых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Ограничения			Необх. группы	Комментарий
		n	q	Доп. ограничения		
0	0	—	—	—	—	Тесты из условия.
1	10	$n \leq 12$	$q \leq 100$	все $ S \leq 12$	0	
2	10	$n \leq 100$	$q \leq 100$	$s_i, a \leq 100$	0	
3	20	$n \leq 2000$	$q \leq 2000$	—	0 – 2	
4	15	$n \leq 50\,000$	$q \leq 50\,000$	—	0 – 3	
5	15	$n \leq 100\,000$	$q \leq 100\,000$	—	0 – 4	
6	15	$n \leq 150\,000$	$q \leq 150\,000$	—	0 – 5	Offline-проверка.
7	15	—	—	—	0 – 6	Offline-проверка.

Задача N. Прибавление

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

Дан двумерный массив целых чисел a , состоящий из n строк и m столбцов. Изначально все числа в массиве равны нулю. Необходимо обработать q запросов следующего вида:

- 1 r_1 c_1 r_2 c_2 t ($1 \leq r_1 \leq r_2 \leq n$, $1 \leq c_1 \leq c_2 \leq m - t + 1$, $1 \leq t \leq m$). Назовём операцией *прибавления в прямоугольнике*, заданном углами (x_1, y_1) и (x_2, y_2) , где $x_1 \leq x_2$, $y_1 \leq y_2$, процедуру, в результате которой ко всем элементам массива $a_{x,y}$ для которых $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$, прибавляется 1.

Необходимо для всех i , таких что $0 \leq i < t$ выполнить операцию прибавления в прямоугольнике, заданном углами $(r_1, c_1 + i)$, $(r_2, c_2 + i)$.

- 2 r_1 c_1 r_2 c_2 ($1 \leq r_1 \leq r_2 \leq n$, $1 \leq c_1 \leq c_2 \leq m$). Необходимо вычислить сумму элементов $a_{x,y}$ в прямоугольнике, заданном углами (r_1, c_1) , (r_2, c_2) . Поскольку данная сумма может быть очень большой, надо вычислить остаток от её деления на 2^{31} .

Формат входных данных

В первой строке заданы три целых числа n , m и q ($1 \leq n, m \leq 10^9$, $1 \leq q \leq 100\,000$) — размеры массива и количество запросов.

В следующих q строках находятся описания запросов в формате, описанном выше.

Формат выходных данных

Для каждого запроса второго типа выведите ответ в отдельной строке. Гарантируется, что хотя бы один запрос имеет тип 2.

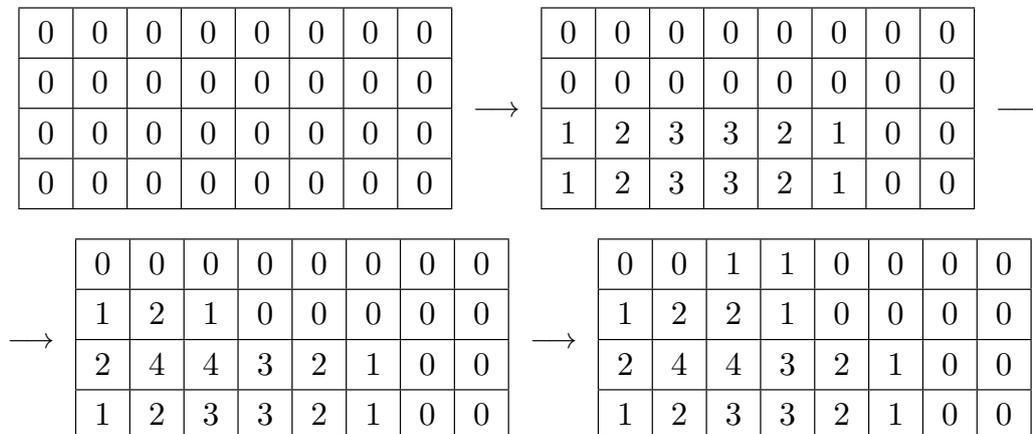
Примеры

стандартный ввод	стандартный вывод
1 4 2 1 1 1 1 2 3 2 1 1 1 4	6
4 8 6 1 1 1 2 4 3 1 2 1 3 2 2 1 3 3 4 4 1 2 3 3 4 6 2 1 2 3 5 2 1 5 4 8	5 28 6
6 6 6 1 1 1 3 5 1 1 2 1 4 4 3 2 1 1 6 5 1 2 2 6 5 1 1 2 2 5 6 1 2 2 2 3 4	48 34
13 8 8 2 2 1 8 5 1 4 2 7 8 1 2 1 2 8 4 1 3 2 13 5 2 1 4 2 8 5 3 1 2 2 9 4 1 1 4 2 9 5 2 2 3 2 7 5	0 12 130

Замечание

В первом примере мы имеем дело с одномерным массивом длины 4. Изначально массив равен $(0, 0, 0, 0)$. После первого запроса к элементам на отрезках $[1; 2]$, $[2; 3]$ и $[3; 4]$ прибавляем по 1, и получается массив $(1, 2, 2, 1)$ с суммой элементов 6.

Во втором примере в результате первых трёх запросов происходят следующие преобразования (строки нумеруются снизу вверх):



Система оценки

Тесты к этой задаче состоят из девяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		n, m	q		
0	0	–	–	–	Тесты из условия.
1	7	$n, m \leq 100$	$q \leq 100$	0	
2	8	$n, m \leq 700$	$q \leq 700$	0 – 1	
3	10	–	$q \leq 10\,000$	–	$t_i = 1$.
4	15	–	$q \leq 10\,000$	0 – 3	
5	9	–	–	–	В запросах второго типа $(r_{i,1}, c_{i,1}) = (1, 1)$, $(r_{i,2}, c_{i,2}) = (n, m)$.
6	11	$n, m \leq 1000$	–	–	Все запросы типа 2 идут после запросов типа 1, $t_i = 1$.
7	10	$n, m \leq 1000$	–	6	Все запросы типа 2 идут после запросов типа 1.
8	14	–	–	3, 6	$t_i = 1$.
9	16	–	–	0 – 8	

Задача О. Медианный горный хребет

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

Берляндия — огромная страна с разнообразной географией. Одной из самых знаменитых природных достопримечательностей Берляндии является «Медианный горный хребет». Этот горный хребет представляет из себя n подряд идущих горных вершин, расположенных на одной прямой, пронумерованных в порядке следования от 1 до n . Высота i -й горной вершины равна a_i .

«Медианный горный хребет» знаменит тем, что с ним ежедневно происходит так называемое *выравнивание горных вершин*. В момент выравнивания одновременно для каждой вершины от 2 до $n - 1$ её высота становится равна медианной высоте среди неё и двух соседних гор. А именно, если до выравнивания высоты были равны b_i , то новые высоты a_i устроены следующим образом: $a_1 = b_1$, $a_n = b_n$, а для всех i от 2 до $n - 1$ $a_i = \text{median}(b_{i-1}, b_i, b_{i+1})$. Медианой трёх чисел называется второе по счёту число, если отсортировать эти три числа по возрастанию. Например, $\text{median}(5, 1, 2) = 2$, а $\text{median}(4, 2, 4) = 4$.

Недавно Берляндские учёные доказали, что какими бы ни были высоты гор, процесс выравнивания рано или поздно стабилизируется, то есть в какой-то момент высоты гор перестанут изменяться после выравнивания. Правительство Берляндии хочет понять через сколько лет это произойдёт, то есть, найти величину s — сколько произойдет выравниваний, при которых у хотя бы одной горы изменится её высота. Помогите ученым решить эту важную задачу!

Обратите внимание, что в некоторых группах тестов помимо значения s вам необходимо определить также и высоты гор после s выравниваний, то есть узнать, какими высоты гор останутся навсегда.

Формат входных данных

Первая строка содержит целые числа n и t ($1 \leq n \leq 500\,000$, $0 \leq t \leq 1$) — количество гор и параметр, который определяет, необходимо ли определить итоговые высоты гор.

Вторая строка содержит целые числа $a_1, a_2, a_3, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — текущие высоты гор.

Формат выходных данных

В первой строке выведите s — число выравниваний вершин, при которых высота хотя бы одной горы изменится.

Если $t = 1$, то во второй строке выведите n чисел — итоговые высоты гор после s выравниваний.

Примеры

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

Замечание

В первом примере высоты на позициях 1 и 5 не меняются. Так как медиана чисел 1, 2, 1 это 1, то на позициях 2 и 4 после первого выравнивания оказываются числа 1, и так как медиана чисел 2, 1, 2 это 2, то на позиции 3 после первого выравнивания оказывается число 2. Итого после первого выравнивания горных вершин горы имеют высоты 1, 1, 2, 1, 1. После второго выравнивания высоты становятся 1, 1, 1, 1, 1 и дальше они меняться не будут, соответственно всего было 2 меняющих высоты выравнивания.

В третьем примере после выравнивания ни у одной горы её высота не изменится и число выравниваний, при которых высоты изменятся, равно 0. Так как $t = 0$, то выводить итоговые высоты гор не нужно.

Система оценки

Тесты к этой задаче состоят из шести групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения			Необх. группы	Комментарий
		n	a_i	t		
0	0	–	–	–	–	Тесты из условия.
1	19	$n \leq 1000$	–	–	0	Гарантируется, что $c \leq 10\,000$.
2	24	–	$a_i \leq 2$	–	–	
3	14	–	$a_i \leq 100$	$t = 0$	–	
4	14	–	$a_i \leq 100$	–	0, 2, 3	
5	14	–	–	$t = 0$	3	
6	15	–	–	–	0, 1, 2, 3, 4, 5	Offline-проверка.