

## Задача А. Катый ноль

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

Реализуйте эффективную структуру данных, позволяющую изменять элементы массива и вычислять индекс  $k$ -го слева нуля на данном отрезке в массиве.

### Формат входных данных

В первой строке вводится одно натуральное число  $N$  ( $1 \leq N \leq 200\,000$ ) — количество чисел в массиве. Во второй строке вводятся  $N$  чисел от 0 до 100 000 — элементы массива. В третьей строке вводится одно натуральное число  $M$  ( $1 \leq M \leq 200\,000$ ) — количество запросов. Каждая из следующих  $M$  строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (`s` — вычислить индекс  $k$ -го нуля, `u` — обновить значение элемента). Следом за `s` вводится три числа — левый и правый концы отрезка и число  $k$  ( $1 \leq k \leq N$ ). Следом за `u` вводятся два числа — номер элемента и его новое значение.

### Формат выходных данных

Для каждого запроса  $s$  выведите результат. Все числа выводите в одну строку через пробел. Если нужного числа нулей на запрашиваемом отрезке нет, выводите  $-1$  для данного запроса.

### Примеры

<code>kthzero.in</code>	<code>kthzero.out</code>
5	4
0 0 3 0 2	
3	
u 1 5	
u 1 0	
s 1 5 3	

### Замечание

TL для Python 8 секунд

## Задача В. Разрезанные таблицы

Имя входного файла: `sparse.in`  
Имя выходного файла: `sparse.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Дан массив из  $n$  чисел. Требуется написать программу, которая будет отвечать на запросы следующего вида: найти минимум на отрезке между  $u$  и  $v$  включительно.

### Формат входных данных

В первой строке входного файла даны три натуральных числа  $n$ ,  $m$  ( $1 \leq n \leq 10^5$ ,  $1 \leq m \leq 10^7$ ) и  $a_1$  ( $0 \leq a_1 < 16\,714\,589$ ) — количество элементов в массиве, количество запросов и первый элемент массива соответственно. Вторая строка содержит два натуральных числа  $u_1$  и  $v_1$  ( $1 \leq u_1, v_1 \leq n$ ) — первый запрос.

Элементы  $a_2, a_3, \dots, a_n$  задаются следующей формулой:

$$a_{i+1} = (23 \cdot a_i + 21563) \bmod 16714589.$$

Например, при  $n = 10$ ,  $a_1 = 12345$  получается следующий массив:  $a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095)$ .

Запросы генерируются следующим образом:

$$u_{i+1} = ((17 \cdot u_i + 751 + ans_i + 2i) \bmod n) + 1,$$
$$v_{i+1} = ((13 \cdot v_i + 593 + ans_i + 5i) \bmod n) + 1,$$

где  $ans_i$  — ответ на запрос номер  $i$ .

Обратите внимание, что  $u_i$  может быть больше, чем  $v_i$ .

### Формат выходных данных

В выходной файл выведите  $u_m$ ,  $v_m$  и  $ans_m$  (последний запрос и ответ на него).

### Примеры

<code>sparse.in</code>	<code>sparse.out</code>
10 8 12345 3 9	5 3 1565158

### Замечание

Пояснение к тесту из примера: запросы и результаты.

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$
12345	305498	7048017	11694653	1565158	2591019	9471233	570265	13137658	1325095

#	$u$	$v$	$ans$
1	3	9	570265
2	10	1	12345
3	1	2	12345
4	10	10	1325095
5	5	9	570265
6	2	1	12345
7	3	2	305498
8	5	3	1565158

## Задача С. Окна

Имя входного файла: `windows.in`  
Имя выходного файла: `windows.out`  
Ограничение по времени: 2 секунды  
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  $x_{(1,i)}$   $y_{(1,i)}$   $x_{(2,i)}$   $y_{(2,i)}$ , где  $(x_{(1,i)}, y_{(1,i)})$  — координаты левого верхнего угла  $i$ -го окна, а  $(x_{(2,i)}, y_{(2,i)})$  — 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

<code>windows.in</code>	<code>windows.out</code>
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
5 0 0 1 1 0 1 1 2 0 0 2 2 1 0 2 1 1 1 2 2	5 1 1

## Задача D. 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 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.

### 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 Е. Перестановки

Имя входного файла: permutation.in  
Имя выходного файла: permutation.out  
Ограничение по времени: 1.5 секунды  
Ограничение по памяти: 256 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 —  $1 \leq N \leq 100\,000$  — количество чисел, которые выписал Вася и  $1 \leq M \leq 100\,000$  — 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.

### Примеры

permutation.in	permutation.out
4 2	1
1 2 3 4	3
1 2 2 3	
1 3 1 3	

## Задача F. Друзья и последовательности

Имя входного файла: friends.in  
Имя выходного файла: friends.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Майк и !Майк соперничают еще со школьных лет, они противоположны во всем что делают, кроме программирования. Сегодня у них возникла проблема, которую сами друзья сами решить не могут, но вместе с вами — кто знает?

Каждый из них знает две последовательности  $n$  чисел  $a$  и  $b$ . По запросу в виде пары целых чисел  $(l, r)$  Майк может сразу сообщить значение  $\max_{i=l}^r a_i$ , а !Майк — значение  $\min_{i=l}^r b_i$ .

Предположим, что робот задает им каждый из возможных различных запросов в виде пары целых чисел  $(l, r)$  ( $1 \leq l \leq r \leq n$ ) (то есть он сделает ровно  $n(n+1)/2$  запросов) и считает, сколько раз их ответы на один и тот же запрос совпадают, то есть для скольких пар выполняется  $\max_{i=l}^r a_i = \min_{i=l}^r b_i$ .

Сколько случаев совпадения посчитает робот?

### Формат входных данных

В первой строке содержится единственное целое число  $n$  ( $1 \leq n \leq 200\,000$ ).

Во второй строке содержатся  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $-10^9 \leq a_i \leq 10^9$ ) — элементы последовательности  $a$ .

В третьей строке содержатся  $n$  целых чисел  $b_1, b_2, \dots, b_n$  ( $-10^9 \leq b_i \leq 10^9$ ) — элементы последовательности  $b$ .

### Формат выходных данных

Выведите одно целое число — количество совпадений ответов, которые посчитает робот, то есть для скольких пар выполняется  $\max_{i=l}^r a_i = \min_{i=l}^r b_i$ .

### Примеры

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

## Задача G. Рогатка

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

Фермер Джон не любит возить навоз. Он придумал перемещать лотки с навозом по воздуху с помощью гигантской рогатки.

Ферма Джона построена вдоль прямой дороги, поэтому любое место фермы может быть описано позицией на этой дороге (точка на числовой прямой). ФД построил  $N$  рогаток,  $i$ -я из которых описывается тремя целыми числами  $x_i, y_i, t_i$ , указывающими, что рогатка может переместить навоз из позиции  $x_i$  в позицию  $y_i$  за  $t_i$  единиц времени.

У ФД есть  $M$  лотков с навозом для транспортировки.  $j$ -й лоток нужно переместить из позиции  $a_j$  в позицию  $b_j$ . Перемещение лотка с навозом на тракторе на расстояние  $d$  занимает  $d$  единиц времени. ФД надеется сократить время использованием рогаток. Время перемещения трактора без навоза не учитывается.

Для каждого из  $M$  лотков определите минимально возможное время транспортировки, при условии использования не более одной рогатки.

### Формат входных данных

Первая строка ввода содержит  $N$  и  $M$  ( $1 \leq N \leq 100\,000, 1 \leq M \leq 100\,000$ ). Каждая из следующих  $N$  строк описывает одну рогатку тремя числами  $x_i, y_i, t_i$  ( $0 \leq x_i, y_i, t_i \leq 10^9$ ). Последние  $M$  строк описывают лотки навоза, которые необходимо перемещать, двумя целыми числами  $a_j$  и  $b_j$  ( $0 \leq a_j, b_j \leq 10^9$ ).

### Формат выходных данных

Выведите  $M$  строк, по одной для каждого лотка с навозом, содержащих минимальное время для транспортировки соответствующего лотка с навозом.

### Пример

стандартный ввод	стандартный вывод
2 3	4
0 10 1	3
13 8 2	10
1 12	
5 2	
20 7	

### Замечание

Здесь первый лоток необходимо переместить из позиции 1 в позицию 12. Без использования рогаток это займёт 11 единиц времени. Однако, если использовать первую рогатку, это займёт 1 единицу времени, чтобы переместить лоток из позиции 1 в позицию 0 (исходная точка первой рогатки), 1 единицу времени перебросить по воздуху в позицию 10 (финишная точка первой рогатки) и затем 2 единицы времени переместить на тракторе в позицию 12. Второй лоток навоза выгоднее перемещать без рогатки. Третий лоток навоза выгоднее всего перемещать с помощью второй рогатки.