

Задача А. Наивный RMQ

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

Вам дан массив $a[]$ из n элементов и m – количество запросов, состоящих из двух чисел l и r , выведите минимальный элемент в массиве $a[]$ с индексами от l до r .

Формат входного файла

В первой строке входного файла содержится число n – длина массива $a[]$ ($1 \leq n \leq 1000$). Во второй строке содержится n чисел a_i – элементы массива ($-10^9 \leq a_i \leq 10^9$). В следующей строке содержится число m – количество запросов ($1 \leq m \leq 1000$). В последних m строках содержатся запросы l, r ($1 \leq l, r \leq n$).

Формат выходного файла

Для каждого запроса выведите ответ в новой строке.

Примеры

<code>stupid_rmq.in</code>	<code>stupid_rmq.out</code>
5	-2
1 -2 3 4 1	3
3	-2
1 3	
3 4	
1 5	
2	-100000
-100000 -1111111	-1111111
2	
1 1	
1 2	

Задача D. Сумма 2

Имя входного файла: `sum2.in`
Имя выходного файла: `sum2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам дан массив $a[]$ из n элементов и m – количество запросов, состоящих из двух чисел l и r , выведите сумму элементов в массиве $a[]$ с индексами от l до r .

Формат входного файла

В первой строке входного файла содержится число n – длина массива $a[]$ ($1 \leq n \leq 100000$). Во второй строке содержится n чисел a_i – элементы массива ($-10^9 \leq a_i \leq 10^9$). В следующей строке содержится число m – количество запросов ($1 \leq m \leq 100000$). В последних m строках содержатся запросы l, r ($1 \leq l, r \leq n$).

Формат выходного файла

Для каждого запроса выведите ответ в новой строке.

Примеры

<code>sum2.in</code>	<code>sum2.out</code>
5 1 2 3 4 5 3 1 5 1 1 4 5	15 1 9
6 -1 1 -1 1 -1 1 2 1 6 3 3	0 -1

Задача А. Сумма на отрезке

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

Дан массив из N элементов, нужно научиться находить сумму чисел на отрезке.

Формат входного файла

Первая строка входного файла содержит два целых числа N и K — число чисел в массиве и количество запросов. ($1 \leq N \leq 100\,000$), ($0 \leq K \leq 100\,000$). Следующие K строк содержат запросы

1. `A i x` — присвоить i -му элементу массива значение x ($1 \leq i \leq n$, $0 \leq x \leq 10^9$)
2. `Q l r` — найти сумму чисел в массиве на позициях от l до r . ($1 \leq l \leq r \leq n$)

Изначально в массиве живут нули.

Формат выходного файла

На каждый запрос вида `Q l r` нужно вывести единственное число — сумму на отрезке.

Примеры

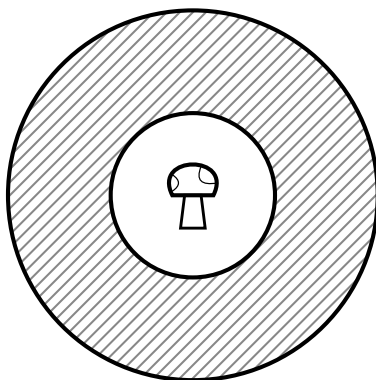
<code>sum.in</code>	<code>sum.out</code>
5 9	0
A 2 2	2
A 3 1	1
A 4 2	2
Q 1 1	0
Q 2 2	5
Q 3 3	
Q 4 4	
Q 5 5	
Q 1 5	

Задача В. Грибы

Имя входного файла:	<code>mushrooms.in</code>
Имя выходного файла:	<code>mushrooms.out</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В очень большом лесу раскинулась огромная поляна. После проливного дождя на поляне последовательно вырастают грибы в течении n минут. На первой минуте в центре поляны вырастает гриб высотой h_1 сантиметров. На второй минуте на расстоянии 1 метр вокруг первого гриба вырастают грибы высотой h_2 . На n -ой минуте на расстоянии 1 метр вокруг уже существующих грибов вырастают грибы высотой h_n . Грибы растут кольцами.

Разумеется, что такой урожай не мог остаться незамеченным. Поэтому была сформирована Регистрационная Единая Государственная Грибная Инспекция, которая занимается исследованием высоты грибов выросших после дождя. А именно, инспекцию интересует, какой самый высокий гриб находится в кольце от r до R . Будем называть кольцом область заштрихованную на рисунке (включая границы). Причем r - радиус меньшей окружности, а R - радиус большей окружности, центры окружностей находятся в точке, где вырос первый гриб.



Так как поляна может быть достаточно большой, то инспекция должна проверить несколько колец. Причем, если на $i - 1$ -ой проверке было выявлено, что самый высокий гриб имеет высоту H (в кольце от r до R), то на i -ой проверке необходимо выявить гриб в кольце от $\min(R, (r \cdot H + H^2) \bmod n)$ до $\max(R, (r \cdot H + H^2) \bmod n)$. Для отчета инспекция хочет узнать только сумму высот грибов полученных, в результате всех проверок.

Формат входного файла

В первой строке входного файла задано целое число n ($1 \leq n \leq 2 \cdot 10^5$) – количество минут в течении которых вырастают грибы. Во второй строке задано n целых чисел h_i – высота грибов, выросших на i минуте ($1 \leq h_i \leq 10^9$). В третьей строке задано число m ($1 \leq m \leq 5 \cdot 10^6$) – количество проверок. Последняя строка содержит два целых числа r и R ($0 \leq r \leq R < n$) – описание первой проверки.

Формат выходного файла

Выведите ответ на задачу – сумму высот грибов, полученных в результате всех проверок.

Пример

mushrooms.in	mushrooms.out
5 1 2 3 4 5 3 1 4	15
7 1 1 4 2 3 5 1 1 0 6	5

Задача В. Запрос изменения на отрезке

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

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	

Задача А. Это как пятый, но на один пониже

Имя входного файла: floor4.in
Имя выходного файла: floor4.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Знаете ли вы, почему четвертый этаж заперт и там не останавливается лифт? Потому что на самом деле четвертый, запертый, этаж, где не останавливается лифт, содержит бесконечное количество комнат, пронумерованных натуральными числами. На этот этаж регулярно приезжают дети, каждый из которых заранее выбрал, в какую комнату он хочет заселиться. Если выбранная комната оказывается свободна, то ребенок занимает ее, в противном случае он занимает первую свободную комнату с большим номером.

Кроме того, некоторые дети уезжают в середине смены. Сразу после отъезда ребенка его комната становится доступна для заселения следующего.

Промоделируйте работу преподавателей, ответственных за четвертый этаж, и научитесь быстро сообщать приезжающим детям, какую комнату им следует занимать.

Формат входного файла

Первая строка входного файла содержит натуральное число n — количество прибытий и отъездов, происходящих в течение смены ($n \leq 100\,000$).

Следующие n строк содержат информацию об детях. Число $a > 0$ обозначает, что приехал школьник, желающий занять комнату номер a ($a \leq 100\,000$). Число $a < 0$ обозначает, что с комнаты номер $|a|$ уехал школьник. (Гарантируется, что эта комната не была пуста).

Формат выходного файла

Для каждого приезжающего школьника выведите одно натуральное число — номер комнаты, в которую он поселится.

Пример

floor4.in	floor4.out
6	5
5	6
5	7
5	6
-6	8
5	
5	

Задача С. Сумма

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

Дан массив из N элементов, нужно научиться находить сумму чисел на отрезке.

Формат входного файла

Первая строка входного файла содержит два целых числа N и K — число чисел в массиве и количество запросов. ($1 \leq N \leq 100\,000$), ($0 \leq K \leq 100\,000$). Следующие K строк содержат запросы:

1. `A l r x` — присвоить элементам массива с позициями от l до r значение x ($1 \leq l \leq r \leq n$, $0 \leq x \leq 10^9$)
2. `Q l r` — найти сумму чисел в массиве на позициях от l до r . ($1 \leq l \leq r \leq n$)

Изначально массив заполнен нулями.

Формат выходного файла

На каждый запрос вида `Q l r` нужно вывести единственное число — сумму на отрезке.

Примеры

<code>sum.in</code>	<code>sum.out</code>
5 9	3
A 2 3 2	2
A 3 5 1	3
A 4 5 2	4
Q 1 3	2
Q 2 2	7
Q 3 4	
Q 4 5	
Q 5 5	
Q 1 5	

Задача В. Большая стена

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

Жители страны X поссорились с жителями страны Y . В результате конфликта правительство страны X решило построить стену на протяжении всей границы со страной Y (граница имеет форму отрезка с началом в точке 1 и концом в точке n).

Правительством страны X был тут же составлен план строительства стены. В момент времени i планируется поставить на участок границы от точки l_i до r_i включительно стройматериалов достаточных для увеличения стены на высоту a_i .

Конечно же правительство страны X хочет контролировать процесс возведения стены. Поэтому в плане строительства присутствуют проверки чиновников, в некоторый момент времени j чиновники приезжают на границу и замеряют максимальную высоту части стены от точки l_j до точки r_j , о результатах они должны немедленно сообщить в правительство страны X .

Ваша задача - найти ожидаемые результаты проверок, с которыми будет сверяться правительство страны X при получении данных от чиновников.

Формат входного файла

В первой строке входного файла даны два числа n - длина границы и m - количество пунктов в плане ($1 \leq n, m \leq 2 \cdot 10^5$).

В следующих m строках даны описания пунктов плана двух видов:

- *build* l r a - надстройка стены от точки l до r включительно ($1 \leq l \leq r \leq n$) на высоту a ($1 \leq a \leq 10^4$).
- *test* l r - проверка стены от точки l до r включительно ($1 \leq l \leq r \leq n$)

Формат выходного файла

Для каждого пункта проверки выведите единственное число в новой строке - результат проверки.

Примеры

bigwall.in	bigwall.out
5 6	2
build 1 4 2	3
build 2 5 1	13
test 1 1	
test 1 5	
build 3 4 10	
test 2 3	
3 3	0
test 1 3	1
build 2 2 1	
test 1 3	

Задача С. Художник

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

Не успев дорисовать свой гениальный футуристический шедевр, М. Калевич увлекся рисованием одномерных черно-белых картин. Он пытается найти оптимальное местоположение и количество черных участков картины. Для этого он проводит на прямой белые и черные отрезки, и после каждой из таких операций хочет знать количество черных отрезков на получившейся картине и их суммарную длину.

Изначально прямая — белая. Ваша задача — написать программу, которая после каждой такой операции выводит в выходной файл интересующие художника данные.

Формат входного файла

В первой строке входного файла содержится общее количество нарисованных отрезков ($1 \leq N \leq 100\,000$). В последующих N строках содержится описание операций. Каждая операция описывается строкой вида $s\ x\ l$, где s — цвет отрезка (W для белых отрезков, B для черных), а сам отрезок имеет вид $[x; x+l]$, причем координаты обоих концов — целые числа, не превосходящие по модулю 500 000. Длина задается положительным целым числом.

Формат выходного файла

После выполнения каждой из операций необходимо вывести в выходной файл на отдельной строке количество черных отрезков на картине и их суммарную длину, разделенные одним пробелом.

Пример

<code>painter.in</code>	<code>painter.out</code>
7	0 0
W 2 3	1 2
B 2 2	1 4
B 4 2	1 4
B 3 2	2 6
B 7 2	3 5
W 3 1	0 0
W 0 10	