

## Задача А. Мега-инверсии

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 В. Разреженные таблицы

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

Дан массив из  $n$  чисел. Требуется написать программу, которая будет отвечать на запросы следующего вида: найти минимум на отрезке между  $u$  и  $v$  включительно.

### Формат входного файла

В первой строке входного файла даны три натуральных числа  $n, m$  ( $1 \leq n \leq 10^5$ ,  $1 \leq m \leq 10^7$ ) и  $a_1$  ( $0 \leq a_1 < 16\,714\,589$ ) — количество элементов в массиве, количество запросов и первый элемент массива соответственно. Вторая строка содержит два натуральных числа  $u_1$  и  $v_1$  ( $1 \leq u_1, v_1 \leq n$ ) — первый запрос.

Элементы  $a_2, a_3, \dots, a_n$  задаются следующей формулой:

$$a_{i+1} = (23 \cdot a_i + 21563) \bmod 16714589.$$

Например, при  $n = 10$ ,  $a_1 = 12345$  получается следующий массив:  $a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095)$ .

Запросы генерируются следующим образом:

$$u_{i+1} = ((17 \cdot u_i + 751 + ans_i + 2i) \bmod n) + 1,$$

$$v_{i+1} = ((13 \cdot v_i + 593 + ans_i + 5i) \bmod n) + 1,$$

где  $ans_i$  — ответ на запрос номер  $i$ .

### Формат выходного файла

В выходной файл выведите  $u_m, v_m$  и  $ans_m$  (последний запрос и ответ на него).

### Примеры

sparse.in	sparse.out
10 8 12345	5 3 1565158
3 9	

## Задача С. Range Variation Query

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

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 значениями 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 ( $1 \leq k \leq 100\,000$ ). 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 значениями 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  $y_i$ . В этом случае  $-100\,000 \leq x_i \leq -1$  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 значениями на соответствующем отрезке.

**Примеры**

rvq.in	rvq.out
7	34
1 3	68
2 4	250
-2 -100	234
1 5	1
8 9	
-3 -101	
2 3	

**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	

**Задача D. 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 увлёкся рисованием одномерных чёрно-белых картин. Он пытается найти оптимальные 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.