

Задача А. Художник

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Итальянский художник-абстракционист Ф. Мандарино 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 из таких операций выводит в выходной файл интересующие художника данные.

Формат входных данных

В первой строке входного файла содержится общее количество нарисованных отрезков ($1 \leq N \leq 10^5$). В последующих N строках содержится описание операций. Каждая операция описывается строкой вида $c\ x\ l$, где c — 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 $5 \cdot 10^5$. Длина задается положительным целым числом

Формат выходных данных

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

стандартный ввод	стандартный вывод
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	

Задача В. Отрезки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны n отрезков на прямой. Пара отрезков называется вложенной, если отрезки не совпадают, и один отрезок содержит второй. Посчитать количество пар вложенных отрезков.

Формат входных данных

Целое число n ($1 \leq n \leq 3 \cdot 10^5$) и n пар целых чисел $0 \leq l_i \leq r_i \leq 10^9$.

Формат выходных данных

Одно число — количество пар вложенных отрезков.

Пример

стандартный ввод	стандартный вывод
4 1 4 2 5 1 3 3 4	3

Задача С. Окна

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
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 ($1 \leq n \leq 5 \cdot 10^4$).

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 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, разделенные пробелом — координаты точки, покрытой максимальным числом окон. Окна считаются замкнутыми, т.е. покрывающими свои граничные точки.

Пример

стандартный ввод	стандартный вывод
2	2
0 0 3 3	1 3
1 1 4 4	

Задача D. Прямоугольники

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

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 сверху. 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 -й 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 e_j 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 входных данных

В первой строке входного файла записано n — количество отрезков, на которые разделена ось OX ($1 \leq n \leq 5 \cdot 10^4$).

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 ($1 \leq m \leq 10^5$).

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 выходных данных

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

стандартный ввод	стандартный вывод
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	

Задача Е. Перестановки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 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 — $1 \leq N \leq 10^5$ — количество чисел, которые выписал Вася и $1 \leq M \leq 10^5$ — 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.

Пример

стандартный ввод	стандартный вывод
4 2	1
1 2 3 4	3
1 2 2 3	
1 3 1 3	

Задача F. Минимум в прямоугольнике

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 6 секунд
Ограничение по памяти: 256 мегабайт

Вам дана таблица из $n \cdot m$ целых чисел и последовательность из q запросов $(x_{i_1}, y_{i_1}, x_{i_2}, y_{i_2})$. Для каждого запроса найдите минимальное значение на прямоугольнике, состоящем из чисел a_{xy} по всем (x, y) , удовлетворяющим условиям $x_{i_1} \leq x \leq x_{i_2}$ и $y_{i_1} \leq y \leq y_{i_2}$

Формат входных данных

Первая строка входного файла содержит числа m и n — размеры таблицы ($1 \leq m, n \leq 500$). Следующие m строк содержат по n целых чисел каждая — значения таблицы. Все значения лежат в диапазоне значений типа данных `int` для C++.

Далее следует число q — количество запросов ($1 \leq q \leq 2 \cdot 10^5$). Следующие q строк содержат по четыре целых числа: $x_{i_1}, y_{i_1}, x_{i_2}, y_{i_2}$ ($1 \leq x_{i_1} \leq x_{i_2} \leq m, 1 \leq y_{i_1} \leq y_{i_2} \leq n$).

Формат выходных данных

Выведите q чисел — ответы на запросы в порядке их следования во входном файле.

Пример

стандартный ввод	стандартный вывод
1 1	-2147483648
-2147483648	
1	
1 1 1 1	

Задача G. Simple 2D

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Дан набор точек на плоскости. Напишите структуру данных, отвечающую на запросы «количество точек в прямоугольнике».

Формат входных данных

В первой строке идёт целое число N ($N \leq 10^5$) — количество точек.

В последующих N строках идут координаты точек в формате (x_i, y_i) — числа, не превосходящие по модулю 10^9 .

Далее идёт целое число M ($M \leq 10^5$) — количество запросов.

В последующих M строках идут запросы в формате $(x_{1j}, y_{1j}, x_{2j}, y_{2j})$ — числа, не превосходящие по модулю 10^9 , задающие координаты нижнего-левого и верхнего-правого угла j -ого прямоугольника-запроса. Стороны считаем принадлежащими многоугольнику.

Формат выходных данных

Для каждого запроса выведите единственное целое число — количество точек, попавших внутрь прямоугольника.

Пример

стандартный ввод	стандартный вывод
5	3
-1 -1	5
-1 0	0
0 1	1
0 -1	1
1 -1	
5	
0 -3 3 1	
-1 -1 1 1	
3 4 5 6	
-1 -1 -1 -1	
0 0 0 5	

Задача Н. Эффективное тестирование

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Начиная с 20xx года все организаторы всех школьных олимпиад по программированию договорились проводить соревнования исключительно по интернету, для чего было создано общество с ограниченной ответственностью «Организация онлайн-олимпиад» (ООО «ООО»). Разумеется, такая серьёзная организация не может обойтись без собственной тестирующей системы, поэтому для её создания были наняты эффективные менеджеры, закуплены доски и подготовлена синяя изолента.

Для повышения эффективности процесса тестирования была разработана следующая архитектура. Сначала все m тестов задачи располагаются в порядке от 1 к m в очереди тестирования. Затем модуль планирования последовательно выполняет n действий. Действие i состоит в том, чтобы выбрать отрезок очереди с позиции l_i по r_i включительно (в нумерации с единицы) и проверить решение на каждом втором тесте на этом отрезке, а именно на тестах на позициях $l_i, l_i+2, l_i+4, \dots, r_i$ очереди (при этом гарантируется, что l_i и r_i имеют одинаковую чётность). После этого те тесты, на которых было проведено тестирование, удаляются из очереди, а все оставшиеся тесты сдвигаются по очереди таким образом, чтобы пустых мест не осталось. Например, если в очереди находились тесты с исходными номерами 2, 3, 4, 5, 10, 12, 13, 20 и была применена операция с $l_i = 3, r_i = 7$, то посылка будет протестирована на тестах с позиций 3, 5 и 7, которые исходно имели номера 4, 10 и 13. После выполнения данной операции очередь тестирования будет состоять из тестов с исходными номерами 2, 3, 5, 12, 20.

Вам поручено реализовать модуль, который для каждого из n описанных выше действий будет определять минимальный и максимальный номер теста в изначальной нумерации из тех, на которых на этом шаге проверялось решение.

Формат входных данных

В первой строке входных данных находятся два числа n и m ($1 \leq n \leq 100\,000, 1 \leq m \leq 10^{18}$) — количество действий модуля планирования и количество тестов в задаче.

В каждой из последующих n строк записаны два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq m$) — параметры i -го действия модуля планирования. Гарантируется, что перед началом выполнения действия i в очереди тестирования находятся хотя бы r_i тестов и что числа l_i и r_i имеют одинаковую чётность.

Формат выходных данных

Для каждого из n действий модуля планирования выведите два целых числа — минимальный и максимальный номер теста в исходной нумерации из тех, на которых проверялось решение на соответствующем шаге.

Примеры

стандартный ввод	стандартный вывод
2 10	2 8
2 8	1 5
1 3	
4 6	1 1
1 1	2 2
1 1	3 3
1 1	5 5
2 2	

Замечание

Рассмотрим, как изменяется очередь тестирования в первом примере.

1. Изначально в очереди тестирования находятся все тесты от 1 до 10, то есть очередь имеет вид 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

2. При выполнении первого запроса будут удалены тесты 2, 4, 6, 8, и очередь примет вид 1, 3, 5, 7, 9, 10.
3. При выполнении второго запроса будут удалены тесты 1 и 5, очередь примет вид 3, 7, 9, 10.