

Задача А. Менеджер памяти

Имя входного файла: `memory.in`
Имя выходного файла: `memory.out`
Ограничение по времени: 6 секунды
Ограничение по памяти: 512 мегабайт

Одно из главных нововведений новейшей операционной системы Indows 7 — новый менеджер памяти. Он работает с массивом длины N и позволяет выполнять три самые современные операции:

- `copy(a, b, l)` — скопировать отрезок длины $[a, a + l - 1]$ в $[b, b + l - 1]$
- `sum(l, r)` — посчитать сумму элементов массива на отрезке $[l, r]$
- `print(l, r)` — напечатать элементы с l по r , включительно

Вы являетесь разработчиком своей операционной системы, и Вы, безусловно, не можете обойтись без инновационных технологий. Вам необходимо реализовать точно такой же менеджер памяти.

Формат входных данных

Первая строка входного файла содержит целое число N ($1 \leq N \leq 1\,000\,000$) — размер массива, с которым будет работать Ваш менеджер памяти.

Во второй строке содержатся четыре числа $1 \leq X_1, A, B, M \leq 10^9 + 10$. С помощью них можно сгенерировать исходный массив чисел X_1, X_2, \dots, X_N . $X_{i+1} = (A * X_i + B) \bmod M$

Следующая строка входного файла содержит целое число K ($1 \leq K \leq 200\,000$) — количество запросов, которые необходимо выполнить Вашему менеджеру памяти.

Далее в K строках содержится описание запросов. Запросы заданы в формате:

- `cpy a b l` — для операции `copy`
- `sum l r` — для операции `sum` ($l \leq r$)
- `out l r` — для операции `print` ($l \leq r$)

Гарантируется, что суммарная длина запросов `print` не превышает 3000. Также гарантируется, что все запросы корректны.

Формат выходных данных

Для каждого запроса `sum` или `print` выведите в выходной файл на отдельной строке результат запроса.

Примеры

memory.in	memory.out
6	1 2 6 1 2 6
1 4 5 7	1 2 1 2 2 6
7	6
out 1 6	1 1 2 1 2 6
cpy 1 3 2	13
out 1 6	
sum 1 4	
cpy 1 2 4	
out 1 6	
sum 1 6	

Задача В. Ферма

Имя входного файла: `segtree2d.in`
Имя выходного файла: `segtree2d.out`
Ограничение по времени: 10 секунды
Ограничение по памяти: 256 мегабайт

Настала весна и фермер решил заняться удобрением своего земельного участка размерами $x \times y$ метров. Для этого он закупил удобрения. До начала посевов остаётся n дней, и фермер хочет успеть сделать как можно больше.

За день фермер может одну из следующих вещей:

- увеличить продуктивность прямоугольного участка земли со сторонами, параллельными осям координат с углами (x_1, y_1) и (x_2, y_2) на значение w
- посчитать суммарную продуктивность участка $(x_1, y_1) - (x_2, y_2)$

Удобрять фермер любит сам, а вот заниматься скучными расчетами ему не интересно. Помогите ему в этом.

Формат входных данных

В первой строке входного файла записаны числа x и y ($1 \leq x, y \leq 1000$). В следующей строке написано количество оставшихся до начала посевов дней n ($1 \leq n \leq 100000$). Следующие n строк описывают действия фермера в соответственный день в следующем формате:

- 1 x_1 y_1 x_2 y_2 w — фермер удобряет участок. ($1 \leq x_1 \leq x_2 \leq x$, $1 \leq y_1 \leq y_2 \leq y$, $-10000 \leq w \leq 10000$)
- 2 x_1 y_1 x_2 y_2 — фермер просит посчитать плодородность участка. ($1 \leq x_1 \leq x_2 \leq x$, $1 \leq y_1 \leq y_2 \leq y$)

Формат выходных данных

Для каждого запроса плодородности участка в отдельной строке выведите плодородность этого участка.

Примеры

segtree2d.in	segtree2d.out
8 8	3
3	
1 2 2 8 8 2	
1 1 1 2 2 1	
2 2 2 2 2	

Задача С. Почтовая реформа

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

В Флатландии идет пора реформ. Недавно была проведена реформа дорог, так что теперь по дорогам страны из любого города можно добраться в любой другой, причем только одним способом. Также была проведена реформа волшебников, так что в каждом городе остался ровно один волшебник. Теперь же началась реформа почтовой системы.

Недавно образованное почтовое агентство «Экс-Федя» предлагает уникальную услугу — коллективную посылку. Эта услуга позволяет отправлять посылки жителям всех городов на каком-либо пути по цене обычной посылки. Удивительно, но пользоваться такой услугой стали только волшебники Флатландии, которые стали в большом количестве отправлять друг другу магические кактусы. Агентство столкнулось с непредвиденной проблемой: как известно, все волшебники живут в башнях и мало того, что не строят в них лестницы, так еще время от времени меняют их высоту. Поэтому, чтобы доставить посылку волшебнику, который живет в башне высотой h , курьеру агентства требуется иметь с собой не менее h метров веревки.

Вам поручено руководить отделом логистики — по имеющимся данным о высотах башен и об их изменениях вам нужно определять минимальную длину веревки, которую нужно выдать курьеру, который доставляет посылки между городами i и j .

Формат входных данных

Первая строка входного файла содержит число n — количество городов в Флатландии ($1 \leq n \leq 50\,000$). Во второй строке находится n положительных чисел, не превосходящих 10^5 — высоты башен в городах. В следующих $n - 1$ строках содержится по два числа u_i и v_i — описание i -й дороги, $1 \leq u_i, v_i \leq n, u_i \neq v_i$. В следующей строке содержится число k — количество запросов ($1 \leq k \leq 100\,000$). В следующих k строках содержатся описания запросов в следующем формате:

- Уведомление от волшебника из города i о том, что высота его башни стала равна h , имеет вид $! i h, 1 \leq i \leq n, 1 \leq h \leq 10^5$.
- Запрос от курьера о выдаче веревки для доставки посылок во все города на пути от i до j включительно имеет вид $? i j, 1 \leq i, j \leq n$.

Формат выходных данных

Для каждого запроса доставки посылок выведите минимальную длину веревки, которую необходимо выдать курьеру.

Примеры

mail.in	mail.out
3 1 2 3 1 3 2 3 5 ? 1 2 ! 1 5 ? 2 3 ! 3 2 ? 1 2	3 3 5
1 100 5 ! 1 1 ? 1 1 ! 1 1000 ? 1 1 ! 1 1	1 1000

Задача D. Урны и шары

Имя входного файла: `balls.in`
Имя выходного файла: `balls.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Путь у вас есть n урн, в каждой из которых лежит по одному шару. Урна с номером i содержит шарик под номером i . У вас есть специальное устройство, которое позволяет перемещать шары. Им чрезвычайно просто пользоваться: сначала вы выбираете некоторый отрезок последовательных урн. После этого вы выбираете некоторый другой отрезок последовательных урн такой же длины, как и исходный, и затем шары из урн первого отрезка перемещаются в соответствующие урны второго отрезка.

Дана последовательность перемещений. Установите, в какой урне окажется каждый шарик.

Формат входных данных

Первая строка входных данных содержит два числа n и m — число урн и число перемещений, соответственно ($1 \leq n \leq 100\,000$, $1 \leq m \leq 50\,000$). Каждая из следующих m строк содержит три числа $count_i$, $from_i$ и to_i , которые означают одновременное перемещение всех шариков из урны $from_i$ в урну to_i , всех шариков из урны $from_i + 1$ в урну $to_i + 1$, ..., всех шариков из урны $from_i + count_i - 1$ в урну $to_i + count_i - 1$ ($1 \leq count_i, from_i, to_i \leq n$, $\max(from_i, to_i) + count_i \leq n + 1$).

Формат выходных данных

Выведите n чисел — итоговые позиции каждого шарика.

Примеры

<code>balls.in</code>	<code>balls.out</code>
2 3	1 1
1 1 2	
1 2 1	
1 2 1	