Задача 1.

Сложная

**Перестановки**

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.

ВХОДНЫЕ ДАННЫЕ

В первой строке лежит два натуральных числа -— 1≤N≤100000 -— количество чисел, которые выписал Вася и  
1≤M≤100000 -— 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≤x≤y≤N и  
1≤k≤l≤N.

ВЫХОДНЫЕ ДАННЫЕ

Выведите M строк, каждая должна содержать единственное число~-— ответ на Васин вопрос.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 4 2 1 2 3 4 1 2 2 3 1 3 1 3 | 1 3 |

Задача 2.

Сложная

**Окна**

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

ВХОДНЫЕ ДАННЫЕ

В первой строке входного файла записано число окон n (1≤n≤50000). 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⋅105.

ВЫХОДНЫЕ ДАННЫЕ

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.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 2 0 0 3 3 1 1 4 4 | 2 3 2 |

Задача 3.

Сложная

**Различные числа**

Сколько различных чисел на отрезке массива?

ВХОДНЫЕ ДАННЫЕ

На первой строке длина массива n (1≤n≤300000).  
На второй строке n целых чисел от 0 до 109−1.  
На третьей строке количество запросов q (1≤q≤300000).  
Следующие q строк содержат описание запросов, по одному на строке.  
Каждый запрос задаётся парой целых чисел l, r (1≤l≤r≤n).

ВХОДНЫЕ ДАННЫЕ

Выведите ответы на запросы по одному в строке.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 5 1 1 2 1 3 3 1 5 2 4 3 5 | 3 2 3 |

Задача 4.

Средняя

**Точки на плоскости**

Есть квадратная клетчатая плоскость состоящая из n×n клеток (1≤n≤1000).  
Изначально в каждой клетке записано значение ноль.  
Ваша задача -— написать программу, умеющую отвечать на следующие запросы:  
ADD x y -— увеличить значение в ячейке x,y на 1.  
GET x1 y1 x2 y2 -— вернуть сумму значений в прямоугольнике с углами в x1,y1 и x2,y2 соответственно.

ВХОДНЫЕ ДАННЫЕ

В первой строке входного файла содержится два числа -— n и k -— размер доски и число запросов соответственно.  
Следующие k строк содержат сами запросы.  
Гарантируется, что общее число запросов не превосходит 300000.

ВЫХОДНЫЕ ДАННЫЕ

Для каждого запроса типа GET выведите в отдельную строку одно целое число -—  
ответ на соответствующий запрос.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 5 15 ADD 1 1 ADD 2 2 ADD 3 3 ADD 4 4 ADD 5 5 ADD 1 5 ADD 2 4 ADD 3 3 ADD 4 2 ADD 5 1 GET 1 1 5 5 GET 2 1 5 5 GET 1 2 5 5 GET 2 2 4 4 GET 3 3 3 3 | 10 8 8 6 2 |

Начало формы

Задача 5.

Олимпиадная

**Connect and Disconnect**

Вы когда-нибудь слышали про обход в глубину?  
Например, используя этот алгоритм, вы можете проверить является ли  
граф связным за время O(E). Вы можете даже посчитать количество  
компонент связности за то же время. А вы когда нибудь слышали про систему непересекающихся множеств?  
Используя эту структуру, вы можете быстро обрабатывать запросы "Добавить ребро в граф" и  
"Посчитать количество компонент связности в графе". А вы когда-нибудь слышали о динамической задаче связности?  
В этой задаче вам необходимо обрабатывать три типа запросов:  
  
Добавить ребро в граф.  
Удалить ребро из графа.  
Посчитать количество компонент связности в графе.

Граф является неориентированным. Изначально граф пустой.

ВХОДНЫЕ ДАННЫЕ

В первой строке находятся два целых числа N и K - количество вершин и количество  
запросов, соответственно (1≤N≤300000, 0≤K≤300000).  
Следующие K строк содержат запросы, по одному в строке.  
Каждый запрос имеет один из трех типов:  
  
+uv: Добавить ребро между вершинами u и v.  
Гарантируется, что такого ребра нет.  
−uv: Удалить ребро между u и v.  
Гарантируется, что такое ребро есть.  
?: Посчитать количество компонент связности в графе.  
  
Вершины пронумерованы целыми числами от 1 до N. Во всех запросах u≠v.

ВЫХОДНЫЕ ДАННЫЕ

Для каждого запроса ? выведите количество компонент связности в момент запроса.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 5 11 ? + 1 2 + 2 3 + 3 4 + 4 5 + 5 1 ? - 2 3 ? - 4 5 ? | 5 1 1 2 |