

## Задача А. Лунки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.25 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. Запросы композиции перестановок

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Вам дан массив  $a_1, \dots, a_n$ , состоящий из перестановок длины  $m$ .

Мы можем определить операцию  $\cdot$  для двух перестановок  $x$  и  $y$  длины  $m$  как такую перестановку  $z = x \cdot y$ , что  $z_i = y_{x_i}$  для всех  $1 \leq i \leq m$ . Заметьте, что порядок перемножения важен.

Вам даны  $q$  запросов, каждый запрос задается двумя числами  $1 \leq l \leq r \leq n$ . Рассмотрим перестановку  $b$  длины  $m$ , такую что  $b = ((\dots((a_l \cdot a_{l+1}) \cdot a_{l+2}) \cdot \dots) \cdot a_r)$ . Тогда ответом на запрос будет являться сумма  $\sum_{i=1}^m i \cdot b_i$ . Реализуйте программу, быстро отвечающую на эти запросы.

### Формат входных данных

Первая строка содержит единственное целое число  $t$  равное количеству тестовых случаев ( $1 \leq t \leq 1000$ ). Далее следует описание  $t$  тестовых случаев, каждое в следующем формате:

Первая строка каждого описания содержит два целых числа  $n, m$  ( $1 \leq n, m \leq 10^5$  и  $1 \leq n \cdot m \leq 2 \cdot 10^5$ ). Следующие  $n$  строк содержат по  $m$  различных целых чисел  $a_{i1}, a_{i2}, \dots, a_{im}$ , разделенных пробелами ( $1 \leq a_{ij} \leq m$ ). Следующая строка содержит единственное целое число  $q$  ( $1 \leq q \leq 2 \cdot 10^5$ ). Следующие  $q$  строк содержат по два целых числа  $l, r$ , разделенных пробелами ( $1 \leq l \leq r \leq n$ ).

Гарантируется, что сумма  $n \cdot m$  и сумма  $q$  по всем тестовым случаям не превосходит  $2 \cdot 10^5$ .

### Формат выходных данных

Выведите ответы на запросы в том порядке, к которому они заданы во входных данных.

### Пример

стандартный ввод	стандартный вывод
1	10
4 3	11
3 2 1	11
1 3 2	14
1 2 3	11
2 3 1	
5	
1 1	
1 4	
3 4	
3 3	
1 3	

## Задача С. Варенье

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
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  $l, r, x, y$  а затем выполняет следующие операции: в банку номер  $l$  он добавляет  $x$  грамм варенья, в банку номер  $l + 1$  добавляет  $x + y$  грамм варенья, в банку номер  $l + 2$  —  $x + 2 \cdot y$ , 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  $l, r, x, y$  ( $1 \leq l \leq r \leq n, 0 \leq x, y \leq 10^5$ ).

### Формат выходных данных

Выведите  $n$  чисел в одной строке, разделенные пробелом. Число номер  $i$  должно быть равно 0, 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 D. Потoki

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

В городе Рязань весьма специфический водопровод. Он представляет из себя одну длинную трубу проходящую через  $N + 1$  узел, занумерованный от 0 до  $N$ . Каждый отсек трубы имеет пропускную способность не более  $K$  литров. Из-за технических работ, связанных с древнерусской традицией отключения горячей воды летом, коммунальным службам города иногда требуется начать перекачивать по литру воды в секунду от узла  $L$  до узла  $R$ . Так как пропускная способность трубы ограничена, с учётом всех предыдущих перекачек это не всегда возможно. Требуется на каждый запрос отвечать, можно ли пропустить поток воды между двумя узлами, и если это возможно, пустить её между этими узлами.

### Формат входных данных

В первой строке содержатся три числа  $N$  — количество узлов ( $1 \leq N \leq 200\,000$ ),  $K$  — максимальная пропускная способность каждого отсека трубы ( $1 \leq K \leq 1000$ ) и  $M$  — количество запросов ( $1 \leq M \leq 100\,000$ ). В следующих  $M$  строках описаны запросы, каждый из которых состоит из двух чисел  $L$  и  $R$  ( $0 \leq L < R \leq N$ ).

### Формат выходных данных

На каждый запрос ваша программа должна выдавать результат в виде числа 0 если поток пустить нельзя и 1, если это получилось. Каждый результат должен быть на отдельной строке

### Система оценки

Программа, верно работающая при  $N \leq 100$ , будет оцениваться в 30 баллов.

### Пример

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

## Задача Е. Треугольники

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

Дан граф. Найдите количество циклов длины 3.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 3 \cdot 10^5$ ) — количество вершин и рёбер, соответственно.

Каждая из следующих  $m$  строк содержит по два целых числа от 1 до  $n$  — вершины, которые соединяет соответствующее ребро.

Гарантируется, что в графе нет петель и кратных рёбер.

### Формат выходных данных

Выведите одно число — ответ.

### Пример

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

## Задача F. 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  $N$  копейщиков(и он очень гордится этим). Как известно у любого копейщика есть сила  $a_i$  и здоровье  $b_i$ . Уровень представляет собой шеренгу из  $M$  мечников. Как и копейщики, мечники имеют атаку  $c_i$  и запас здоровья  $d_i$ . Прохождение уровня выглядит следующим образом: сначала первый копейщик сражается с первым мечником. Если умирает мечник, то этот же копейщик сражается со следующим мечником и так далее, пока не погибнет сам, затем вступает в бой следующий копейщик. Так как лечения в этой игре пока нет, то следующий бой и мечники и копейщики 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 копейщик, 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 копейщика его армии сказать, сколько мечников он убьёт.

### Формат входных данных

В первой строке записаны два натуральных числа  $n$  и  $m$  — количество копейщиков 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 копейщика ( $1 \leq a_i, b_i \leq 10^9$ ).

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 копейщик.

### 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 G. Нестабильность сети

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

Вася — системный администратор в большой компании под названием Глюкософт. В сети компании  $n$  компьютеров, и какие-то пары компьютеров напрямую соединены сетевыми кабелями, всего таких соединений ровно  $m$ , при чем никакой кабель не соединяет компьютер с самим собой, и между любой парой компьютеров не больше одного кабеля.

На каждом компьютере в Глюкософте установлена специальная программа, поддерживаемая разработчиками Глюкософта, под названием СетеБаг. Новые версии этой программы выходят почти каждый день, но система обновлений в компании совершенно непостижима. Так, обновление версии программы происходит не синхронно для всех компьютеров. Это происходит следующим образом — выбирается один компьютер, и на него устанавливается какая-то версия СетеБага. Поскольку система обновлений настроена очень странным образом, вполне возможно, что на компьютер установят не последнюю версию СетеБага, а какую-то другую, например, ту что уже установлена на нем, или даже какую-то более раннюю.

После нескольких месяцев работы, Вася обнаружил, что наиболее частая причина нестабильной работы сети в несовместимости версий СетеБага двух компьютеров в Глюкософте. Если два компьютера, напрямую соединенных кабелем, имеют установленный Сетебаг разных версий, то соединение между этими компьютерами *нестабильно*: попытка передачи данных между ними может вызвать отказ системы. Чем больше нестабильных соединений между компьютерами, тем больше вероятность отказа системы. С другой стороны, обмен данными между компьютерами с одинаковой версией СетеБага обычно не приводит ни к каким ошибкам.

Задача Васи — предотвратить и исправить ошибки в сети Глюкософта. Сделать он это может только если он находится на рабочем месте целый день. Вася хочет быть на работе, когда вероятность отказа системы достаточно высока. Он хочет распланировать свое расписание на следующий год, но оценка вероятностей отказа системы — довольно сложная задача.

Помогите Васе найти число нестабильных соединений между компьютерами после каждого обновления версий СетеБага.

### Формат входных данных

В первой строке даны два целых числа  $n, m$  — количество компьютеров и соединений между ними ( $1 \leq n, m \leq 10^5$ ).

Во второй строке даны  $n$  целых чисел  $v_1, v_2, \dots, v_n$  — версии СетеБага, изначально установленные на компьютеры Глюкософта.

В следующих  $m$  строках даны пары целых чисел  $a_i, b_i$  — номера компьютеров, соединенных  $i$ -м кабелем ( $1 \leq a_i, b_i \leq n, a_i \neq b_i$ ). Гарантируется, что никакие два компьютера не соединены больше, чем одним кабелем.

В следующей строке дано целое число  $q$  — количество запланированных обновлений версий СетеБага ( $1 \leq q \leq 10^5$ ).

В следующих  $q$  строках даны пары целых чисел  $c_i, v_i$  — номер компьютера, на котором обновляется версия СетеБага, и новая версия СетеБага, которая установится на компьютер ( $1 \leq c_i \leq n, 1 \leq v_i' \leq 10^5$ ). Все обновления даны в хронологическом порядке, и никакие два обновления не происходят одновременно.

### Формат выходных данных

Выведите для каждого запроса изменения одно целое число — количество нестабильных соединений сразу после очередного обновления версии СетеБага.

**Примеры**

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



## Задача Н. RMQ

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

Реализуйте структуру данных, которая на данном массиве из  $N$  целых чисел позволяет узнать максимальное значение на этом массиве и индекс элемента, на котором достигается это максимальное значение.

### Формат входных данных

В первой строке вводится натуральное число  $N$  ( $1 \leq N \leq 10^5$ ) – количество элементов в массиве. В следующей строке содержатся  $N$  целых чисел, не превосходящих по модулю  $10^9$  – элементы массива. Гарантируется, что в массиве нет одинаковых элементов. Далее идет число  $K$  ( $0 \leq K \leq 10^5$ ) – количество запросов к структуре данных. Каждая из следующих  $K$  строк содержит два целых числа  $l$  и  $r$  ( $1 \leq l \leq r \leq N$ ) – левую и правую границы отрезка в массиве для данного запроса.

### Формат выходных данных

Для каждого из запросов выведите два числа: наибольшее значение среди элементов массива на отрезке от  $l$  до  $r$  и индекс одного из элементов массива, принадлежащий отрезку от  $l$  до  $r$ , на котором достигается этот максимум.

### Пример

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

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

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 J. Турбо-кузнечик

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

Один всем известный кузнечик как всегда прыгает по кочкам и собирает монетки. Всего кочек 30001 штука, они расположены в ряд и пронумерованы от 0 до 30000. Всего на этих кочках  $n$  монет, при чем  $i$ -я из них расположена на кочке номер  $p_i$ .

Кузнечик только что прибыл на 0-ю кочку и дальше будет прыгать по следующим правилам:

Сначала он прыгнет с кочки номер 0 на кочку номер  $d$ .

Затем он продолжит прыжки следующим образом: Пусть длина предыдущего прыжка равняется  $l$ , т. е., если его предыдущий прыжок был с кочки  $prev$  на кочку  $cur$ , то  $l = cur - prev$ . Следующим шагом он совершит прыжок вперед длины  $l - 1$ ,  $l$  или  $l + 1$ . Таким образом, он прыгнет на кочку  $(cur + l - 1)$ ,  $(cur + l)$  или  $(cur + l + 1)$  (конечно, в случае, если соответствующей кочки не существует, прыгнуть нельзя). Длины прыжков должны быть положительными, то есть, нельзя совершить прыжок длины 0, если сейчас  $l = 1$ . Если корректного пункта назначения нет, то прыжки прекращаются.

Кузнечик собирает монетки на кочках, которые он посещает в процессе. Найдите максимальное количество монеток, которые он может собрать.

### Формат входных данных

В первой следуют два целых числа через пробел,  $n$  и  $d$  ( $1 \leq n, d \leq 30000$ ), обозначающих количество монеток на кочка и длину первого прыжка кузнечика, соответственно.

В следующих  $n$  строках следуют расположения монеток. В  $i$ -й из них ( $1 \leq i \leq n$ ) записано целое число  $p_i$  ( $d \leq p_1 \leq p_2 \leq \dots \leq p_n \leq 30000$ ), обозначающее номер острова, содержащего  $i$ -ю монетку.

### Формат выходных данных

Выведите единственное число - максимальное число монеток, которые может собрать кузнечик

**Примеры**

стандартный ввод	стандартный вывод
4 10 10 21 27 27	3
8 8 9 19 28 36 45 55 66 78	6
13 7 8 8 9 16 17 17 18 21 23 24 24 26 30	4