

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

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

Компания Gigggle 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  $i$ -м 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 В. 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 С. Стратегии [В']

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	10 секунд
Ограничение по памяти:	512 мегабайт

Глеб очень любит в свободное время решать задачи и играть в пошаговые стратегии. Вот и сейчас Глеб играет в одну очень известную стратегию. Сейчас Глеб начинает проходить новый уровень, в его армии целых  $N$  копеечников(и он очень гордится этим). Как известно у любого копеечника есть сила  $a_i$  и здоровье  $b_i$ . Уровень представляет собой шеренгу из  $M$  мечников. Как и копеечники, мечники имеют атаку  $c_i$  и запас здоровья  $d_i$ . Прохождение уровня выглядит следующим образом: сначала первый копеечник сражается с первым мечником. Если умирает мечник, то этот же копеечник сражается со следующим мечником и так далее, пока не погибнет сам, затем вступает в бой следующий копеечник. Так как лечения в этой игре пока нет, то следующий бой и мечники и копеечники начинают с количеством здоровья оставшимся после уже прошедших боев.

Сражение состоит в одновременном обмене ударами. Когда один из них атакует, он уменьшает здоровье противника на величину равную атаке. Когда запас здоровья одного из сражающихся становится неположительным, он умирает и сражение прекращается. Также возможно, что оба противника умрут одновременно.

Саша, друг Глеба, подарил ему артефакт, который позволяет узнать, сколько мечников убьёт его копеечник, если он пойдёт сражаться первым. Глеб решил доказать, что он и сам сможет справиться с этой задачей, но так как у него впереди коллоквиум, то он попросил вас ему помочь и для каждого копеечника его армии сказать, сколько мечников он убьёт.

### Формат входных данных

В первой строке записаны два натуральных числа  $n$  и  $m$  — количество копеечников в армии Глеба и количество мечников на уровне ( $1 \leq n, m \leq 200\,000$ ).

Затем идут  $N$  строк,  $i$ -ая из них содержит два целых числа  $a_i$  и  $b_i$ , задающих атаку и здоровье  $i$ -го копеечника ( $1 \leq a_i, b_i \leq 10^9$ ).

Следующие  $M$  строк содержат описание мечников в порядке, в котором герой с ними сталкивается. Каждое описание состоит из двух целых чисел  $c_i$  и  $d_i$ , задающих атаку и здоровье  $i$ -го мечника ( $1 \leq c_i, d_i \leq 200\,000$ ).

### Формат выходных данных

Выведите  $N$  строк.  $i$ -я строка должна содержать одно число - сколько мечников убьёт  $i$ -й копеечник.

### Пример

стандартный ввод	стандартный вывод
5 3	0
1 2	1
2 2	2
10 10	3
100 10	3
1 100	
2 2	
7 2	
3 20	

## Задача D. 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 Е. Посчитайте манулов

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

Дети из кружка Тинькофф Поколение любят считать манулов в различных чатиках. В этой задаче вам тоже придётся считать манулов.

Всего есть  $n$  детей, некоторые из которых дружат друг с другом. Формально, заданы  $m$  различных неупорядоченных пар  $(v, u)$ , таких что ребёнок с номером  $v$  дружит с ребёнком  $u$ .

Изначально,  $i$ -й ребёнок уже досчитал до  $a_i$  манулов.

Происходят  $q$  событий, каждое из которых имеет один из двух видов:  $(1 \leq v \leq n; 0 \leq x \leq 10^4)$

- $+ v x$ : Ребёнок с номером  $v$  отправляет по  $x$  стикеров с манулом, каждому из своих друзей. Каждый из его друзей сразу же считает этих полученных манулов.
- $? v$ : Мы просим вас посчитать, сколько манулов на данный момент уже посчитал ребёнок с номером  $v$

### Формат входных данных

В первой строке даны три числа  $n$ ,  $m$  и  $q$  ( $1 \leq n \leq 10^5; 0 \leq m \leq 10^5; 1 \leq q \leq 3 \cdot 10^5$ )

Во второй строке заданы  $n$  чисел,  $i$ -е из которых —  $a_i$  ( $0 \leq a_i \leq 10^9$ )

В каждой из следующих  $m$  строк заданы два числа  $v, u$  ( $1 \leq v \neq u \leq n$ ) — пары друзей. Гарантируется, что пары не повторяются.

В каждой из следующих  $q$  строк описаны события в описанном формате. Гарантируется, что будет хотя бы одно событие типа  $?$ .

### Формат выходных данных

Для каждого события типа  $?$  выведите в отдельной строке требуемое значение.

### Пример

стандартный ввод	стандартный вывод
5 5 5	1
1 2 3 4 5	5
1 2	1
2 3	7
3 4	
4 5	
5 1	
? 1	
? 5	
+ 1 2	
? 1	
? 5	

## Задача F. Сумма цифр

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

Для целых чисел  $b$  и  $n$  определим функцию  $f(b, n)$  так:

- Если  $n < b$ , то  $f(b, n) = n$
- Иначе  $f(b, n) = f(b, \text{floor}(n/b)) + (n \bmod b)$

Здесь  $\text{floor}(n/b)$  означает максимальное целое число не превышающее  $n/b$ , а  $n \bmod b$  означает остаток от деления  $n$  на  $b$ .

Менее формально  $f(n, b)$  — сумма цифр числа  $n$  в записи в системе с основанием  $b$ .

Так например:

- $f(10, 87654) = 8 + 7 + 6 + 5 + 4 = 30$
- $f(100, 87654) = 8 + 76 + 54 = 138$

Вам даны целые числа  $n$  и  $s$ . Найдите минимальное такое  $b \geq 2$ , такое что  $f(b, n) = s$ . Если такого  $b$  не существует, выведите  $-1$ .

### Формат входных данных

Первая строка входных данных содержит целое число  $n$  ( $1 \leq n \leq 10^{11}$ ) Вторая строка входных данных содержит целое число  $s$  ( $1 \leq s \leq 10^{11}$ )

### Формат выходных данных

В единственной строке выходных данных выведите ответ на задачу.

### Примеры

стандартный ввод	стандартный вывод
1 31415926535	-1
87654 30	10
87654 138	100
87654 45678	-1
31415926535 1	31415926535

## Задача Г. 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 Н. 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 5 3 4 2 1 2 3 1 2 1 2 1 1 3 2 1 2	4 1 2
5 9 2 3 5 1 4 2 3 5 2 5 5 2 5 1 2 5 3 2 5 4 1 5 4 2 5 3 2 2 5 2 5 1	3 5 4 2 3 3 3 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 I. Варенье

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

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

Малыш 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.

Малышу хочется определить для каждой банки  $i$  наименьший номер операции, после которой в ней станет хотя бы  $b_i$  грамм варенья. Помогите Малышу: 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 5 4 4 2 1 7 7 4 7 7 3 1 2 2 0 2 5 1 1 3 4 2 2	1 2 0 3 -1

## Задача J. Musaigen no Phantom World

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

В далекой галактике существует загадочный мир, известный как «Призрачный мир». В этом мире существуют различные существа, называемые «Призрачными». Каждый Призрачный обладает уникальными способностями и приоритетом.

В центре Призрачного мира находится древнее дерево, известное как «Высший Разум». Это существо является связующим звеном для всех Призрачных. Каждый Призрачный представлен вершиной в графе, где ребра представляют собой связи между Призрачными.

Но связи между Призрачными не так просты, как кажется. Каждая связь имеет свой приоритет, который определяет силу и значимость этой связи. Чем выше приоритет, тем сильнее связь.

Изначально, между Призрачными нет никаких связей.

В Призрачном мире существуют два типа запросов, которые могут быть сделаны к Высшему Разуму:

1. Запрос на добавление связи:  $+ a b k$ . Этот запрос добавляет новую связь между двумя Призрачными существами с приоритетом  $k$ . Благодаря этому запросу, новая связь будет создана между Призрачными существами  $a$  и  $b$  с приоритетом  $k$ .

2. Запрос на поиск минимального приоритета:  $? p$ . Этот запрос позволяет найти минимальный неотрицательный приоритет  $k$ , при котором ребра с приоритетом меньше или равном  $k$ , образуют ровно  $p$  компонент связности в графе Призрачного мира. Если такого приоритета не существует, выводится значение  $-1$ .

Таким образом, игрокам предстоит использовать эти запросы, чтобы управлять связями в Призрачном мире и находить определенное количество компонент связности с заданным приоритетом. Это поможет им раскрыть тайны и силу Призрачного мира, и, возможно, даже спасти его от опасности.

### Формат входных данных

Первая строка входных данных содержит два целых числа  $n$  — количество вершин и  $q$  — количество запросов ( $1 \leq n \leq 3000$ ,  $0 \leq q \leq 10^5$ )

Далее в следующих  $q$  строчках идет описание запросов. Описание каждого запроса имеет следующий вид:

- Запрос первого типа задается в виде:  $+ a b k$  ( $1 \leq a, b \leq n$ ,  $1 \leq k \leq 10^9$ ). Гарантируется, что никакие два ребра не имеют одинаковый приоритет.
- Запрос второго типа задается в виде:  $? p$  ( $1 \leq p \leq n$ )

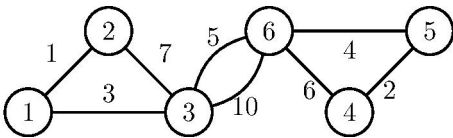
### Формат выходных данных

Выведите ответ на каждый запрос второго типа

Пример

стандартный ввод	стандартный вывод
6 12	-1
+ 1 2 1	0
+ 1 3 3	4
+ 2 3 7	5
+ 4 5 2	
+ 5 6 4	
? 1	
+ 6 3 5	
+ 6 3 10	
+ 6 4 6	
? 6	
? 2	
? 1	

Замечание



Итоговый граф в примере

## Задача К. 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 L. Удалённые рёбра

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

Дан полный неориентированный граф. Некоторые его рёбра удалены. Ваша задача — отвечать на запросы «каков кратчайший путь между двумя вершинами в оставшемся графе».

### Формат входных данных

Первая строка входных данных содержит два целых числа  $N$  ( $1 \leq N \leq 10^4$ ) и  $M$  ( $1 \leq M \leq N\sqrt{N}$ ) — количество вершин в полном графе и количество удалённых рёбер соответственно.

Каждая из следующих  $M$  строк содержит два целых числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq N$ ,  $a_i \neq b_i$ ) — номера вершин, соединённых очередным удалённым ребром.

Далее следует целое число  $Q$  — количество запросов ( $1 \leq Q \leq 10^6$ ). Каждая из последующих  $Q$  строк содержит по два целых числа  $x_i$  и  $y_i$  — номера вершин, кратчайший путь между которыми требуется найти ( $1 \leq x_i, y_i \leq N$ ,  $x_i \neq y_i$ ).

### Формат выходных данных

Для каждого запроса в отдельной строке выведите длину кратчайшего пути или  $-1$ , если пути не существует.

### Система оценки

Задача состоит из четырёх подзадач.

- В первой подзадаче  $N, Q \leq 700$ . Эта подзадача оценивается в 6 баллов.
- Во второй подзадаче  $N \leq 700$ ,  $Q \leq 3 \cdot 10^4$ . Эта подзадача оценивается в 5 баллов.
- В третьей подзадаче  $N \leq 2500$ ,  $Q \leq 2 \cdot 10^5$ . Эта подзадача оценивается в 33 балла.
- В четвёртой подзадаче дополнительных ограничений нет. Эта подзадача оценивается в 56 баллов.

### Пример

стандартный ввод	стандартный вывод
6 10	3
0 2	1
5 0	0
4 1	-1
1 3	
1 5	
2 5	
2 4	
4 0	
5 3	
4 5	
4	
1 4	
3 2	
4 4	
0 5	

## Задача М. Странная сумма

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2.5 секунд  
Ограничение по памяти: 256 мегабайт

В декабре студенту Лёше нужно готовиться к экзаменам. Поэтому он решил за несколько дней повторить все курсы.

Сегодня Лёша решил сесть за подготовку к экзамену по физике. План на сегодня — выучить последовательность  $a$  из  $n$  целочисленных физических констант. К сожалению, поскольку в физике Лёша совсем ничего не понимает, он постоянно отвлекается на какие-то странные занятия. Например, он фиксирует некоторый подотрезок последовательности  $a_l, a_{l+1}, \dots, a_r$ , и считает для него странную сумму следующим образом:

1. Лёша перебирает все пары  $x, y$ , такие что  $l \leq x \leq y \leq r$ .
2. Для каждой пары  $x, y$  Лёша добавляет в странную сумму расстояние Хэмминга между отрезками последовательности  $a_x, a_{x+1}, \dots, a_y$  и  $a_l, a_{l+1}, \dots, a_{l+(y-x)}$ .

Расстоянием Хэмминга между последовательностями  $p_1, p_2, \dots, p_k$  и  $q_1, q_2, \dots, q_k$  называется число индексов  $i$ , таких что  $p_i \neq q_i$ .

Ваша задача — помочь Лёше поскорее посчитать странные суммы для всех интересующих его отрезков  $[l, r]$ , чтобы он поскорее вернулся к учёбе.

### Формат входных данных

В первой строке даны два числа  $n$  и  $q$  ( $1 \leq n, q \leq 2 \cdot 10^5$ ) — длина последовательности и количество отрезков, для которых нужно посчитать странную сумму.

В следующей строке записаны числа  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^6$ ) — имеющуюся у Лёши последовательность.

После этого следует  $q$  строк запросов, в каждой строке находятся числа  $l_i$  и  $r_i$  ( $1 \leq l_i \leq r_i \leq n$ ), означающие, что для отрезка с  $l_i$ -го по  $r_i$ -й символ включительно Лёше нужно посчитать странную сумму.

### Формат выходных данных

В  $i$ -й строке выведите число, равное странной сумме на отрезке из запроса под номером  $i$ .

### Примеры

стандартный ввод	стандартный вывод
4 4 1 2 1 3 1 1 2 3 2 4 1 4	0 1 4 8
7 5 2 1 5 6 6 2 3 1 4 4 7 4 4 1 3 3 5	10 7 0 4 3

### Замечание

Рассмотрим последний запрос в первом примере. В нём последовательность Лёши равна  $[1, 2, 1, 3]$ , а также выбран подотрезок с 1 до 4.

Обозначим за  $h$  расстояние Хэмминга. Тогда ответом является сумма следующих величин:

- $h([1, 2, 1, 3], [1, 2, 1, 3]) = 0$
- $h([1, 2, 1], [1, 2, 1]) = 0$
- $h([2, 1, 3], [1, 2, 1]) = 3$
- $h([1, 2], [1, 2]) = 0$
- $h([2, 1], [1, 2]) = 2$
- $h([1, 3], [1, 2]) = 1$
- $h([1], [1]) = 0$
- $h([2], [1]) = 1$
- $h([1], [1]) = 0$
- $h([3], [1]) = 1$

Итоговая сумма равна 8.