

Задача А. Сумма на отрезке (12,5 баллов)

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 — количество чисел в массиве и количество запросов ($1 \leq N \leq 100\,000$, $0 \leq K \leq 100\,000$). Следующие K строк содержат следующие 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. Ромашки (12,5 баллов)

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

Саша любит наблюдать за ромашками. Но следить за всем полем сразу ей тяжело. Поэтому она наблюдает только за его частью, ограниченной прямоугольником размером $x \times y$. Этот прямоугольник поделён на маленькие прямоугольники размером 1×1 . Во время её наблюдений могут происходить следующие события:

1. в каком-то прямоугольнике появляются или исчезают несколько ромашек;

2. к ней может заглянуть её подруга Настя и поинтересоваться, сколько видно ромашек в части поля, состоящей из нескольких прямоугольников.

Формат входного файла

Первая строка входного файла содержит два натуральных числа x и y ($1 \leq x, y \leq 1\,000$). Следующая строка содержит единственное натуральное число n ($1 \leq n \leq 100\,000$) — количество произошедших событий. Далее следуют n записей о происшедших событиях по одной в строке. В начале строки записано число t . Если t равно:

- 1, то за ним следуют 3 числа: x_1, y_1 ($1 \leq x_1 \leq x, 1 \leq y_1 \leq y$) и k ($-10\,000 \leq k \leq 10\,000$) — координаты прямоугольника и величина, на которую в нём изменилось количество видимых ромашек;
- 2, то за ним следуют 4 числа: x_1, y_1, x_2, y_2 ($1 \leq x_1 \leq x_2 \leq x, 1 \leq y_1 \leq y_2 \leq y$), которые означают, что Настя попросила подсчитать количество ромашек в прямоугольниках (x', y') из области: $x_1 \leq x' \leq x_2, y_1 \leq y' \leq y_2$.

Формат выходного файла

Для каждого Настиного вопроса выведите искомое количество ромашек (*оно может быть отрицательным*).

Примеры

<code>camomiles.in</code>	<code>camomiles.out</code>
8 8	3
3	
1 2 2 2	
1 1 1 1	
2 1 1 2 2	

Задача С. Минимум и максимум (8 баллов)

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

Пусть есть множество целых чисел. Необходимо реализовать структуру данных для их хранения, поддерживающую следующие операции: `GetMin` — извлечение минимума, `GetMax` — извлечение максимума, `Insert(N)` — добавление числа в множество.

Формат входного файла

В первой строке входного файла записано одно целое число N ($1 \leq N \leq 100\,000$) — число запросов к структуре. Затем в N строках следуют запросы по одному в строке: `GetMin`, `GetMax`, `Insert(A)` — извлечение минимума, максимума и добавление числа A ($1 \leq A \leq 2^{31} - 1$). Запросы корректны, то есть нет операций извлечения для пустого множества.

Формат выходного файла

Для каждого запроса `GetMin` или `GetMax` выведите то число, которое было извлечено.

Примеры

minmax.in	minmax.out
10	1
Insert(100)	100
Insert(99)	1
Insert(1)	2
Insert(2)	99
GetMin	
GetMax	
Insert(1)	
GetMin	
GetMin	
GetMax	

Задача D. Мега-инверсии (8 баллов)

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

Инверсией в перестановке p_1, p_2, \dots, p_N называется пара (i, j) такая, что $i < j$ и $p_i > p_j$. Назовём *мега-инверсией* в перестановке p_1, p_2, \dots, p_N тройку (i, j, k) такую, что $i < j < k$ и $p_i > p_j > p_k$. Напишите алгоритм для быстрого подсчёта количества мега-инверсий в перестановке.

Формат входного файла

Первая строка входного файла содержит целое число N ($1 \leq N \leq 100\,000$). Следующие N чисел описывают перестановку: p_1, p_2, \dots, p_N ($1 \leq p_i \leq N$), все p_i попарно различны. Числа разделяются переводами строк.

Формат выходного файла

Единственная строка выходного файла должна содержать одно число, равное количеству мега-инверсий в перестановке p_1, p_2, \dots, p_N .

Примеры

mega.in	mega.out
4	4
4	
3	
2	
1	

Задача E. Художник (9 баллов)

Имя входного файла: `painter.in`
 Имя выходного файла: `painter.out`
 Ограничение по времени: 4 секунды
 Ограничение по памяти: 64 мегабайта

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

Изначально прямая белая. Ваша задача — написать программу, которая после каждой такой операции выводит в выходной файл интересные 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 и 'B' для чёрных), а сам отрезок имеет вид $[x; x + l]$, причём координаты обоих концов — целые числа, по модулю не превосходящие 500 000. Длина задаётся положительным целым числом.

Формат выходного файла

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

Примеры

painter.in	painter.out
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	