

Задача А. Звезды

Имя входного файла: `stars.in`
 Имя выходного файла: `stars.out`
 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 входного файла

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 выходного файла

Для каждого Петингого вопроса выведите искомое количество звезд.

Примеры

stars.in	stars.out
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. Range Minimum Query

Имя входного файла: `rmq.in`
 Имя выходного файла: `rmq.out`
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 256 мегабайт

Компания *Giggle* открывает свой новый офис в Судиславле, и вы приглашены на собеседование. Ваша задача — решить поставленную задачу.

Вам нужно создать структуру данных, которая представляет из себя массив целых чисел. Изначально массив пуст. Вам нужно поддерживать две операции:

- запрос: «? i j » — возвращает минимальный элемент между i -ым и j -м, включительно;
- изменение: «+ i x » — добавить элемент x после i -го элемента списка. Если $i = 0$, то элемент добавляется в начало массива.

Конечно, эта структура должна быть достаточно хорошей.

Формат входного файла

Первая строка входного файла содержит единственное целое число n — число операций над массивом ($1 \leq n \leq 200\,000$). Следующие n строк описывают сами операции. Все операции добавления являются корректными. Все числа, хранящиеся в массиве, по модулю не превосходят 10^9 .

Формат выходного файла

Для каждой операции в отдельной строке выведите её результат.

Примеры

rmq.in	rmq.out
8	4
+ 0 5	3
+ 1 3	1
+ 1 4	
? 1 2	
+ 0 2	
? 2 4	
+ 4 1	
? 3 5	

Задача С. Вперед!

Имя входного файла: `movetofront.in`
 Имя выходного файла: `movetofront.out`
 Ограничение по времени: 3 секунды
 Ограничение по памяти: 256 мегабайт

Капрал Дукар любит раздавать приказы своей роте. Самый любимый его приказ —

“Вперёд!”. Капрал строит солдат в ряд и отдаёт некоторое количество приказов, каждый из них звучит так: “Рядовые с l_i по l_j — вперёд!”

Перед тем, как Дукар отдал первый приказ, солдаты были пронумерованы от 1 до n , слева направо. Услышав приказ “Рядовые с l_i по l_j — вперёд!”, солдаты, стоящие на местах с l_i по l_j включительно, продвигаются в начало ряда, в том же порядке, в котором были.

Например, если в какой-то момент солдаты стоят в порядке 2, 3, 6, 1, 5, 4, то после приказа “Рядовые с 2 по 4 — вперёд!”, порядок будет таким: 3, 6, 1, 2, 5, 4. А если потом Капрал вышлет вперёд солдат с 3 по 4, то порядок будет уже таким: 1, 2, 3, 6, 5, 4.

Вам дана последовательность из приказов Капрала. Найдите порядок, в котором будут стоять солдаты после исполнения всех приказов.

Формат входного файла

В первой строке входного файла указаны числа n и m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 100\,000$) — число солдат и число приказов. Следующие m строк содержат приказы в виде двух целых чисел: l_i и r_i ($1 \leq l_i \leq r_i \leq n$).

Формат выходного файла

Выведите в выходной файл n целых чисел — порядок, в котором будут стоять солдаты после исполнения всех приказов.

Примеры

movetofront.in	movetofront.out
6 3	1 4 5 2 3 6
2 4	
3 5	
2 2	

Задача D. Откат

Имя входного файла: `rollback.in`
 Имя выходного файла: `rollback.out`
 Ограничение по времени: 4 секунды
 Ограничение по памяти: 256 мегабайта

Сергей работает системным администратором в очень крупной компании. Естественно, в круг его обязанностей входит резервное копирование информации, хранящейся на различных серверах и «откат» к предыдущей версии в случае возникновения проблем.

В данный момент Сергей борется с проблемой недостатка места для хранения информации для восстановления. Он решил перенести часть информации на новые сервера. К сожалению, если что-то случится во время переноса, он не сможет произвести откат, поэтому процедура переноса должна быть тщательно спланирована.

На данный момент у Сергея хранятся n точек восстановления различных серверов, пронумерованных от 1 до n . Точка восстановления с номером i позволяет произвести откат для сервера a_i . Сергей решил разбить перенос на этапы, при этом на каждом этапе в случае возникновения проблем будут доступны точки восстановления с номерами $l, l+1, \dots, r$ для некоторых l и r .

Для того, чтобы спланировать перенос данных оптимальным образом, Сергею необходимо научиться отвечать на запросы: для заданного l , при каком минимальном r в процессе переноса будут доступны точки восстановления не менее чем k различных серверов.

Помогите Сергею.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m , разделенные пробелами — количество точек восстановления и количество серверов ($1 \leq n, m \leq 100\,000$). Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n — номера серверов, которым соответствуют точки восстановления ($1 \leq a_i \leq m$).

Третья строка входного файла содержит q — количество запросов, которые необходимо обработать ($1 \leq q \leq 100\,000$). В процессе обработки запросов необходимо поддерживать число p , исходно оно равно 0. Каждый запрос задается парой чисел x_i и y_i , используйте их для получения данных запроса следующим образом: $l_i = ((x_i + p) \bmod n) + 1$, $k_i = ((y_i + p) \bmod m) + 1$ ($1 \leq l_i, x_i \leq n$, $1 \leq k_i, y_i \leq m$). Пусть ответ на i -й запрос равен r . После выполнения этого запроса, следует присвоить p значение r .

Формат выходного файла

На каждый запрос выведите одно число — искомое минимальное r , либо 0, если такого r не существует.

Примеры

rollback.in	rollback.out
7 3	1
1 2 1 3 1 2 1	4
4	0
7 3	6
7 1	
7 1	
2 2	