

## Задача А. Сумма на отрезке

Имя входного файла: `sum.in`  
Имя выходного файла: `sum.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дан массив из  $N$  элементов, нужно научиться находить сумму чисел на отрезке.

### Формат входных данных

Первая строка входного файла содержит два целых числа  $N$  и  $K$  — количество чисел в массиве и количество запросов ( $1 \leq N \leq 100\,000$ ,  $0 \leq K \leq 100\,000$ ). Следующие  $K$  строк содержат следующие запросы:

1. `A i x` — присвоить  $i$ -му элементу массива значение  $x$  ( $1 \leq i \leq n$ ,  $0 \leq x \leq 10^9$ );
2. `Q l r` — найти сумму чисел в массиве на позициях от  $l$  до  $r$  ( $1 \leq l \leq r \leq n$ ).

Изначально в массиве живут нули.

### Формат выходных данных

На каждый запрос вида `Q l r` нужно вывести единственное число — сумму на отрезке.

### Примеры

<code>sum.in</code>	<code>sum.out</code>
5 9	0
A 2 2	2
A 3 1	1
A 4 2	2
Q 1 1	0
Q 2 2	5
Q 3 3	
Q 4 4	
Q 5 5	
Q 1 5	

### Замечание

TL для Python 4 секунды

## Задача В. Варенье

Имя входного файла: `jam.in`  
Имя выходного файла: `jam.out`  
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  $l + 2 - x + 2 \cdot y$ , 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.

### Примеры

jam.in	jam.out
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 С. Мощные юнги

Имя входного файла: `power.in`  
Имя выходного файла: `power.out`  
Ограничение по времени: 5 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

power.in	power.out
3 2	3
1 2 1	6
1 2	
1 3	
8 3	20
1 1 2 2 1 3 1 1	20
2 7	20
1 6	
2 7	

## Задача D. И снова сумма...

Имя входного файла: `sum2.in`  
Имя выходного файла: `sum2.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

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

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

### Формат входных данных

Исходно множество  $S$  пусто. Первая строка входного файла содержит  $n$  — количество операций ( $1 \leq n \leq 300\,000$ ). Следующие  $n$  строк содержат операции. Каждая операция имеет вид либо «+  $i$ », либо «?  $l$   $r$ ». Операция «?  $l$   $r$ » задает запрос  $sum(l, r)$ .

Если операция «+  $i$ » идет во входном файле в начале или после другой операции «+», то она задает операцию  $add(i)$ . Если же она идет после запроса «?», и результат этого запроса был  $y$ , то выполняется операция  $add((i + y) \bmod 10^9)$ .

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до  $10^9$ .

### Формат выходных данных

Для каждого запроса выведите одно число — ответ на запрос.

### Примеры

sum2.in	sum2.out
6	3
+ 1	7
+ 3	
+ 3	
? 2 4	
+ 1	
? 2 4	

## Задача Е. Суровый корректор

Имя входного файла: `corrector.in`  
Имя выходного файла: `corrector.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

По мнению Александра Павловича, текст необычайно красив, если некоторые особые слова (например, «коммунизм», «Ленин», «счастье») встречаются не слишком часто, но и не слишком редко, к тому же достаточно равномерно. Александр Павлович работает корректором. К нему поступают тексты, он имеет право их некоторым образом менять, после чего возвращает уже исправленную версию. В связи со своими воззрениями о красоте Александру Павловичу постоянно приходится проверять, сколько особых слов сейчас в той или иной части текста. Он настолько устал от рутинного подсчёта: «а сколько тут особых слов?», «а сколько тут?», что просит вас помочь ему автоматизировать этот процесс.

### Формат входных данных

Первая строка входного файла содержит текст, длина текста не превосходит  $10^5$ , в котором ищутся особые слова. Следующая строка содержит  $N$  ( $1 \leq N \leq 10^5$ ) — количество особых слов. Следующие  $N$  строк содержат особые слова. Все особые слова различны. Суммарная длина строк не превосходит  $10^5$ . В следующей строке дано  $Q$  ( $1 \leq Q \leq 10^5$ ) — количество интересных Александру Павловичу отрезков. Следующие  $Q$  строк содержат сами отрезки.

### Формат выходных данных

Выведите  $Q$  чисел — количества вхождений особых слов в соответствующий отрезок текста.

### Примеры

<code>corrector.in</code>	<code>corrector.out</code>
abacababa	5 0 2
2	
a	
aba	
3	
5 9	
2 2	
2 6	

## Задача F. Безумие и Отвага

Имя входного файла: `heroes.in`  
Имя выходного файла: `heroes.out`  
Ограничение по времени: 10 секунд  
Ограничение по памяти: 512 мегабайт

Многие из нас с детства мечтали создавать компьютерные игры, а для некоторых это даже стало причиной, по которой они начали изучать информатику и программирование. Мишина мечта сбылась, и теперь он работает в известной и уважаемой корпорации «Метель», выпустившей в своё время такие шедевры, как «Искусство войны» и «Звёздное ремесло».

Недавно Миша присоединился к проекту новой ролевой игры «Безумие и отвага». Её ключевой особенностью является возможность на каждом из уровней заново выбирать персонажа для его прохождения.

Перед стартом очередного уровня игроку доступны  $N$  героев. Каждый герой характеризуется силой атаки  $a_i$  и запасом здоровья  $b_i$ . Уровень представляет собой длинную пещеру, содержащую  $M$  монстров. Каждый монстр также имеет свою силу атаки  $c_i$  и запас здоровья  $d_i$ . Зайдя в пещеру, герой сначала сражается с первым монстром, затем, если остаётся жив, сражается со вторым и так далее, пока не погибнет или не дойдёт до конца. Количество жизней героя не восстанавливается между боями, то есть каждую следующую драку он начинает с меньшим запасом здоровья, чем предыдущую.

Бой между монстром и героем состоит в одновременном обмене ударами. Каждый из них, нанося удар, уменьшает запас здоровья противника на величину, равную силе своей атаки. Как только запас здоровья кого-либо из сражающихся становится неположительным, он умирает, и бой прекращается. Обратите внимание, что при такой схеме боя возможна ситуация, когда оба противника погибнут одновременно.

Компания планирует распространять игру бесплатно, получая доход за счёт продажи разнообразных бонусов, реализовать один из которых и поручено Мише. Данный бонус позволяет игроку узнать, сколько монстров убьёт каждый из героев, если игрок выберет именно его для прохождения данного уровня. Так как монстров и героев может быть очень много, Миша столкнулся со сложностями при вычислении необходимых значений и обратился за помощью к вам.

### Формат входных данных

В первой строке ввода записаны два целых числа  $N$  и  $M$  — количество доступных игроку героев и количество монстров в пещере соответственно ( $1 \leq N, M \leq 200\,000$ ).

Следующие  $N$  строк описывают героев. Каждая из них содержит два целых числа  $a_i$  и  $b_i$ , задающих силу атаки и запас здоровья  $i$ -го героя ( $1 \leq a_i, b_i \leq 10^9$ ).

Далее следуют  $M$  строк, описывающих находящихся в пещере монстров. Каждое описание состоит из двух целых чисел  $c_i$  и  $d_i$ , обозначающих параметры  $i$ -го монстра ( $1 \leq c_i, d_i \leq 200\,000$ ). Порядок расположения монстров в пещере совпадает с порядком их описания, то есть первым необходимо убить монстра, описанного в строке  $N + 2$ , а последним — в строке  $N + M + 1$ .

### Формат выходных данных

Выведите  $N$  чисел по одному в строке.  $i$ -я строка должна содержать ответ для  $i$ -го героя.

### Примеры

heroes.in	heroes.out
5 3	0
1 2	1
2 2	2
10 10	3
100 10	3
1 100	
2 2	
7 2	
3 20	

## Замечание

Бой между первым героем и первым монстром в пещере продлится один ход, после которого герой погибнет, а монстр останется в живых.

Параметры второго героя совпадают с параметрами первого монстра, поэтому они убьют друг друга на первом же ходу боя. Ответ для данного героя равен одному.

Если игрок выберет для прохождения уровня третьего героя, то после боя с первым монстром его запас здоровья будет равен восьми, а после боя со вторым — единице. Для убийства третьего монстра ему необходимо сделать два удара, но он умрёт после первой же его атаки.

У четвёртого героя столько же жизней, сколько и у третьего, но сила атаки гораздо больше, поэтому он пройдёт уровень полностью, хотя и погибнет в последней драке.

Пятый герой обладает минимально возможной силой атаки, но при этом у него большой запас здоровья, поэтому он сможет пройти весь уровень и остаться в живых. После первого боя его запас здоровья будет равен 96, после второго — 82, а в конце игры останется только 22.

## Задача G. MeX на пути дерева

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 10 секунд  
Ограничение по памяти: 256 мегабайт

Вам дано дерево из  $n$  вершин. Каждому ребру дерева сопоставлено неотрицательное целое число  $x_i$ . Ваша задача – ответить на  $q$  запросов.  $j$ -й из запросов – найти минимальное неотрицательное целое число  $y$ , отсутствующее на пути между вершинами  $a_j$  и  $b_j$ .

### Формат входных данных

Первая строка содержит числа  $n$  и  $q$  ( $2 \leq n \leq 10^5$ ,  $1 \leq q \leq 10^5$ ), количество вершин в дереве и число запросов.

Следующие  $n - 1$  строка содержит описания рёбер дерева, тройки целых чисел  $u_i, v_i, x_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ,  $0 \leq x_i \leq 10^9$ ).

Следующие  $q$  строк содержат пары целых чисел  $a_j, b_j$  ( $1 \leq a_j, b_j \leq n$ ), обозначающие запросы.

### Формат выходных данных

На каждый запрос  $a_j, b_j$  на отдельной строке выведите минимальное целое неотрицательное  $y$ , не лежащее на пути между  $a_j$  и  $b_j$ .

### Примеры

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	