

Задача А. Суперминимум

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дано N чисел. Для каждых K подряд идущих чисел найти минимальное среди них.

Решите задачу, используя SQRT-декомпозицию.

Формат входных данных

В первой строке даны числа N и K ($1 \leq N \leq 150000$, $1 \leq K \leq 10000$, $K \leq N$), разделенные пробелом. Во второй строке записано N целых чисел через пробел. Числа находятся в диапазоне от -32768 до 32767 .

Формат выходных данных

Для каждых K подряд идущих чисел вывести минимальное из них.

Пример

стандартный ввод	стандартный вывод
7 4 2019 2020 3000 3001 5000 -1 -3	2019 2020 -1 -3

Задача В. Максимумы на подотрезках

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Реализуйте структуру данных для эффективного вычисления максимумов подряд идущих элементов массива. **Решите эту задачу, используя SQRT-декомпозицию.**

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leq N \leq 100000$) — количество чисел в массиве.

Во второй строке вводятся N чисел от 1 до 100000 — элементы массива.

В третьей строке вводится одно натуральное число K ($1 \leq K \leq 30000$) — количество запросов на вычисление максимума.

В следующих K строках вводится по два числа — номера левого и правого элементов отрезка массива (считается, что элементы массива нумеруются с единицы).

Формат выходных данных

Для каждого запроса выведите значение максимального элемента на указанном отрезке массива.

Пример

стандартный ввод	стандартный вывод
5	2 5
2 2 2 1 5	
2	
2 3	
2 5	

Задача С. Количество нулей на подотрезке

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Реализуйте структуру данных для эффективного вычисления количества нулей в отрезке массива.

Решите эту задачу, используя SQRT-декомпозицию.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leq N \leq 100000$) — количество чисел в массиве.

Во второй строке вводятся N чисел от 0 до 100000 — элементы массива.

В третьей строке вводится одно натуральное число K ($1 \leq K \leq 30000$) — количество запросов на вычисление количества нулей.

В следующих K строках вводится по два числа — номера левого и правого элементов отрезка массива (считается, что элементы массива нумеруются с единицы).

Формат выходных данных

Для каждого запроса выведите количество нулей на соответствующем участке массива. Числа выводите в одну строку через пробел.

Пример

стандартный ввод	стандартный вывод
5	2 3
0 0 0 0 2	
2	
2 3	
2 5	

Задача D. Билеты на МЦД

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В новых элитных электричках МЦД каждому пассажиру положено сидячее место. Естественно, количество сидячих мест ограничено и на всех их может не хватить. Маршрут электрички проходит через $N + 1$ станцию, станции занумерованы от 0 до N . Когда человек хочет купить билет, он называет два числа x и y — номера станций, откуда и куда он хочет ехать. При наличии хотя бы одного сидячего места на этом участке на момент покупки ему продается билет, иначе выдается сообщение «билетов нет» и билет не продается. Ваша задача — написать программу, обслуживающую такого рода запросы в порядке их прихода.

Решите эту задачу, используя SQRT-декомпозицию.

Формат входных данных

В первой строке содержатся три числа N — количество станций ($1 \leq N \leq 200000$), K — количество мест в электричке ($1 \leq K \leq 1000$) и M — количество запросов ($1 \leq M \leq 100000$). В следующих M строках описаны запросы, каждый из которых состоит из двух чисел x и y ($0 \leq x < y \leq N$).

Формат выходных данных

На каждый запрос ваша программа должна выдавать результат в виде числа 0 если билет не продается и 1 если билет был продан. Каждый результат должен быть на отдельной строке

Пример

стандартный ввод	стандартный вывод
5 2 4	1
0 4	1
1 2	0
1 4	1
2 4	

Задача E. Лунки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Маленький Петя очень любит играть. Больше всего он любит играть в игру «Лунки». Это игра для одного игрока со следующими правилами:

Есть N лунок, расположенных в ряд, пронумерованных слева направо числами от 1 до N . У каждой лунки изначально установлена своя сила выброса (у лунки с номером i она равна a_i). Если вбросить шарик в лунку i , то он тут же вылетит из нее и попадет в лунку $i + a_i$, после чего он опять вылетит и т.д.. Если же лунки с таким номером нету, то он просто вылетит за край ряда. На каждом из M ходов игрок выбирает одно из двух действий:

- Установить силу выброса лунки a равной b .
- Вбросить шарик в лунку a и посчитать количество прыжков шарика, прежде чем он вылетит за край ряда, а так же записать номер лунки, после выпрыгивания из которой шарик вылетел за край.

У Пети есть некоторые проблемы с математикой, поэтому, как Вы уже догадались, именно Вам предстоит произвести все подсчеты.

Формат входных данных

Первая строка содержит два числа N и M ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^5$) — количество лунок в ряду и количество ходов. Следующая строка содержит N целых положительных чисел, не превышающих N — начальные силы выброса лунок. Следующие M строк задают ходы, сделанные Петей. Каждая строка может быть двух типов:

- 0 a b
- 1 a

Тут, первый тип означает что требуется установить силу выброса лунки a равной b , а второй означает что требуется вбросить мячик в лунку с номером a . Числа a и b — целые положительные и не превышают N .

Формат выходных данных

Для каждого хода второго типа (задающего вбрасывание шарика) в порядке следования во входном файле выведите два числа через пробел в отдельной строке — номер последней лунки перед вылетом шарика за край и количество прыжков.

Пример

стандартный ввод	стандартный вывод
8 5	8 7
1 1 1 1 1 2 8 2	8 5
1 1	7 3
0 1 3	
1 1	
0 3 4	
1 2	

Задача F. $\sqrt{\text{Range Minimum Query}}$

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Компания Giggle открывает свой новый офис в Судиславле, и вы приглашены на собеседование. Ваша задача — решить поставленную задачу.

Вам нужно создать структуру данных, которая представляет из себя массив целых чисел. Изначально массив пуст. Вам нужно поддерживать две операции:

- запрос: «? i j » — возвращает минимальный элемент между i -ым и j -м, включительно;
- изменение: «+ i x » — добавить элемент x после i -го элемента списка. Если $i = 0$, то элемент добавляется в начало массива.

Конечно, эта структура должна быть достаточно хорошей.

Формат входных данных

Первая строка входного файла содержит единственное целое число n — число операций над массивом ($1 \leq n \leq 200\,000$). Следующие n строк описывают сами операции. Все операции добавления являются корректными. Все числа, хранящиеся в массиве, по модулю не превосходят 10^9 .

Формат выходных данных

Для каждой операции в отдельной строке выведите её результат.

Пример

стандартный ввод	стандартный вывод
8	4
+ 0 5	3
+ 1 3	1
+ 1 4	
? 1 2	
+ 0 2	
? 2 4	
+ 4 1	
? 3 5	

Задача G. Река

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Во Флатландии протекает богатая рыбой река Большой Флат. Много лет назад река была поделена между n рыболовными предприятиями, каждое из которых получило непрерывный отрезок реки. При этом i -е предприятие, если рассматривать их по порядку, начиная от истока, изначально получило отрезок реки длиной a_i .

С тех пор с рыболовными предприятиями во Флатландии k раз происходили различные события. Каждое из событий было одного из двух типов: банкротство некоторого предприятия или разделение некоторого предприятия на два. При некоторых событиях отрезок реки, принадлежащий предприятию, с которым это событие происходит, делится на две части. Каждый такой отрезок имеет длину большую или равную 2. Деление происходит по следующему правилу. Если отрезок имеет четную длину, то он делится на две равные части. Иначе он делится на две части, длины которых различаются ровно на единицу, при этом часть, которая ближе к истоку реки, имеет меньшую длину.

При банкротстве предприятия происходит следующее. Отрезок реки, принадлежавший обанкротившемуся предприятию, переходит к его соседям. Если у обанкротившегося предприятия один сосед, то этому соседу целиком передается отрезок реки обанкротившегося предприятия. Если же соседей двое, то отрезок реки делится на две части описанным выше способом, после чего каждый из соседей присоединяет к своему отрезку ближайшую к нему часть.

При разделении предприятия отрезок реки, принадлежавший разделяемому предприятию, всегда делится на две части описанным выше способом. Разделившееся предприятие ликвидируется, и образуются два новых предприятия. Таким образом, после каждого события каждое предприятие владеет некоторым отрезком реки.

Министерство финансов Флатландии предлагает ввести налог на рыболовные предприятия, пропорциональный квадрату длины отрезка реки, принадлежащего соответствующему предприятию. Чтобы проанализировать, как будет работать этот налог, министр хочет по имеющимся данным узнать, как изменялась величина, равная сумме квадратов длин отрезков реки, принадлежащих предприятиям, после каждого произошедшего события.

Требуется написать программу, которая по заданному начальному разделению реки между предприятиями и списку событий, происходивших с предприятиями, определит, чему равна сумма квадратов длин отрезков реки, принадлежащих предприятиям, в начальный момент времени и после каждого события.

Формат входных данных

Первая строка входного файла содержит два целых числа: n и p — исходное количество предприятий ($2 \leq n \leq 100\,000$) и просто число ($0 \leq p \leq 4$). Вторая строка входного файла содержит n целых чисел a_1, a_2, \dots, a_n — длины исходных отрезков реки.

Третья строка входного файла содержит целое число k — количество событий, происходивших с предприятиями ($1 \leq k \leq 100\,000$).

Последующие k строк содержат описания событий, i -я строка содержит два целых числа: e_i и v_i — тип события и номер предприятия, с которым оно произошло. Значение $e_i = 1$ означает, что предприятие, которое после всех предыдущих событий является v_i -м по порядку, если считать с единицы от истока реки, обанкротилось, а значение $e_i = 2$ означает, что это предприятие разделилось на два.

Гарантируется, что значение v_i не превышает текущее количество предприятий.

Гарантируется, что если отрезок предприятия при банкротстве или разделении требуется поделить на две части, то он имеет длину большую или равную 2. Гарантируется, что если на реке осталось единственное предприятие, оно не банкротится.

Пример

стандартный ввод	стандартный вывод
4 0	75
3 5 5 4	105
5	73
1 1	101
2 1	83
1 3	113
2 2	
1 3	