

Задача 1. A+B

<i>Входной файл</i>	<code>aplusb.in</code> или ввод с клавиатуры
<i>Выходной файл</i>	<code>aplusb.out</code> или вывод на экран
<i>Ограничение по времени</i>	1 сек
<i>Ограничение по памяти</i>	256 МБ
<i>Максимальный балл за задачу</i>	300

Напишите программу, которая складывает два числа.

Формат входных данных

На первой и единственной строке входных данных находятся два числа, A и B . Числа целые и не превосходят по модулю 1000.

Формат выходных данных

Выведите одно число — сумму двух данных чисел.

Пример

<i>Входной файл</i>	<i>Выходной файл</i>
2 2	4
-4 2	-2

Задача 2. Цивилизация

Входной файл	civ.in или ввод с клавиатуры
Выходной файл	civ.out или вывод на экран
Ограничение по времени	0.8 сек
Ограничение по памяти	256 МБ
Максимальный балл за задачу	100

В компьютерной игре «Цивилизация-X» каждый игрок управляет империей, состоящей из нескольких городов. Некоторые города (как одного и того же игрока, так и разных игроков) могут быть соединены торговыми путями.

Основной задачей каждого игрока является захват городов других игроков. После того, как город захвачен, в нем начинаются беспорядки; на ликвидацию беспорядков в городе надо потратить некоторое количество золотых монет. Количество монет равно количеству городов, которые соединены с захваченным городом торговыми путями, и которые при этом принадлежат другим игрокам.

Обратите внимание, что между парой городов может быть несколько торговых путей, но все равно при расчете количества монет каждый город, соединенный с захваченным городом торговыми путями, учитывается только один раз.

Торговые пути двунаправленные, т.е. каждый путь учитывается при захвате любого из городов, которые соединены этим путем.

После захвата города все торговые пути сохраняются.

Например, если игрок А захватывает город, соединенный торговыми путями с семью городами, четыре из которых принадлежат игроку А, два — игроку В и один — игроку С, то игрок А должен будет потратить три монеты на ликвидацию беспорядков. Если после этого этот же город захватит игрок В, то на ликвидацию беспорядков он должен будет потратить 5 монет.

Чтобы победить в игре, надо захватить все города, которые есть в мире.

Будем считать, что игроки не строят новые города и не создают новые торговые пути. Рассмотрим некоторую конфигурацию городов и торговых путей. Ценой победы игрока назовем минимальное количество монет, которые ему придется потратить на ликвидацию беспорядков, если он хочет победить. Определите игрока, цена победы которого минимальна.

Формат входных данных

В первой строке входных данных находятся три целых числа N , M и K — количество городов, торговых путей и количество игроков ($1 \leq N, K \leq 100\,000$; $0 \leq M \leq 100\,000$).

Во второй строке находятся N целых чисел от 1 до K , i -е из которых равно номеру игрока, которому изначально принадлежит i -й город.

Далее следуют M строк, каждая из которых содержит два различных целых числа — номера городов, соединяемых очередным торговым путем.

Города и игроки нумеруются начиная с 1. Могут быть игроки, которым изначально не принадлежит ни один город.

Формат выходных данных

Выведите два числа — номер игрока, цена победы которого минимальна, и его цену победы. Если таких игроков несколько, выведите игрока с наименьшим номером.

Система оценивания

Подзадача 1, 30 баллов: $N, M, K \leq 20$.

Подзадача 2, 30 баллов: $N, M, K \leq 1000$.

Подзадача 3, 30 баллов: нет городов, соединенных более чем одним торговым путем.

Подзадача 4, 10 баллов: дополнительных ограничений нет.

Кроме того, в тестах общей стоимостью 30 баллов (распределенных по разным подзадачам) будет выполняться условие $K = 2$.

Пример

Входной файл	Выходной файл
4 5 3 2 2 3 1 1 2 1 4 3 4 2 3 3 2	2 1

Задача 3. Деревья на аллее

Входной файл	<code>trees.in</code> или ввод с клавиатуры
Выходной файл	<code>trees.out</code> или вывод на экран
Ограничение по времени	1 сек
Ограничение по памяти	256 МБ
Максимальный балл за задачу	100

Представим себе бесконечную аллею, которая идет вдоль оси OX и начинается в точке $X = 0$.

С каждой из сторон аллеи посажены деревья. С одной стороны деревья посажены в точках $A, A + B, A + 2 \cdot B$ и так далее, то есть в точках $A + N \cdot B$ для любого целого неотрицательного N . При этом $0 \leq A < B$. Аналогично, с другой стороны аллеи деревья посажены в точках $C, C + D, C + 2 \cdot D$ и так далее. При этом $0 \leq C < D$. Обратите внимание, что в некоторых точках может оказаться два дерева — по одному с каждой из сторон аллеи (см. пример 1).

Вася любит фотографировать деревья. Он хочет прогуляться по аллее и сфотографировать минимум K деревьев. Однако, он может фотографировать дерево, только если находится в одной точке с ним. Вася довольно ленив и хочет пройти вдоль аллеи наименьшее возможное расстояние. При этом он может начать движение в любой точке аллеи — он просто подъедет к ней на машине. Помогите Васе найти отрезок аллеи минимальной длины, который ему нужно пройти пешком, чтобы сфотографировать хотя бы K деревьев.

Формат входных данных

В первой строке заданы 4 целых числа A, B, C и D ($0 \leq A < B \leq 10^9, 0 \leq C < D \leq 10^9$).

Во второй строке задано одно натуральное число K ($1 \leq K \leq 10^9$).

Формат выходных данных

Выведите два целых числа L и R — концы отрезка аллеи, который нужно пройти Васе. Отрезок должен содержать хотя бы K деревьев и иметь минимальную возможную длину. Несмотря на то, что аллея бесконечная, вам нужно найти такие числа L и R , что $0 \leq L \leq R \leq 8 \cdot 10^{18}$. Гарантируется, что среди оптимальных ответов существует ответ, удовлетворяющий этим ограничениям. Если таких оптимальных ответов несколько, можно вывести любой из них.

Система оценивания

Подзадача 1, 25 баллов: все входные числа не превосходят 5000.

Подзадача 2, 25 баллов: все входные числа не превосходят 10^7 .

Подзадача 3, 50 баллов: дополнительных ограничений нет.

Пример

Входной файл	Выходной файл
1 2 3 5 4	1 5
0 2 1 2 6	1000 1005

Примечание: В первом примере Вася может сфотографировать дерево в точке 1, два дерева в точке 3 и дерево в точке 5. Другой возможный оптимальный вариант — пройти от точки 5 до точки 9, длина пути так же будет равна 4. Более короткого маршрута, содержащего хотя бы 4 дерева, не существует.

Во втором примере в каждой точке находится ровно по одному дереву (с одной стороны деревья на четных позициях, с другой — на нечетных). Любой маршрут длины 5 будет содержать 6 деревьев.

Задача 4. Счастливые моменты времени

Входной файл	lucky.in или ввод с клавиатуры
Выходной файл	lucky.out или вывод на экран
Ограничение по времени	1 сек
Ограничение по памяти	256 МБ
Максимальный балл за задачу	100

В обычной жизни для записи некоторого момента времени можно использовать разные форматы. В этой задаче мы будем использовать следующий формат: 'hh:mm DD.MM.Y' (без кавычек), где:

- hh — количество часов, $00 \leq hh \leq 23$;
- mm — количество минут, $00 \leq mm \leq 59$;
- DD — номер дня в месяце, $01 \leq DD \leq 31$;
- MM — номер месяца в году, $01 \leq MM \leq 12$;
- Y — номер года, $1000 \leq Y \leq 2 \cdot 10^9$.

Обратите внимание, что значения 'hh', 'mm', 'DD', 'MM' содержат **ровно по две** десятичных цифры, в том числе они могут содержать ведущие нули. В свою очередь, значение 'Y' записывается **без** ведущих нулей и может содержать от 4 до 10 цифр. Между временем ('hh:mm') и датой ('DD.MM.Y') находится ровно один пробел.

Например, первая Нижегородская городская олимпиада школьников по информатике началась 19 февраля 2005 года примерно в 10:00. Эта дата будет записана следующим образом: '10:00 19.02.2005'. Последняя минута 2021 года будет записана как '23:59 31.12.2021'.

Момент времени будем называть 'счастливым', если значения 'hh', 'mm', 'DD', 'MM' равны и совпадают с записью последних двух цифр года 'Y'. Например, моменты '08:08 08.08.2008' и '12:12 12.12.12312' — счастливые, а '01:02 03.04.2005' и '08:08 08.08.2018' — нет.

Вам даны два момента времени. Определите, сколько между ними (включительно) счастливых моментов времени.

Формат входных данных

В первой строке задан первый момент времени в соответствии с форматом, указанным в условии.

Во второй строке задан второй момент времени.

Гарантируется, что оба момента времени корректны. В частности, значение 'DD' не превосходит количества дней в месяце 'MM' с учетом високосности года 'Y'. Гарантируется, что второй момент времени наступает не раньше первого.

Формат выходных данных

Выведите одно число — количество счастливых моментов времени между первым и вторым включительно.

Система оценивания

Подзадача 1, 60 баллов: $Y < 10^5$ для обоих моментов времени.

Подзадача 2, 40 баллов: дополнительных ограничений нет.

Пример

Входной файл	Выходной файл
10:00 19.02.2005 12:00 30.01.2007	2
11:00 15.05.2021 16:00 15.05.2021	0
00:00 01.01.9999 01:01 01.01.10001	1

Примечание: В первом примере в указанном интервале два счастливых момента времени: '05:05 05.05.2005' и '06:06 06.06.2006'.

Во втором примере счастливых моментов времени нет — номер дня никогда не совпадет с последними цифрами года.

В третьем примере один счастливый момент времени — он соответствует второй строке входных данных.

Задача 5. Магические посохи

Входной файл	magic.in или ввод с клавиатуры
Выходной файл	magic.out или вывод на экран
Ограничение по времени	1.5 сек
Ограничение по памяти	256 МБ
Максимальный балл за задачу	100

В провинции Скайрим, а если быть точнее, в городе Вайтран вполне себе припеваючи поживал придворный маг Фаренгар Тайный Огонь, но недавно началась суматоха с возвращением драконов и появлением драконорождённого, обычные будни...

«Ходят слухи, что драконорождённый уже прибыл в Вайтран, интересно, кто это — может быть высокий или тёмный эльф, у которых есть врождённая способность к магии, а может Бретонец! Нас явно ждут разговоры о рунах и заклинаниях целыми днями...» — думал про себя Фаренгар. Надо ли говорить, как он удивился, когда увидел перед собой орка в тяжёлых доспехах, который с большим трудом разговаривал на человеческом.

Ярл¹ Вайтрана приказал Фаренгару собрать Драконорождённого в поход и выдать ему магические посохи. Но, как мы знаем, орки не отличаются интеллектом, поэтому каждым выданным Фаренгаром магическим посохом драконорождённый может ударить врага только один раз, после чего посох ломается, а врагу наносится урон, равный длине посоха. У Фаренгара изначально нет посохов, но есть N заклинаний, i -е из которых может создать посох веса w_i и длины $l_i - c_i \cdot k$ для любого неотрицательного k . К сожалению, каждое заклинание с каждым конкретным k можно использовать не более одного раза. (При этом разные заклинания с одним и тем же k использовать можно, но каждое не более одного раза.) Фаренгар — сильный маг, он может создать сколько угодно посохов (в рамках описанных ограничений), и даже физика для него не препятствие, он вполне может создавать посохи отрицательной длины, но есть ли в этом смысл...

Всего драконорождённый может нести на себе вес W , поэтому ваша задача помочь Фаренгару собрать драконорождённого в путь, создав ему такие посохи, чтобы он мог нанести суммарно как можно больше урона.

Формат входных данных

В первой строке входных данных содержатся два целых числа N и W — количество заклинаний и вес, который может выдержать драконорождённый ($1 \leq N \leq 200\,000$, $1 \leq W \leq 3000$).

В следующих N строках содержится по три числа w_i , l_i и c_i — параметры i -го заклинания ($1 \leq w_i \leq 3000$, $1 \leq l_i \leq 100\,000$, $1 \leq c_i \leq l_i$).

Формат выходных данных

Выведите максимальный урон, который драконорождённый сможет нанести всеми посохами, которые ему создаст Фаренгар.

Система оценивания

Подзадача 1, 10 баллов: $N \leq 400$, $W \leq 2500$.

Подзадача 2, 50 баллов: $N \leq 50000$.

Подзадача 3, 40 баллов: дополнительных ограничений нет.

Пример

Входной файл	Выходной файл
2 10 5 5 5 2 6 2	15
5 15 2 1 1 2 2 1 2 3 1 2 4 1 2 5 1	24
2 10 3 4 2 2 1 1	7

Примечание: В первом примере оптимально создать посох первого типа с $k = 0$ и два посоха второго типа с $k \in [0; 1]$, суммарный вес будет 9, а суммарный урон максимален.

Во втором примере оптимально создать посохи пятого типа с $k \in [0; 3]$, четвёртого типа с $k \in [0; 1]$ и третьего типа с $k = 0$, суммарный вес равен 14.

В третьем примере оптимально создать два посоха первого типа с $k \in [0; 1]$ и один посох второго типа с $k = 0$, все остальные возможные посохи будут иметь неположительную длину, поэтому они не улучшат ответ.

¹ Ярл (др.-сканд. Jarl) — один из высших титулов в иерархии в средневековой Скандинавии.

Задача 6. Проектор

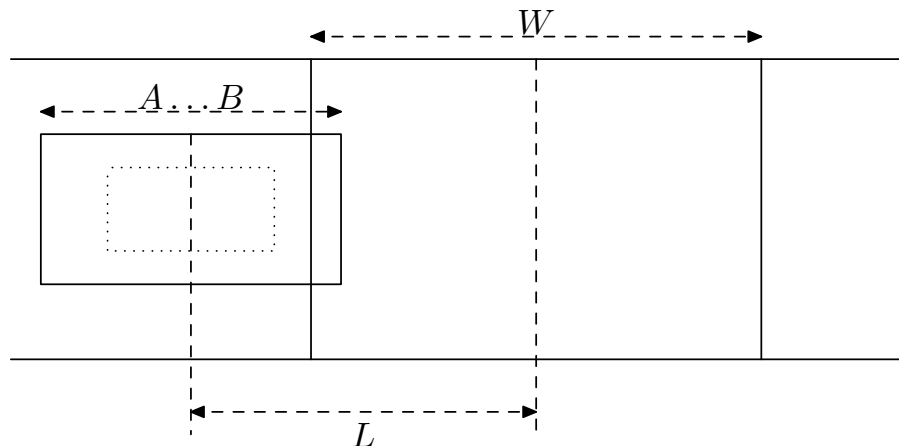
Входной файл	<code>projector.in</code> или ввод с клавиатуры
Выходной файл	<code>projector.out</code> или вывод на экран
Ограничение по времени	1 сек
Ограничение по памяти	256 МБ
Максимальный балл за задачу	100

В кабинете у Ивана Алексеевича рабочие повесили новый проектор. Проектор проецирует на стену изображение в виде прямоугольника со сторонами, направленными по горизонтали и по вертикали, при этом на проекторе есть колесико, позволяющее менять размер изображения — от A до B сантиметров по ширине (высота изображения нас интересоваться не будет). При изменении размера изображения центр изображения остается на месте, т.е. левая и правая границы изображения двигаются одновременно.

Также рабочие повесили на эту стену экран — длинную вертикальную полосу ширины W . Но, к сожалению, экран висит не совсем точно по центру изображения от проектора — расстояние по горизонтали между серединой экрана и серединой изображения равно L сантиметров. Экран закреплен на рельсах, благодаря которым его можно двигать влево-вправо на расстояние не более чем Y сантиметров от исходного положения.

Помогите рабочим подвинуть экран и отрегулировать размер изображения так, чтобы изображение полностью помещалось на экране, и при этом размер изображения был как можно больше.

Вам надо учитывать только положение изображения и экрана по горизонтали; считайте, что по вертикали изображение всегда гарантированно влезает на экран.



На рисунке показано изображение от проектора (в левой части рисунка, размером от A до B) и экран (в правой части рисунка, ширины W).

Формат входных данных

В единственной строке входных данных записаны числа A , B , W , L , Y . Все числа целые, неотрицательные, и не превосходят 10^{18} по модулю; гарантируется, что $A \leq B$.

Формат выходных данных

Выведите два неотрицательных целых числа — требуемую ширину изображения (от A до B включительно) в сантиметрах, и расстояние, на которое надо сдвинуть экран (от 0 до Y включительно).

Если решения не существует, выведите два раза число -1 . Если существует несколько решений с максимальной шириной изображения, выведите любое.

Обратите внимание, что вы должны выводить именно целые числа, хотя могут существовать и решения с дробными числами. Гарантируется, что если хотя бы какое-то решение существует, то существует и решение, выражающееся полностью целыми числами.

Система оценивания

Подзадача 1, 80 баллов: все числа во входном файле не превосходят 10^8 .

Подзадача 2, 20 баллов: нет дополнительных ограничений.

Пример

Входной файл	Выходной файл
5 15 20 10 30	15 8
5 15 20 100 30	-1 -1

Задача 7. Divan любит тыквы

Входной файл	pumpkins.in или ввод с клавиатуры
Выходной файл	pumpkins.out или вывод на экран
Ограничение по времени	6 сек
Ограничение по памяти	1024 МБ
Максимальный балл за задачу	100

Однажды стране Берляндии очень понадобились деньги, и было принято решение продать прекрасный парк с тыквами. К большому счастью правителей Берляндии, в мире есть ровно один бизнесмен, который может выкупить этот парк — Divan.

Узнав о таком заманчивом предложении, Divan первым делом поинтересовался, а что же представляет собой этот парк? Ему ответили, что это множество тыкв. Некоторые из них соединены между собой дорогами, да так, что между любой парой тыкв существует лишь один путь. (Строго говоря, парк — это дерево с тыквами в вершинах.) Всего парк состоит из n тыкв.

Такой ответ устроил бизнесмена, и обсуждение сделки продолжилось. Так как парк тыкв — не особо практичная вещь, Divan хочет, чтобы она была удивительной. Узнав план парка, Divan решил, что будет водить по нему своих друзей. Для этого он разработает несколько маршрутов. Каждый маршрут будет задаваться парой вершин (v, u) . Это значит, что маршрут начинается в v и идет до u по единственному пути.

Так как друзья бизнесмена не очень впечатлительны, их нельзя удивить тыквами размера меньше, чем они видели. Проще говоря, до начала прогулки они не видели ни одной тыквы. Дальше, когда они идут по маршруту, каждый раз при виде тыквы большего размера, чем все ранее увиденные, друзья будут удивляться. Divanu интересно, сколько раз его друзья будут удивляться для каждого маршрута.

Так как Divan — самый богатый бизнесмен мира, он не утруждает себя логичностью построения маршрутов. Он просто выдумывает их и спрашивает у государства Берляндии: а если бы я повел друзей из v в u , то сколько раз они бы удивились? После чего ждет ответа, и все повторяется заново, пока Divan не устанет. А это может быть ой как не скоро, ведь он готов задать q вопросов.

Помогите правительству побороть кризис, ответив на все вопросы Divana.

Формат входных данных

На первой строке входных данных находится два натуральных числа n и q — количество тыкв в парке и количество вопросов ($1 \leq n \leq 200\,000$, $1 \leq q \leq 200\,000$).

На второй строке входных данных находятся n целых чисел W_1, W_2, \dots, W_n — размер тыкв ($1 \leq W_i \leq n$).

Далее следует $n - 1$ строка, в каждой из которых записано два числа a и b ($1 \leq a, b \leq n$, при этом $a \neq b$) — номера тыкв, которые соединяет очередная дорога.

Гарантируется, что от любой тыквы к любой другой можно пройти, и притом единственным способом.

Далее следует q строк, каждая из которых содержит два числа x_i и y_i ($0 \leq x_i, y_i \leq n$) — зашифрованный запрос.

Для того чтобы узнать настоящий запрос, нужно положить $v_i = (x_i + p) \bmod n + 1$, $u_i = (y_i + p) \bmod n + 1$, где v_i — начало пути для очередного запроса, u_i — конец пути, p — ответ на предыдущий запрос, а \bmod — операция взятия остатка от деления. Для первого запроса $p = 0$.

Формат выходных данных

Для каждого запроса выведите одно целое число — сколько раз удивятся друзья Divana, если их проведет по такому маршруту.

Система оценивания

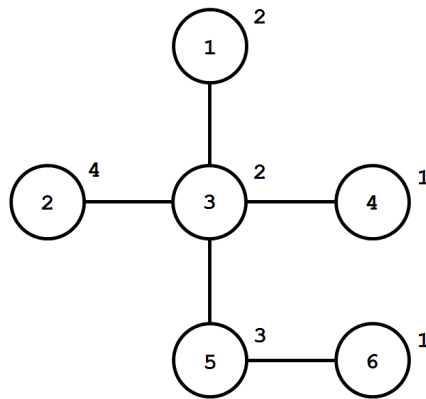
Подзадача 1, 40 баллов: $n, q \leq 5000$.

Подзадача 2, 60 баллов: дополнительных ограничений нет.

Пример

Входной файл	Выходной файл
6 3	3
2 4 2 1 3 1	2
1 3	1
2 3	
3 4	
5 3	
6 5	
5 1	
0 3	
5 1	

Примечание: Парк тыкв в примере можно схематично изобразить следующим образом:



Здесь окружности представляют собой тыквы, внутри окружности подписан номер соответствующей тыквы, а снаружи — её размер.

В примере $n = 6$ и заданы три запроса. Первый зашифрованный запрос — $x_1 = 5$, $y_1 = 1$. Поскольку $p = 0$, получаем настоящий запрос $v_1 = (5 + 0) + 1 = 6$, $u_1 = (1 + 0) + 1 = 2$. Маршрут между этими тыквами проходит через тыквы — 6, 5, 3, 2, соответственно, их размеры равны 1, 3, 2, 4. На этом маршруте друзья будут удивляться 3 раза — при виде тыкв с размерами 1, 3, 4.

Второй зашифрованный запрос — $x_2 = 0$, $y_2 = 3$. Поскольку ответ на предыдущий запрос — 3, то получаем $v_2 = (3 + 0) + 1 = 4$, $u_2 = (3 + 3) \bmod 6 + 1 = 1$. Маршрут проходит через тыквы 4, 3, 1 с размерами 1, 2, 2. Друзья удивятся при виде первых двух тыкв, поэтому ответ — 2. Обратите внимание, что при виде второй тыквы размера 2 друзья **не** удивляются.

В третьем запросе $x_3 = 5$, $y_3 = 1$, $v_3 = (5 + 2) \bmod 6 + 1 = 2$, $u_3 = (1 + 2) + 1 = 4$. Маршрут проходит через тыквы 2, 3, 4 размерами 4, 2, 1. Друзья удивляются только при виде первой тыквы маршрута. Обратите внимание, что зашифрованный запрос совпадает с первым: $x_3 = x_1 = 5$, $y_3 = y_1 = 1$. Однако, поскольку ответ на предыдущий запрос (p) изменился, то реальные значения v_3 , u_3 отличаются от первого запроса, поэтому ответ на него тоже другой.