A\*. Копатыч и искусство легких касаний

|  |  |
| --- | --- |
| Ограничение времени | 0.2 секунд |
| Ограничение памяти | 8.0 Мб |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

*Если честно, то в жизни мне повезло лишь однажды: когда я встретил вас — моих друзей.*

За окном rain, на душе pain, а смешарики устраивают смотр ледяных скульптур! Правила конкурса следующие: статуи выставляются в ряд, каждый смешарик голосует за понравившиеся ему бюсты. После этого Копатыч, как единственный независимый судья (независимый, потому что проспал весь творческий процесс) пытается определить лучшую ледяную композицию.

По мнению медведя композицией может быть любой подотрезок из ряда скульптур (Копатыч сельский житель и плохо разбирается в искусстве). Также Копатыч верит в то, что лучшая композиция — композиция, у скульптур в которой суммарное количество голосов максимально.

Помогите Копатычу тем, что напишете программу, которая по его запросу будет говорить суммарное количество голосов за ледяную композицию.

Формат ввода

В первой строке вводится одно натуральное число N*N* (1≤N≤105)(1≤*N*≤105) — количество скульптур, выставленных в ряд.

Во второй строке вводятся N*N* чисел от 1 до 105105 — количество голосов у скульптур в ряду.

В третьей строке вводится одно натуральное число K*K* (1≤K≤3⋅104)(1≤*K*≤3⋅104) — количество запросов Копатыча.

В следующих K*K* строках вводится по два числа — номера левого и правого концов композиции (считается, что скульптуры нумеруются с единицы).

Формат вывода

Для каждого запроса выведите сумму чисел соответствующей композиции скульптур. Числа выводите в одну строку через пробел.

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 5  4 4 8 7 8  2  1 2  1 3 | 8 16 |

Примечания

Сложность решения должна быть O(N+K)O(*N*+*K*).

Набрать здесьОтправить файл

B\*. История Кар-Карыча или Несколько мелочей на фоне природы

|  |  |
| --- | --- |
| Ограничение времени | 0.2 секунд |
| Ограничение памяти | 8.0 Мб |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

*Можно всю жизнь прождать подходящий момент и в итоге так и не сказать друг другу что-то важное.*

Кто-то за ночь уничтожил выставку ледяных скульптур. Известно, что ни одному из смешариков в одиночку не хватило бы времени на такое. Налицо тайный сговор или действия организованной преступной группировки. Смешарики решают провести суд Кар-Карычей — самый гуманный суд в мире. Всех смешариков построили в шеренгу, чтобы выявить самую подозрительную группировку.

Кар-Карыч считает, что группировкой может быть любой подотрезок из шеренги смешариков (общеизвестно, что все налетчики стараются держаться рядом). Также Кар-Карыч верит в то, что самая подозрительная группировка — группировка, у смешариков которой наибольшая суммарная подозрительность.

Помогите Кар-Карычу и напишите программу, которая найдет самую подозрительную группировку.

Формат ввода

В первой строке дано натуральное число N*N* (1≤n≤105)(1≤*n*≤105) — длина шеренги. Во второй строке через пробел перечислена подозрительность у каждого из смешариков (учтите, что она может быть отрицательной). Числа по модулю не превышают 104104.

Формат вывода

В одну строку через пробел выведите границы группировки (в 1-индексации), а также ее подозрительность.

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 5  -1 2 3 -2 5 | 2 5 8 |

Примечания

Сложность решения должна быть O(N+K)O(*N*+*K*).

C\*. Пинкод

|  |  |
| --- | --- |
| Ограничение времени | 0.5 секунд |
| Ограничение памяти | 32.0 Мб |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

*Очень трудно жить, думая о каждой ошибке, которую ты совершил, поэтому всё плохое забываешь, а помнишь только хорошее — очень удобно, но от себя не уйдёшь.*

Для новой серии научно-популярного шоу смешарикам необходимо найти в творческой команде самого сообразительного смешарика, принявшего участие в наименьшем количестве выпусков. Командой может быть любая шеренга смешариков. Вам нужно помочь нашим героям для каждой команды, которую они попросят, найти самого редкого гостя на экране.

Формат ввода

В первой строке вводятся два числа 1≤n,q≤1051≤*n*,*q*≤105 - число смешариков и число запросов. В следующей строке n*n* чисел - количество выпусков с каждым из смешариков. В следующих q*q* строках запросы 1≤l≤r≤n1≤*l*≤*r*≤*n*

Формат вывода

Выведите ответы на q*q* запросов

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 10 7  1 2 3 4 5 6 7 8 9 0  3 9  2 6  5 10  4 5  5 9  4 6  2 5 | 3  2  0  4  5  4  2 |

D\*. Шаровидение: Как стать звездой?

|  |  |
| --- | --- |
| Ограничение времени | 0.5 секунд |
| Ограничение памяти | 32.0 Мб |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

*Вальсы, твисты, ламбады... Мода меняется. Но моё сердце бьётся только в ритме диско!*

Бараш тоже решил поруководить толпой и заодно попасть на Шаровидение. Он выстроил всех смешариков так, чтобы любой подотрезок шеренги был ансамблем песни и пляски.

Тонко чувствующему поэту позвонили с телевидения и сказали, что общая талантливость ансамбля равна НОДу талантов смешариков, состоящих в нем.

Вам нужно снова помочь одному из наших героев. Напишите программу, которая сможет отвечать на запросы о талантливости ансамблей.

Формат ввода

В первой строке вводится одно натуральное число N*N* (1≤N≤105)(1≤*N*≤105) — количество смешариков в шеренге.

Во второй строке вводятся N*N* чисел от 1 до 105105 — талантливость каждого смешарика.

В третьей строке вводится одно натуральное число K*K* (1≤K≤3⋅104)(1≤*K*≤3⋅104) — количество запросов на вычисление талантливости ансамбля.

В следующих K*K* строках вводится по два числа — номера левого и правого элементов концов шеренги в 1-индексации.

Формат вывода

Для каждого запроса выведите талантливость музыкального коллектива. Разделяйте числа переводами строк.

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 5  2 2 2 1 5  2  2 3  2 5 | 2  1 |

# E\*. Последняя ошибка Пина

|  |  |
| --- | --- |
| Ограничение времени | 0.2 секунд |
| Ограничение памяти | 8.0 Мб |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Я тебя породить, я тебе и говорить «до свидания»! Auf Wiedersehen!

Пин готовится к технологической выставке, продумывая лучшую расстановку экспонатов. По мнению пингвина, расстановка будет наилучшей, если все его механизмы будут расположены вдоль одной прямой.

Теперь Пин ищет, а как лучше распределить устройства вдоль стенда, чтобы произвести впечатление на комиссию. Он попросил вас, как лучшего программиста Ромашковой долины, написать программу, которая сможет отвечать на два типа запросов:

* "slr*slr*" — узнать суммарную технологическую сложность устройств, расположенных на отрезке [l,r][*l*,*r*] в 1-индексации;
* "uiv*uiv*" — переставить приборы так, чтобы в позиции с индексом i*i* технологическая сложность стала v*v*. Считайте, что он просто заменяет технологическую сложность на позиции i*i* на сложность v*v*. Не задумывайтесь, как Пин это делает, - его гениальность нам не постижима.

## Формат ввода

В первой строке вводится одно натуральное число N*N* (1≤N≤105)(1≤*N*≤105) — длина стенда.

Во второй строке вводятся N*N* чисел от 0 до 105105 — изначальная расстановка экспонатов.

В третьей строке вводится одно натуральное число M*M* (1≤M≤3⋅104)(1≤*M*≤3⋅104) — количество запросов.

Каждая из следующих M*M* строк представляет собой описание запроса в формате выше.

## Формат вывода

Для каждого запроса s*s* выведите результат. Все числа выводите в одну строку через пробел.

## Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 10  613 263 312 670 216 142 976 355 488 370  10  s 2 7  s 4 8  u 7 969  u 1 558  s 2 7  u 2 731  s 4 9  s 1 3  u 8 76  u 5 377 | 2579 2359 2572 2840 1601 |

F\*. Приключения Биби или Космическая одиссея

|  |  |
| --- | --- |
| Ограничение времени | 0.5 секунд |
| Ограничение памяти | 16.0 Мб |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

*Вот так всегда: для кого-то балласт, а для кого-то сокровище.*

Ромашковая долина наконец дождалась великого дня — робот Биби вернулся из космического путешествия во главе целой космической флотилии. По опыту Биби для космоса не существует трехмерной фигуры лучше куба, поэтому его космофлот выглядит как куб с длиной стороны N*N*.

Для организации управления наш герой разделил исходный куб на N3*N*3 кубиков с ребром длины 1.

Биби хочет научиться быстро обрабатывать запросы следующих типов:

* В каком-то секторе изменилось число космолетов;
* Узнать количество космолетов в прямоугольном параллелепипеде.

Формат ввода

Первая строка входного файла содержит натуральное число N*N* (N≤27)(*N*≤27). Координаты секторов~--- целые числа от 0 до N−1*N*−1. Далее следуют записи о происходивших событиях по одной в строке.

В начале строки записано число m*m*. Если m*m* равно:

* 1, то за ним следуют 4 числа: x,y,z,k*x*,*y*,*z*,*k* (0≤x,y,z≤N,−20000≤k≤20000)(0≤*x*,*y*,*z*≤*N*,−20000≤*k*≤20000)~--- координаты сектора и величина, на которую в нем изменилось количество космолетов;
* 2, то за ним следуют 6 чисел: x1,y1,z1,x2,y2,z2*x*1​,*y*1​,*z*1​,*x*2​,*y*2​,*z*2​ (0≤xi,yi,zi≤N,x1≤x2,y1≤y2,z1≤z2)(0≤*xi*​,*yi*​,*zi*​≤*N*,*x*1​≤*x*2​,*y*1​≤*y*2​,*z*1​≤*z*2​), которые означают, что нужно узнать, сколько космолетов в прямоугольном параллелепипеде с данными противолежащими вершинами и сторонами, параллельными осям координат.
* 3, то это означает, что Биби утомился, а значит можно закончить работу программы.

Количество записей во входном файле не превосходит 100500. Изначально в воздухе нет ни одного космолета.

Формат вывода

Для каждого запроса второго типа выведите на отдельной строке одно число~--- искомое количество космолетов.

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 2  2 1 1 1 1 1 1  1 0 0 0 1  1 0 1 0 3  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 | 0  1  4  2 |

# G\*. История Кроша или Зачем нужны друзья?

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 8.0 Мб |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Если ты не можешь бросить старый зонтик, то как мы можем бросить тебя — старого друга? Ты ведь тоже часть нашей биографии.

Наконец-то к популярному среди старшего поколения явлению потянулась и перспективная молодежь. Крош изучает тренды Шар-Тока и пытается найти лучших кандидатов для совместного видео. Чтобы видео залетело в тренды, нашему другу нужно знать, сколько именно смешариков в конкретном творческом объединении имеет максимальное число подписчиков.

Если вы вдруг забыли, то напомним, что все выборы в Шарараме примечательны тем, что любой подотрезок ряда является творческим объединением.

## Формат ввода

В первой строке вводится одно натуральное число N*N* (1≤N≤105)(1≤*N*≤105)~--- количество смешариков в ряду.

Во второй строке вводятся N*N* чисел от 1 до 105105~--- количество подписчиков у смешарика.

В третьей строке вводится одно натуральное число K*K* (1≤K≤3⋅104)(1≤*K*≤3⋅104)~--- количество запросов Кроша.

В следующих K*K* строках вводится по два числа~--- номера левого и правого смешарика, границ творческого объединения (в 1-индексации).

## Формат вывода

Для каждого запроса выведите в отдельной строке через пробел максимальное количество подписчиков у смешарика в указанном объединении и количество смешариков с числом подписчиков, равным максимальному.

## Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 5  2 2 2 1 5  2  2 3  2 5 | 2 2  5 1 |

# H\*. Укуси меня пчела

|  |  |
| --- | --- |
| Ограничение времени | 0.2 секунд |
| Ограничение памяти | 8.0 Мб |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Ёлки-иголки, да что же это такое тут происходит?

После ударного дачного сезона Копатыч расставил свои ульи в ряд, чтобы определить лучшую пчелиную семью (подотрезок ряда).

Качество пчелиной семьи Копатыч определяет по наибольшему количеству собранного меда в одном улье. Периодически Копатыч собирает дополнительный мед из того или иного улья.

Необходимо помочь Копатычу определить качество каждой пчелиной семьи и разобраться в его отношениях со сбором меда.

## Формат ввода

В первой строке вводится одно натуральное число N*N* (1≤N≤100000)(1≤*N*≤100000) — количество ульев в ряду.

Во второй строке вводятся N*N* чисел от 00 до 100000100000 — количество собранного меда с каждого улья в начальный момент.

В третьей строке вводится одно натуральное число M*M* (1≤M≤30000)(1≤*M*≤30000) — количество запросов Копатыча.

Каждая из следующих M*M* строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (m*m* — найти наибольшее число собранного меда в семье, a*a* — увеличить число собранного меда у пчелиной семьи).

Следом за m*m* вводятся два числа — левая и правая граница ряда.

Следом за a*a* вводятся три числа — левый и правый концы ряда и число add*add*, на которое нужно увеличить количество собранного меда у каждого улья данной пчелиной семьи (0≤add≤100000)(0≤*add*≤100000).

## Формат вывода

Выведите в одну строку через пробел ответы на каждый запрос m*m*.

## Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 5  2 4 3 1 5  5  m 1 3  a 2 4 100  m 1 3  a 5 5 10  m 1 5 | 4 104 104 |



# I\*. Деккардова организация

|  |  |
| --- | --- |
| Ограничение времени | 0.2 секунды |
| Ограничение памяти | 16Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Доминик решил пересмотреть свой автопарк и организовать информацию о нем. Для этого он присвоил каждой машине номер и приоритет. Доминик уверен, что если расположить машины как вершины в декартовом дерева поиска, то он сможет подбирать нужную тачку на гонку оптимально.

Так как Доминик, как и всегда, торопится на встречу с семьей, ему нужно построить дерево для хранения информации о машинах *за линейное время от числа машин*.

## Формат ввода

В первой строке записано число *N* — количество пар номер-приоритет. Далее следует *N* *(1 ≤ N ≤ 50000)* пар *(ai, bi)*. Для всех пар *|ai|, |bi| ≤ 30000*. *ai ≠ aj* и *bi ≠ bj* для всех *i ≠ j*. Гарантируется, что пары отсортированы по возрастанию *ai*.

## Формат вывода

Если Доминик выбрал неверные приоритеты, и дерево построить невозможно, то выведите NO. Иначе выведите *N* строк, каждая из которых должна описывать вершину. Описание вершины состоит из трёх чисел: номер предка, номер левого сына и номер правого сына. Если у вершины отсутствует предок или какой-либо из сыновей, то выводите на его месте число 0.

Если подходящих деревьев несколько, выведите любое.

## Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 7  0 5  1 3  2 2  3 9  4 11  5 4  6 6 | YES  2 0 0  3 1 0  0 2 6  6 0 5  4 0 0  3 4 7  6 0 0 |

## Примечания

«Неважно, что ты стоишь рядом с тачкой и она твоя. Важно другое: то, как ты на ней ездишь! Запомни это» (c) Форсаж

J\*. Дерево настроения семьи

|  |  |
| --- | --- |
| Ограничение времени | 5 секунд |
| Ограничение памяти | 256Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

В этой задаче вам предстоит доказать, что вы достойный член семьи. А именно, вам предстоит реализовать сбалансированное дерево семьи.

Формат ввода

Входной файл содержит описание операций с деревом семьи, их количество не превышает *105*. В каждой строке находится одна из следующих операций.

* *insert x* — добавить в дерево члена семьи с настроением *x*. Если такой уже есть, то ничего делать не надо.
* *delete x* — удалить из дерева члена семьи с настроением *x*. Если его там нет, то ничего делать не надо.
* *exists x* — если имеется член семьи с настроением *x*, выведите true, иначе, false.
* *next x* — выведите настроение такого члена семьи, что оно лучше чем *x*, но минимально среди таких или none, если такого нет. Это нужно, чтобы понимать, кого на предстоящую гонку можно брать в команду, а кому надо отдохнуть.
* *prev x* — выведите настроение такого члена семьи, что оно хуже чем *x*, но максимально среди таких или none, если такого нет.
* *kth k* — выведите *k*-ю величину настроения (нумерация с нуля). Если такого не существует, то выведите none.

Все числа во входном файле по модулю не превышают *109*.

Формат вывода

Выведите последовательно результат выполнения всех операций выше. Следуйте формату выходного файла из примера.

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| insert 2  insert 5  insert 3  exists 2  exists 4  next 4  prev 4  delete 5  next 4  prev 4  kth 0  kth 2 | true  false  5  3  none  3  2  none |

Примечания

«Нет ничего важнее, чем личный кодекс» (c) Форсаж 4. Новые тачки. Старая команда. С целью соблюдения личного кодекса, было принято решение в этой задаче необходимо реализовать декартово дерево, однако для получения удвоенного балла можно и красно-черное дерево или *B*-дерево.

K\*. Еще раз сумма на отрезке

|  |  |
| --- | --- |
| Ограничение времени | 2 секунды |
| Ограничение памяти | 32Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Реализуйте структуру данных, которая поддерживает множество *S* целых чисел, с которым разрешается производить следующие операции:

* *add(i)* — добавить в множество *S* число *i* (если оно там уже есть, то множество не меняется);
* *sum(l, r)* — вывести сумму всех элементов *x* из *S*, которые удовлетворяют неравенству*l ≤ x ≤ r*.

Формат ввода

Исходно множество *S* пусто. Первая строка входного файла содержит *n* — количество операций *(1 ≤ n ≤ 3 ⋅ 105)*. Следующие *n* строк содержат операции. Каждая операция имеет вид либо «+ i», либо «? l r». Операция «+ i» задает запрос *add(i)*. Операция «? l r» задает запрос *sum(l, r)*.

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до *109*.

Формат вывода

Для каждого запроса выведите одно число — ответ на запрос.

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 6  + 1  + 3  + 3  ? 2 4  + 4  ? 2 4 | 3  7 |

# L\*. Ёжик и Кордебалет

|  |  |
| --- | --- |
| Ограничение времени | 0.5 секунд |
| Ограничение памяти | 17.5 Мб |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Крош: Чем больше трудностей, тем лучше. В трудностях познается характер.

Ёжик: А может, некоторым лучше и не знать про свой характер. Вдруг ты такое узнаешь, что потом жить… не-за-хо-чет-ся!

Ёжика вновь назначили ответственным за репетицию кордебалета. И вот смешарики под угрозой записи в книжечку вновь выстроились в шеренгу и рассчитались по порядку (с 11 по n*n*).

В поисках оптимального построения Ёжик то и дело отдает команды перестроения шеренги. Звучат они так: Смешарики с li*li*​ по ri*ri*​ --- в начало!. После этой команды смешарики стоящие в позициях с li*li*​ по ri*ri*​ включительно, немедленно перемещаются в начало шеренги, сохраняя относительный порядок.

Например, если в некоторый момент смешарики стоят в порядке 2,3,6,1,5,4 (каждому смешарику присвоен свой номер), то после команды: Смешарики с 2 по 4 --- в начало! порядок будет 3,6,1,2,5,4.

Помогите Ёжику разобраться, в каком порядке должны стоять смешарики после выполнения всех его команд, иначе он вас запишет в книжечку.

## Формат ввода

В первой строке два целых числа n*n* and m*m* (2≤n≤100000,1≤m≤100000)(2≤*n*≤100000,1≤*m*≤100000) --- количество смешариков и количество команд от Ёжика. Следующие m строк содержат по два целых числа li*li*​ и ri*ri*​ (1≤li≤ri≤n)(1≤*li*​≤*ri*​≤*n*).

## Формат вывода

Выведите n*n* целых чисел --- порядок после выполнения всех команд.

## Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 6 3  2 4  3 5  2 2 | 1 4 5 2 3 6 |

# M. Совунья и чистый спорт

|  |  |
| --- | --- |
| Ограничение времени | 3 секунды |
| Ограничение памяти | 256.0 Мб |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Я в детстве тоже хотела иметь волшебную палочку, чтобы сразу стать старенькой, как моя бабушка, получать пенсию и ничего не делать. Хорошо, что у меня не было волшебной палочки.

Смешарики устали сжигать калории на диете тревог и забот, поэтому сегодня происходит торжественное открытие Спартакиады. Совунье выпала честь наградить всех участников соревнований памятными наградами. Смешарики снова стоят в один ряд, а спортивной командой Совунья называет любой подотрезок этого ряда. Изначально у всех смешариков ноль наград, так что ваша задача — помочь провести награждение максимально быстро. Для этого Совунья сообщила вам, что смешариков всегда n=224*n*=224.

Собственно, награждение устроено достаточно просто, Совунья называет спортивную команду и количество наград, которое надо начислить смешарикам в команде. Всего таких «стадий награждения» будет m*m*.

Совунья хочет знать, не потратит ли она так все имеющиеся медали, а поэтому просит вас определить, сколько будет наград у команд после церемонии награждения. Таких запросов будет q*q*.

## Формат ввода

В первой строке числа m*m*, q*q* (1≤m,q≤2241≤*m*,*q*≤224). Во второй строке пара целых чисел a*a*, b*b* от 11 до 109109 , используемая в генераторе случайных чисел, который был разработан математиками Шарарама.

unsigned int a, b; // даны во входных данных

unsigned int cur = 0; // беззнаковое 32-битное число

unsigned int nextRand() {

cur = cur \* a + b; // вычисляется с переполнениями

return cur >> 8; // число от 0 до 224−1224−1.

}

Каждый запрос первого вида генерируется следующим образом:

add = nextRand(); // число, которое нужно прибавить

l = nextRand();

r = nextRand();

if (l > r) swap(l, r); // получили отрезок [l..r]

Каждый запрос второго вида генерируется следующим образом:

l = nextRand();

r = nextRand();

if (l > r) swap(l, r); // получили отрезок [l..r]

Сперва генерируются запросы на награждение, затем запросы наград у команды.

## Формат вывода

Выведите сумму ответов на все запросы второго типа по модулю 232232.

## Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 5 5  13 239 | 811747796 |

# P. Переселение

|  |  |
| --- | --- |
| Ограничение времени | 0.5 секунд |
| Ограничение памяти | 8Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Скоро начало учебного года, а значит, многие студенты вынуждены переселяться из привычного коридорного общежития в апартаменты премиум-класса. К сожалению, недобросовестные студенты постоянно затягивают свой переезд, поэтому новичкам ПФИМ приходится въезжать в комнаты, заполненные вещами дедов. В некоторых общежитиях практикуется выставление вещей предыдущих жильцов за дверь (иногда даже самими предыдущими жильцами), позднее кто-то (доподлинно неизвестно, кто) эти вещи забирает. Однако выставленные в проход вещи противоречат технике пожарной безопасности, а проверка регулярно посещает общежития.

Итак, сама задача: в общежитии имеется длинный основной коридор, который никак не должен быть захламлен по ТБ, но на мелкие нарушения проверка готова закрыть глаза. В этот коридор постоянно кто-то выставляет вещи, а кто-то постоянно их оттуда забирает. Когда приходит проверка, она считает, сколько предметов выставлено в коридор между двумя комнатами с заданными номерами (эти номера выбираются комиссией, проводящей проверку). Смоделируйте родное общежитие на языке *C++*.

## Формат ввода

В первой строке входного файла содержатся два числа: *N* — количество комнат *(1 ≤ N ≤ 10 000)* и *M* — количество производимых в коридоре действий *(1 ≤ M ≤ 50 000)*. Каждое действие имеет вид "*1 L R S*", что означает, что на участок коридора от *L*-ой комнаты до *R*-ую комнаты между всеми соседними комнатами выставили *S* предметов (*S* может быть и отрицательным, тогда это означает, что предметы из коридора забрали), или "*2 L R*", что значит, что проверка интересуется, сколько предметов выставлено на участке коридора от *L*-ой комнаты до *R*-й. Комнаты нумеруются от *0* до *N*. Гарантируется, что для запросов вида "*1 L R S*" при *S < 0* на каждом участке между соседними комнатами от *L* до *R* число предметов составляет не менее S.

## Формат вывода

На каждую команду 2 (запрос) вы должны выводить число *K* — суммарное число предметов, находящихся между *L*-й и *R*-й комнатами. Каждое число должно выводиться на новой строке. Известно, что в процессе работы суммарное количество предметов на любом интервале не превышает *231*.

## Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 10 5  1 0 9 10  1 1 5 -3  2 4 8  1 0 6 25  2 0 2 | 37  67 |

R. RMQ + insert

|  |  |
| --- | --- |
| Ограничение времени | 2 секунды |
| Ограничение памяти | 128Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Вы уже решали задачу RMQ, даже если есть обновления на отрезке. В этот раз границы решения этой задачи будут расширены. А именно, надо разработать структуру данных в виде массива, который умеет обрабатывать следующие запросы:

* «? l r» — найти минимум на отрезке *[l, r]* в 1-индексации;
* «+ i x» — добавить элемент *x* после *i*-го (в 1-индексации) элемента. Если *i = 0*, то вставка идет в начало списка.

Изначально структура пустая.

Формат ввода

Первая строка входного файла содержит единственное целое число *n* — число операций над массивом *(1 ≤ n ≤ 2 ⋅ 105)*. Следующие *n* строк описывают сами операции. Все операции добавления являются корректными. Все числа, хранящиеся в массиве, по модулю не превосходят *109*.

Формат вывода

Для каждой операции в отдельной строке выведите её результат.

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 8  + 0 5  + 1 3  + 1 4  ? 1 2  + 0 2  ? 2 4  + 4 1  ? 3 5 | 4  3  1 |

S. RMQ + Reverse

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 32Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Мы опять решаем задачу RMQ, только теперь нужно обработать следующие запросы:

* «1 l r» — развернуть отрезок *[l, r]*.
* «2 l r» — найти минимум на отрезке *[l, r]*;

Формат ввода

Первая строка файла содержит два числа *n*, *m*. *(1 ≤ n, m ≤ 105)* Во второй строке находится *n* чисел *ai* *(1-109 ≤ ai ≤ 109)* — исходный массив. Остальные *m* строк содержат запросы в формате, описанном в условии. Для чисел *l, r* выполняется ограничение *(1≤ l ≤ r ≤ n)*.

Формат вывода

На каждый запрос типа 2, во входной файл выведите ответ на него, в отдельной строке.

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 10 7  5 3 2 3 12 6 7 5 10 12  2 4 9  1 4 6  2 1 8  1 1 8  1 8 9  2 1 7  2 3 6 | 3  2  2  2 |