

## Задача А. 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 В. $\sqrt{\text{Range Minimum Query}}$

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 С. Сычи и АСМ

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Есть  $n$  сычей (программистов). Известны пары друзей. Три сыча образуют команду на АСМ, если все три попарно дружат. Требуется найти количество команд сычей на АСМ.

### Формат входных данных

Первая строка входных данных содержит два целых числа  $n$  и  $m$  — количество сычей и дружеских связей соответственно ( $1 \leq n, m \leq 3 \cdot 10^5$ ). Каждая из следующих  $m$  строк содержит по два целых числа от 1 до  $n$  — номера сычей, которые дружат. Гарантируется, что никакая пара не указана два раза и сыч не дружит сам с собой.

### Формат выходных данных

В единственной строке выходного файла выведите количество команд сычей.

### Пример

стандартный ввод	стандартный вывод
6 6 1 2 2 3 3 1 4 2 3 4 5 1	2

## Задача D. Мощные юнги

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.8 секунд
Ограничение по памяти:	256 мегабайт

Имеется список из  $n$  юнг, для каждого из которых известен его рост  $a_1, a_2, \dots, a_n$ . Рассмотрим некоторый его подсписок  $a_l, a_{l+1}, \dots, a_r$ , где  $1 \leq l \leq r \leq n$ , и для каждого натурального числа  $s$  обозначим через  $K_s$  число юнг с ростом  $s$  в этом подсписке. Назовем *мощностью* подсписка сумму произведений  $K_s \cdot K_s \cdot s$  по всем различным натуральным  $s$ . Так как количество различных чисел в массиве конечно, сумма содержит лишь конечное число ненулевых слагаемых.

Необходимо вычислить мощности каждого из  $t$  заданных подсписков.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $t$  ( $1 \leq n, t \leq 200000$ ) — длина списка и количество запросов соответственно.

Вторая строка содержит  $n$  натуральных чисел  $a_i$  ( $1 \leq a_i \leq 10^6$ ) — рост юнг.

Следующие  $t$  строк содержат по два натуральных числа  $l$  и  $r$  ( $1 \leq l \leq r \leq n$ ) — индексы левого и правого концов соответствующего подсписка.

### Формат выходных данных

Выведите  $t$  строк, где  $i$ -ая строка содержит единственное натуральное число — мощность подсписка  $i$ -го запроса.

### Примеры

стандартный ввод	стандартный вывод
3 2 1 2 1 1 2 1 3	3 6
8 3 1 1 2 2 1 3 1 1 2 7 1 6 2 7	20 20 20

## Задача Е. Переворот

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Дан массив. Надо научиться обрабатывать два типа запросов.

- 1 L R - перевернуть отрезок [L, R]
- 2 L R - найти минимум на отрезке [L, R]

### Формат входных данных

Первая строка файла содержит два числа  $n, m$ . ( $1 \leq n, m \leq 10^5$ ) Во второй строке находится  $n$  чисел  $a_i$  ( $1 \leq a_i \leq 10^9$ )- исходный массив. Остальные  $m$  строк содержат запросы, в формате описанном в условии. Для чисел L,R выполняется ограничение ( $1 \leq L \leq R \leq n$ ).

### Формат выходных данных

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

### Пример

стандартный ввод	стандартный вывод
10 7	3
5 3 2 3 12 6 7 5 10 12	2
2 4 9	2
1 4 6	2
2 1 8	
1 1 8	
1 8 9	
2 1 7	
2 3 6	

## Задача F. Различные числа

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

Сколько различных чисел на отрезке массива?

### Формат входных данных

На первой строке длина массива  $n$  ( $1 \leq n \leq 300\,000$ ). На второй строке  $n$  целых чисел от 0 до  $10^9 - 1$ . На третьей строке количество запросов  $q$  ( $1 \leq q \leq 300\,000$ ). Следующие  $q$  строк содержат описание запросов, по одному на строке. Каждый запрос задаётся парой целых чисел  $l, r$  ( $1 \leq l \leq r \leq n$ ).

### Формат выходных данных

Выведите ответы на запросы по одному в строке.

### Пример

стандартный ввод	стандартный вывод
5	3
1 1 2 1 3	2
3	3
1 5	
2 4	
3 5	

## Задача G. Своппер

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

Современные компьютеры зацикливаются  
в десятки раз эффективнее человека

Рекламный проспект OS Vista-N

Перед возвращением в штаб-квартиру корпорации Аазу и Скиву пришлось заполнить на местной таможене декларацию о доходах за время визита. Получилась довольно внушительная последовательность чисел. Обработка этой последовательности заняла весьма долгое время.

- Своппер кривой, — со знанием дела сказал таможенник.
- А что такое своппер? — спросил любопытный Скив.

Ааз объяснил, что своппер — это структура данных, которая умеет делать следующее.

- Взять отрезок чётной длины от  $x$  до  $y$  и поменять местами число  $x$  с  $x + 1$ ,  $x + 2$  с  $x + 3$ , и т.д.
- Посчитать сумму чисел на произвольном отрезке от  $a$  до  $b$ .

Учитывая, что обсчёт может затянуться надолго, корпорация «МИФ» попросила Вас решить проблему со своппером и промоделировать ЭТО эффективно.

### Формат входных данных

Во входном файле заданы один или несколько тестов. В первой строке каждого теста записаны число  $N$  — длина последовательности и число  $M$  — число операций ( $1 \leq N, M \leq 100\,000$ ). Во второй строке теста содержится  $N$  целых чисел, не превосходящих  $10^6$  по модулю — сама последовательность. Далее следуют  $M$  строк — запросы в формате 1  $x_i$   $y_i$  — запрос первого типа, и 2  $a_i$   $b_i$  — запрос второго типа. Сумма всех  $N$  и  $M$  по всему файлу не превосходит 200 000. Файл завершается строкой из двух нулей. Гарантируется, что  $x_i < y_i$ , а  $a_i \leq b_i$ .

### Формат выходных данных

Для каждого теста выведите ответы на запросы второго типа, как показано в примере. Разделяйте ответы на тесты пустой строкой.

### Пример

стандартный ввод	стандартный вывод
5 5	Swapper 1:
1 2 3 4 5	10
1 2 5	9
2 2 4	2
1 1 4	
2 1 3	
2 4 4	
0 0	

## Задача Н. Варенье

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

Мальш и Карлсон решили пойти на прогулку. Они знают, что прогулка будет совсем скучной, если перед ней не опустошить несколько банок варенья.

Мальш достал из кладовки  $N$  банок варенья и выставил их в ряд. В банке номер  $i$  содержится ровно  $a_i$  грамм варенья. Карлсон немного подумал и решил, что в некоторых банках недостаточно варенья, и что в банке номер  $i$  должно быть хотя бы  $b_i$  грамм варенья.

Выходить из этой ситуации Карлсон хочет в  $M$  этапов. На каждом этапе он выбирает числа  $l$ ,  $r$ ,  $x$  и  $y$  а затем выполняет следующие операции: в банку номер  $l$  он добавляет  $x$  грамм варенья, в банку номер  $l + 1 - x + y$  грамм варенья, в банку номер  $l + 2 - x + y \cdot 2$ , и так далее. В банку номер  $r$  наш герой добавит  $x + y \cdot (r - l)$  грамм варенья.

Мальшу хочется определить для каждой банки  $i$  наименьший номер операции, после которой в ней станет хотя бы  $b_i$  грамм варенья. Помогите Мальшу: найдите соответствующее число для каждой банки.

### Формат входных данных

В первой строке входного файла задано одно число  $N$  ( $1 \leq n \leq 10^5$ ) — количество банок. Во второй строке заданы  $N$  чисел  $a_i$  ( $0 \leq a_i \leq 2 \cdot 10^9$ ) — изначальное количество варенья в банке номер  $i$ . В третьей строке заданы  $N$  чисел  $b_i$  ( $0 \leq b_i \leq 2 \cdot 10^9$ ) — минимальное количество варенья, которое должно быть в банке номер  $i$ .

В четвертой строке задано  $M$  ( $0 \leq M \leq 10^5$ ) — число этапов добавления варенья в банки, которые выполнит Карлсон. В следующих  $M$  строках описаны сами этапы в хронологическом порядке. Каждый этап задан четырьмя числами  $l$ ,  $r$ ,  $x$  и  $y$  ( $1 \leq l \leq r \leq N$ ,  $0 \leq x, y \leq 3 \cdot 10^5$ ).

### Формат выходных данных

Выведите  $N$  чисел в одной строке, разделенные пробелом. Число номер  $i$  должно быть равно нулю, если в банке номер  $i$  изначально было достаточно варенья, номеру этапа, после которого в ней станет хотя бы  $b_i$  варенья, или  $-1$ , если даже после выполнения всех этапов, в этой банке будет недостаточно варенья. Этапы нумеруются с единицы.

### Пример

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



## Задача I. Перестановка и запросы

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

Дана перестановка  $p$  из  $n$  элементов. Перестановка из  $n$  элементов — это массив длины  $n$ , в котором каждое целое число от 1 до  $n$  встречается ровно по одному разу. Например,  $[1, 2, 3]$  и  $[4, 3, 5, 1, 2]$  — это перестановки, но  $[1, 2, 4]$  и  $[4, 3, 2, 1, 2]$  — это не перестановки. Вам нужно выполнить  $q$  запросов.

Есть два типа запросов:

- $1\ x\ y$  — поменять местами  $p_x$  и  $p_y$ .
- $2\ i\ k$  — вывести число, которым станет  $i$ , если присвоить  $i = p_i\ k$  раз.

### Формат входных данных

В первой строке находятся два целых числа  $n$  и  $q$  ( $1 \leq n, q \leq 10^5$ ).

Во второй строке находятся  $n$  целых чисел  $p_1, p_2, \dots, p_n$ .

В каждой из следующих  $q$  строк находится по три целых числа. Первое число  $t$  ( $1 \leq t \leq 2$ ) — тип запроса. Если  $t = 1$ , то следующие два целых числа — это  $x$  и  $y$  ( $1 \leq x, y \leq n; x \neq y$ ) — запрос первого типа. Если  $t = 2$ , то следующие два целых числа — это  $i$  и  $k$  ( $1 \leq i, k \leq n$ ) — запрос второго типа.

Гарантируется, что есть хотя бы один запрос второго типа.

### Формат выходных данных

Для каждого запроса второго типа выведите одно целое число в новой строке — ответ на этот запрос.

### Примеры

стандартный ввод	стандартный вывод
5 4	4
5 3 4 2 1	1
2 3 1	2
2 1 2	
1 1 3	
2 1 2	
5 9	3
2 3 5 1 4	5
2 3 5	4
2 5 5	2
2 5 1	3
2 5 3	3
2 5 4	3
1 5 4	1
2 5 3	
2 2 5	
2 5 1	

### Замечание

В первом примере  $p = \{5, 3, 4, 2, 1\}$ .

Первый запрос — вывести  $p_3$ . Ответ — 4.

Второй запрос — вывести  $p_{p_1}$ . Ответ — 1.

Третий запрос — поменять местами  $p_1$  и  $p_3$ . Теперь  $p = \{4, 3, 5, 2, 1\}$ .

Четвёртый запрос — вывести  $p_{p_1}$ . Ответ — 2.

## Задача J. Жесть

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 15 секунд  
Ограничение по памяти: 256 мегабайт

Дан массив из  $N$  чисел. Нужно уметь обрабатывать 3 типа запросов:

- `get(L, R, x)` — сказать, сколько элементов отрезка массива  $[L..R]$  не меньше  $x$ .
- `set(L, R, x)` — присвоить всем элементам массива на отрезке  $[L..R]$  значение  $x$ .
- `reverse(L, R)` — перевернуть отрезок массива  $[L..R]$ .

### Формат входных данных

Число  $N$  ( $1 \leq N \leq 5 \cdot 10^5$ ) и массив из  $N$  чисел. Далее число запросов  $M$  ( $1 \leq M \leq 5 \cdot 10^5$ ) и  $M$  запросов. Формат описания запросов предлагается понять из примера. Для всех отрезков верно  $1 \leq L \leq R \leq N$ . Исходные числа в массиве и числа  $x$  в запросах — целые от 0 до  $10^9$ .

### Формат выходных данных

Для каждого запроса типа `get` нужно вывести ответ.

### Пример

стандартный ввод	стандартный вывод
5	3
1 2 3 4 5	1
6	3
get 1 5 3	1
set 2 4 2	
get 1 5 3	
reverse 1 2	
get 2 5 2	
get 1 1 2	

## Задача К. МЕХ на пути

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Дано дерево, на каждом ребре которого написано неотрицательное целое число. Вам необходимо ответить на несколько запросов вида «для данных вершин  $u$ ,  $v$  назовите наименьшее неотрицательное целое число, которое **не** встречается среди чисел, написанных на ребрах на пути от  $u$  до  $v$ ».

### Формат входных данных

Первая строка входных данных содержит два числа  $n$  и  $q$  ( $2 \leq n \leq 10^5$ ,  $1 \leq q \leq 10^5$ ), количество вершин и количество запросов.

Следующие  $n - 1$  строк содержат по три числа  $u_i, v_i, x_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ,  $0 \leq x_i \leq 10^9$ ), которые описывают ребро дерева  $(u_i, v_i)$ , на котором написано число  $x_i$ .

Следующие  $q$  строк содержат по паре чисел  $a_j, b_j$  ( $1 \leq a_j, b_j \leq n$ ), которая обозначает запрос на пути от  $a_j$  до  $b_j$ .

### Формат выходных данных

Для каждого запроса выведите единственное число — минимальное число, которое не встречается на пути.

### Пример

стандартный ввод	стандартный вывод
7 6	0
2 1 1	1
3 1 2	2
1 4 0	2
4 5 1	3
5 6 3	3
5 7 4	
1 3	
4 1	
2 4	
2 5	
3 5	
3 7	

## Задача L. Машинное обучение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

На курсе машинного обучения вам выдали первое домашнее задание — вам предстоит проанализировать некоторый массив из  $n$  чисел.

В частности, вы интересуетесь так называемой *равномерностью* массива. Предположим, что в массиве число  $b_1$  встречается  $k_1$  раз,  $b_2$  —  $k_2$  раз, и т.д. Тогда *равномерностью* массива называется такое минимальное целое число  $c \geq 1$ , что  $c \neq k_i$  для любого  $i$ .

В рамках вашего исследования вы хотите последовательно проделать  $q$  операций.

- Операция  $t_i = 1, l_i, r_i$  задаёт запрос исследования. Необходимо вывести равномерность массива, состоящего из элементов на позициях от  $l_i$  до  $r_i$  включительно.
- Операция  $t_i = 2, p_i, x_i$  задаёт запрос уточнения данных. Начиная с этого момента времени  $p_i$ -му элементу массива присваивается значения  $x_i$ .

### Формат входных данных

Первая строка содержит  $n$  и  $q$  ( $1 \leq n, q \leq 100\,000$ ) — размер массива и число запросов соответственно.

Во второй строке записаны ровно  $n$  чисел —  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

Каждая из оставшихся  $q$  строк задаёт очередной запрос.

Запрос первого типа задаётся тремя числами  $t_i = 1, l_i, r_i$ , где  $1 \leq l_i \leq r_i \leq n$  — границы соответствующего отрезка.

Запрос второго типа задаётся тремя числами  $t_i = 2, p_i, x_i$ , где  $1 \leq p_i \leq n$  — позиция в которой нужно заменить число, а  $1 \leq x_i \leq 10^9$  — его новое значение

### Формат выходных данных

Для каждого запроса первого типа выведите одно число — равномерность соответствующего отрезка массива.

### Пример

стандартный ввод	стандартный вывод
10 4	2
1 2 3 1 1 2 2 2 9 9	3
1 1 1	2
1 2 8	
2 7 1	
1 2 8	

### Замечание

Первый запрос состоит из ровно одного элемента — 1. Минимальное подходящее  $c = 2$ .

Отрезок второго запроса состоит из четырёх 2, одной 3 и двух 1. Минимальное подходящее  $c = 3$ .

Отрезок четвёртого запроса состоит из трёх 1, трёх 2 и одной 3. Минимальное подходящее  $c = 2$ .