



Stars. Звезды

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

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 записью.

Количество записей во входном файле не больше 100 002.

Формат выходного файла

Для каждого Петингого вопроса выведите на отдельной строке одно число — искомое количество звезд.

Пример

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	

RVQ. Range Variation Query

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

В начальный момент времени последовательность a_n задана следующей формулой:
 $a_n = n^2 \bmod 12345 + n^3 \bmod 23456$.

Требуется много раз отвечать на запросы следующего вида:

- найти разность между максимальным и минимальным значением среди элементов a_i, a_{i+1}, \dots, a_j ;
- присвоить элементу a_i значение j .

Формат входного файла

Первая строка входного файла содержит натуральное число k — количество запросов ($k \leq 100\,000$). Следующие k строк содержат запросы, по одному на строке. Запрос номер i описывается двумя целыми числами x_i, y_i .

Если $x_i > 0$, то требуется найти разность между максимальным и минимальным значением среди элементов a_{x_i}, \dots, a_{y_i} . При этом $1 \leq x_i \leq y_i \leq 100\,000$.

Если $x_i < 0$, то требуется присвоить элементу $a_{|x_i|}$ значение y_i . При этом $-100\,000 \leq x_i \leq -1$ и $|y_i| \leq 100\,000$.

Формат выходного файла

Для каждого запроса первого типа в выходной файл требуется вывести одну строку, содержащую разность между максимальным и минимальным значением на соответствующем отрезке.

Пример

rvq.in	rvq.out
7	34
1 3	68
2 4	250
-2 -100	234
1 5	1
8 9	
-3 -101	
2 3	



Windows. Окна

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

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

Формат входного файла

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 $2 \cdot 10^5$.

Формат выходного файла

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

Пример

windows.in	windows.out
2 0 0 3 3 1 1 4 4	2 3 2

Sum2. И снова сумма...

Имя входного файла: `sum2.in`
Имя выходного файла: `sum2.out`

Реализуйте структуру данных, которая поддерживает множество S целых чисел, с которым разрешается производить следующие операции:

- $add(i)$ — добавить в множество S число i (если оно там уже есть, то множество не меняется);
- $sum(l, r)$ — вывести сумму всех элементов x из S , которые удовлетворяют неравенству $l \leq x \leq r$.

Формат входного файла

Изначально множество S пусто. Первая строка входного файла содержит n — количество операций ($1 \leq n \leq 300\,000$). Следующие n строк содержат операции. Каждая операция имеет вид либо «+ i », либо «? l r ». Операция «? l r » задает запрос $sum(l, r)$.

Если операция «+ i » идет во входном файле в начале или после другой операции «+», то она задает операцию $add(i)$. Если же она идет после запроса «?», и результат этого запроса был y , то выполняется операция $add((i + y) \bmod 10^9)$.

Во всех запросах и операциях добавления параметры целые и лежат в интервале от 0 до 10^9 .

Формат выходного файла

Для каждого запроса выведите одно число — ответ на запрос.

Пример

sum2.in	sum2.out
6 + 1 + 3 + 3 ? 2 4 + 1 ? 2 4	3 7



Movetofront. Вперёд!

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

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 1, 3, 6, 2, 5, 4, то после приказа «Рядовые с 2 по 3 — 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.

Пример

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

Step. Шаги

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

Саша и Вася начали заниматься чечеткой. Этот танец состоит из притопывания ногой по полу. Но так как они очень быстро учатся, они решили поэкспериментировать с хореографией.

Хореография чечетки описывается последовательностью из двух букв 'L' и 'R'. 'L' означает притопывание левой ногой, а 'R', соответственно, правой. Саша понял, что самая лучшая часть этого танца это та, в которой не используется одна и та же нога подряд. Он определил значение хореографии как наибольшую непрерывную последовательность, в которой нет двух одинаковых последовательных символов.

Как известно, создание красивого танца — очень непростой процесс, с множеством маленьких изменений, прежде чем оптимальный вариант будет найден. Поэтому Вася хочет знать значение хореографии после каждого изменения. Изменение — это замена L на R (или наоборот) в какой-то позиции.

До всех изменений последовательность состоит только из одних букв L.

Формат входного файла

На первой строке задано количество два целых числа: длина хореографии N ($1 \leq N \leq 200\,000$) и число изменений Q ($1 \leq Q \leq 200\,000$).

Следующие Q строк содержат по одному числу, определяющего в какой позиции происходит замена.

Формат выходного файла

Вы должны вывести Q чисел, по одному на каждой строке: значения хореографии после каждого изменения.

Пример

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



Elephants. Слоники

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

Танцующие слоны — это зрелищное шоу в Судиславле, в котором участвуют N слонов, танцующих на одной линии, называемой сценой.

В результате многолетних тренировок слоны, участвующие в шоу, разучили большое количество танцевальных движений. Все шоу состоит из последовательности актов. В каждом акте только один слон совершает одно танцевальное движение, в результате которого он может переместиться на другую позицию на сцене.

Постановщики шоу хотят сделать фотоальбом, который бы содержал фотографии всего шоу. После каждого акта они хотят сделать фотографии всех слонов.

В любой момент времени на протяжении шоу некоторое количество слонов может находиться в одной и той же позиции — это значит, что слоны просто стоят рядом. Одна фотокамера может фотографировать группу слонов тогда и только тогда, когда все позиции, в которых находятся слоны, лежат на отрезке длины L (обе границы отрезка включаются в него). Так как слоны могут располагаться вдоль всей сцены, то может потребоваться несколько фотокамер, чтобы сфотографировать всех слонов одновременно.

Вы же должны определить для каждого акта минимальное количество фотокамер, которые нужны, чтобы сфотографировать всех слонов после него.

Формат входного файла

Первая строка входного файла содержит три целых неотрицательных числа N, L и M ($0 \leq N \leq 15 \cdot 10^4$, $0 \leq L \leq 10^9$, $0 \leq M \leq 15 \cdot 10^4$). Далее следуют N строк по одному числу x_i в каждой — позиция каждого слона ($0 \leq x_i \leq 10^9$). Гарантируется, что все входные позиции упорядочены. Следует заметить, что в течение танца может меняться порядок слонов. Далее следует M строк. В каждой из них содержится два числа i и y , означающих, что в соответствующем акте слон с номером i перемещается в позицию с координатами y . Гарантируется, что y удовлетворяет ограничению: $0 \leq y \leq 10^9$

Формат выходного файла

Выведите M строк по одному числу в каждой: минимальное количество фотокамер, необходимых, чтобы сфотографировать всех слонов после него.

Пример

elephants.in	elephants.out
4 10 5	1
10	2
15	2
17	2
20	3
2 16	
1 25	
3 35	
0 38	
2 0	

rect2. Прямоугольники - 2

Имя входного файла: `rect2.in`
Имя выходного файла: `rect2.out`

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 вверху. 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 -ый 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 её 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 входного файла

В первой строке входного файла записано n — количество отрезков, на которые разделена ось OX ($1 \leq n \leq 50000$). 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 ($1 \leq m \leq 100000$). 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 выходного файла

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

rect2.in	rect2.out
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	

Upit. Вопрос

Имя входного файла: upit.in
Имя выходного файла: upit.out

Вова задолбался писать всевозможные структуры данных в различных задачах и решил написать одну структуру данных, которая сможет оперировать с его любой последовательностью чисел. Помогите ему!

Вова предоставит вам его числовую последовательность и последовательность запросов к структуре. Каждый запрос либо запрашивает информацию, либо изменяет существующую последовательность. Возможные запросы указаны ниже.

Тип запроса	Описание	Пример
1 A B X	Присвоить значение X всем элементам с A по B включительно	(9, 8, 7, 6, 5, 4, 3, 2, 1) \rightarrow 1 3 5 0 \rightarrow (9, 8, 0, 0, 0, 4, 3, 2, 1)
2 A B X	Прибавить X к A -тому элементу, $2X$ к $A + 1$ -му, ..., и $(B - A + 1) * X$ к B -тому элементу	(9, 8, 7, 6, 5, 4, 3, 2, 1) \rightarrow 2 3 5 2 \rightarrow (9, 8, 9, 10, 11, 4, 3, 2, 1)
3 C X	Вставить новый элемент со значением X прямо перед элементом с номером C	(9, 8, 7, 6, 5, 4, 3, 2, 1) \rightarrow 3 4 100 \rightarrow (9, 8, 7, 100, 6, 5, 4, 3, 2, 1)
4 A B	Найти сумму с A по B включительно	(2, 18, 7, 6, 1, 4, 7, 7, 2) \rightarrow 4 6 7 \rightarrow 11

Формат входного файла

Первая строка содержит два числа N и Q ($1 \leq N, Q \leq 100000$), длина начальной последовательности и число запросов. Следующая строка содержит неотрицательные числа не превосходящие 100000.

Следующие Q строк содержат запросы в формате, описанном выше. Во всех запросах $1 \leq X \leq 100$, $1 \leq A \leq B \leq \text{currentSequenceLength}$, $1 \leq C \leq \text{currentSequenceLength} + 1$.

Формат выходного файла

Для каждого запроса четвертого типа выведите получившуюся сумму.

Пример

upit.in	upit.out
5 5	4
1 2 3 4 5	0
1 5 5 0	25
4 4 5	
4 5 5	
2 1 5 1	
4 1 5	
1 7	17
100	27
3 1 17	100
3 2 27	37
3 4 37	
4 1 1	
4 2 2	
4 3 3	
4 4 4	



Matrix. Матрица

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

Дана матрица целых чисел размера $n \times n$. Требуется выбрать в этой матрице n элементов, никакие два из которых не расположены в одной строке или одном столбце, минимизировав сумму выбранных элементов.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($1 \leq n \leq 239$). Следующие n строк содержат по n целых чисел в каждой. Все эти числа не превосходят по абсолютной величине 10^6 .

Формат выходного файла

Первая строка должна содержать значение минимизируемой функции. В следующие n строк необходимо записать пары чисел, описывающих выбранные ячейки. Первой координатой выводится номер строки.

Пример

matrix.in	matrix.out
2	2
1 1	1 1
1 1	2 2