

## Задача А. Yet another data structure

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

Нужно отвечать на запросы вида

- `+ x` – добавить в мультимножество число  $x$ .
- `? x` – посчитать сумму чисел не больших  $x$ .

### Формат входных данных

Каждая строка файла содержит один запрос.

Все числа  $x$  целые от 0 до  $10^9 - 1$ .

Количество запросов от 1 до  $10^5$ .

### Формат выходных данных

Ответы на все запросы вида “?  $x$ ”.

### Примеры

<code>ds.in</code>	<code>ds.out</code>
<code>+ 1</code>	<code>3</code>
<code>+ 2</code>	<code>3</code>
<code>? 3</code>	<code>1</code>
<code>? 2</code>	<code>0</code>
<code>? 1</code>	<code>2</code>
<code>? 0</code>	
<code>+ 1</code>	
<code>? 1</code>	

## Задача В. Звезды

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

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 входных данных

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 происшедших 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.

Количество записей во входном файле не больше 100 002.

### Формат выходных данных

Для каждого Петиного вопроса выведите искомое количество звезд.

### Примеры

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	

## Задача С. Перестановки

Имя входного файла: `permutation.in`  
Имя выходного файла: `permutation.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 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 —  $1 \leq N \leq 100\,000$  — количество чисел, которые выписал Вася и  $1 \leq M \leq 100\,000$  — 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.

### Примеры

<code>permutation.in</code>	<code>permutation.out</code>
4 2	1
1 2 3 4	3
1 2 2 3	
1 3 1 3	

## Задача D. Перестановки `strike back`

Имя входного файла: `permutation2.in`  
Имя выходного файла: `permutation2.out`  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Вася выписал на доске в каком-то порядке все числа от 1 по  $N$ , каждое число ровно по одному разу. Иногда он стирает какое-то число и записывает на его место другое. Количество чисел, выписанных Васей, оказалось довольно большим, поэтому Вася не может окинуть взглядом все числа. Однако ему надо всё-таки представлять эту последовательность, поэтому он написал программу, которая в любой момент 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 —  $1 \leq N \leq 100\,000$  — количество чисел, которые выписал Вася и  $1 \leq M \leq 100\,000$  — суммарное количество вопросов и изменений сделанных Васей. Во второй строке дано  $N$  чисел — последовательность чисел, выписанных Васей. Далее в  $M$  строках находятся описания вопросов. Каждый запрос на изменение числа в некоторой позиции начинается со слова `SET` и имеет вид `SET a b` ( $1 \leq a \leq N$ ,  $1 \leq b \leq N$ ). Это означает, что Вася изменил число, записанное в позиции  $a$  на число  $b$ . Каждый Васин вопрос начинается со слова `GET` и имеет вид `GET x y k l` ( $1 \leq x \leq y \leq N$ ,  $1 \leq k \leq l \leq N$ ).

### Формат выходных данных

Для каждого Васиного вопроса выведите единственное число — ответ на Васиин вопрос.

### Примеры

<code>permutation2.in</code>	<code>permutation2.out</code>
4 4	1
1 2 3 4	3
GET 1 2 2 3	2
GET 1 3 1 3	
SET 1 4	
GET 1 3 1 3	

## Задача Е. Центроиды дерева

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

Дано дерево из  $n$  вершин. У каждой вершины есть цвет. Нужно обработать  $q$  запросов  $(v_i, c_i)$ : найти расстояние от  $v_i$  до ближайшей к  $v_i$  вершины цвета  $c_i$ . Расстоянием между вершинами называется минимальное количество рёбер в пути между ними.

### Формат входных данных

На первой строке число  $n$  ( $1 \leq n \leq 10^5$ ), следующая строка содержит числа  $p_1, p_2, \dots, p_{n-1}$ .  $0 \leq p_i < i$ .  $p_i$  – отец вершины  $i$  в дереве. Далее строка с числами  $a_0, a_1, \dots, a_{n-1}$ .  $0 \leq a_i < n$ .  $a_i$  – цвет вершины  $i$ . Далее строка с числом  $q$  ( $1 \leq q \leq 10^5$ ). Следующие  $q$  строк содержат запросы  $v_i q_i$  ( $0 \leq v_i < n, 0 \leq c_i < n$ ).

### Формат выходных данных

Для каждого запроса выведите одно число – расстояние до ближайшей вершины нужного цвета, или  $-1$ , если в дереве нет вершин такого цвета.

### Примеры

centroid.in	centroid.out
5	0 1 2 -1 2 1 2 1 1
0 1 1 3	
1 2 3 2 1	
9	
0 1	
0 2	
0 3	
1 0	
2 1	
2 2	
3 3	
3 1	
4 2	