

Музичне шоу (Fixshow)

Оскільки Дім Діміч у черговий раз забув привітати бабусю з днем народження, він вирішив влаштувати для неї грандіозне музичне шоу. Для цього він знайшов рівний майданчик, у певних координатах якого розставив N найсучасніших електронних синтезаторів звуку. За задумом Дім Діміча, у деякі (можливо різні) моменти часу, вони почнуть видавати звуки (гарантовано різні), і звучання буде тривалим. Оскільки Дім Діміч хоче, щоб бабуся почула усі звуки якомога швидше, він вирішив знайти таку точку на майданчику, в якій вперше будуть чути всі N звуків одночасно. Вирішити таку складну задачу самостійно він не зміг, а тому вдався до вашої допомоги.

Напишіть програму, яка обчислює координати точки і час, за який у ній звуки від усіх синтезаторів будуть чути вперше, за умови, що звук поширюється тільки по повітрю зі швидкістю C , однаковою для усіх синтезаторів і напрямків. Іншими затримками (наприклад, кінцевою швидкістю поширення електричних сигналів) можна знехтувати.

Формат введення-виведення:

Програма **fixshow** зчитує з клавіатури (стандартного пристрою введення) спочатку число N ($1 \leq N \leq 555$) – кількість синтезаторів, потім N трійок чисел – координати синтезаторів на майданчику та час початку їх звучання i , нарешті, число C ($0.1 \leq C \leq 1000$) – швидкість розповсюдження звуку.

Значення координат не перевищують по модулю 10^6 і не обов'язково цілі.

Програма **fixshow** виводить на екран (стандартний пристрій виведення) три числа через пробіл: координати точки, в якій вперше будуть чути всі звуки одночасно, і час, коли це відбудеться.

Результат зараховується, якщо час відрізняється від правильного не більше ніж на 10^{-3} та у вказаний час у вказаній точці будуть чути всі синтезатори.

Приклад вхідних та вихідних даних

Введення Виведення

4

2 8 6

16 4 0 5 4 11

5 1 8

3 6 2

1

Професор і помічник (helper)

Незбаром у фіксіків очікуються престижні змагання з ремонту складного обладнання. Як відомо, для усунення особливо складних поломок обладнання фіксікі використовують помічник, однак їм треба вміти користуватися досконало!

Професор Геній Євгенович винайшов нову експериментальну модель помічника. І всі хочуть з нею терміново ознайомитися. Але ... у професора відпустка! Сімка почала вмовляти його дати помічник у чиня і після довгих умовлянь професор погодився, але з деякими умовами:

- Геній Євгенович віддасть пристрій фіксіку X і хоче забрати його після відпустки у фіксіка Y.
- Кожен день помігатор повинен міняти свого власника.

Всього в класі навчається M учнів, а відпустка професора триває N днів.

Дім Дімічу відразу стало цікаво, які можуть бути варіанти «подорожі» помігатора по учнях, а Сімка – скільки всього існує таких варіантів.

Формат введення виведення:

Програма **helper** зчитує з клавіатура (стандартного пристрою введення) чотири цілих числа N, M, X, Y ($1 \leq N, M \leq 10^9, X, Y (1 \leq X, Y \leq M)$).

Програма **helper** виводить на екран (стандартний пристрій виведення) єдине число – кількість можливих варіантів «подорожі» помігатора по учнях, за заданими правилами. Оскільки число може бути дуже великим, виведіть залишок від ділення цього числа на $10^9 + 7$.

Приклад вхідних та вихідних даних

Введення Виведення

```

5
4 3 1 2
3 5 2 2 12
4 5 1 5 51

```

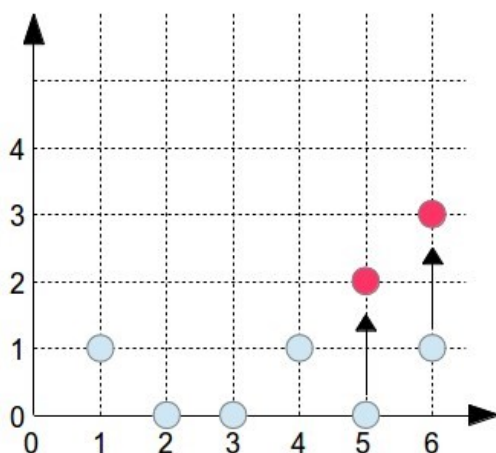
Примітка. У першому прикладі існують такі варіанти «подорожі» помігатора по учнях:

```

1 2 1 3 2
1 2 3 1 2
1 3 1 3 2
1 3 2 3 2
1 3 2 1 2

```

Копіє (copier)



Фіксіки настільки цікаві істоти, що неприємності Нулика під час його знайомства з копієм нічому їх не навчили. Тепер вже цілих N фіксіків забралися під кришку пристрою й опинилися у різних місцях скляної панелі ксерокса. Таємниця їх існування опинилася на межі розкриття, бо Лізонька, як завжди не вчасно, вирішила зробити копію якогось оголошення про загублені речі професора Генія Євгеновича. Щоб ніхто не зацікавився істотами на отриманій копії, потрібно одержати зображення, на якому рівно k фіксіків (точок) розташовано по діагоналі від низу до верху. Перелякані

фіксикі зовсім розгубилися і недовго було початися паніці, якби Сімка не запропонувала такий вихід.

Представимо скануючу поверхню ксерокса координатною площиною з початком координат у лівому нижньому куту панелі. Тоді розташування фіксиків можна задати координатами точок (i, y_i) , $i \in [1..N]$, $0 \leq y_i \leq N \leq 200000$.

По команді `fixir#i` фіксик з номером i може переміститися на одну клітинку вгору, тобто з точки з координатами (i, y_i) у точку з координатами (i, y_i+1) . Тепер потрібно за мінімальну кількість команд досягти того, щоб k фіксиків опинилися в точках з координатами $(x_i, 1)$, $(x_i+1, 2)$, ..., (x_i+k-1, k) (див. рисунок) або переконатися, що це неможливо.

Формат введення-виведення:

Програма **copier** спочатку зчитує з клавіатури (стандартного пристрою введення) два цілих числа N та k ($1 \leq k \leq N \leq 200000$). Потім зчитується N цілих чисел y_1, y_2, \dots, y_N – початкові положення фіксиків.

Програма **copier** виводить на екран (стандартний пристрій виведення) єдине число – мінімальну кількість команд, що потрібна для отримання необхідної конфігурації фіксиків, або «-1» (без лапок), якщо це зробити неможливо.

Приклад вхідних та вихідних даних

Введення Виведення

6 3

4

1 0 0 1 0 1

3 3

-1

1 3 2

Фііксикі рахують (*fixcount*)

І знову Дім Діміч з Ваською готуються до олімпіади з усного рахунку. Тільки тепер їм необхідно розв'язувати приклади не тільки на додавання, але й на множення. Довелося залучити не тільки всіх фіксиків, а й професора Генія Євгенійовича для складання прикладів. Це було настільки обтяжливе заняття, що хлопцям почали давати одну й ту саму послідовність чисел, яку треба було спочатку додати, а потім перемножити. Виявилось, що у деяких випадках результат отримувався абсолютно однаковий! Наприклад, для послідовності чисел 1, 2 і 3 виконується властивість $1 + 2 + 3 = 1 * 2 * 3 = 6$. Щоб трохи відволікти хлопців від рутинної роботи, Сімка запропонувала розв'язати обернену задачу: знайти число, яке можна представлено у вигляді набору натуральних чисел, сума і добуток яких дає одне й те саме число. І таке число знайшлося швидко - 4, адже $4 = 2 + 2 = 2 * 2$. Через деякий час загальними зусиллями було знайдено число, у якого було РІВНО ДВА різних набори натуральних чисел $\{a_1, a_2, \dots, a_p\}$ і $\{b_1, b_2, \dots, b_q\}$ ($2 \leq p, 2 \leq q$) таких, що

$$x = a_1 + a_2 + \dots + a_p = b_1 + b_2 + \dots + b_q = a_1 \cdot a_2 \cdot \dots \cdot a_p = b_1 \cdot b_2 \cdot \dots \cdot b_q$$

Та професор Геній Євгенович не зупинився на досягнутому! Він остаточно ускладнив задачу тим, що запропонував порахувати кількість цілих чисел на заданому інтервалі, для яких виконується вказана властивість, причому набори, що відрізняються тільки порядком елементів, потрібно вважати однаковими. Наприклад, $1+2+3$ і $2+3+1$ – однакові набори.

Формат введення-виведення:

Програма *fixcount* зчитує з клавіатури (стандартного пристрою введення) два цілих числа N та M ($1 \leq N \leq M \leq 10^{18}$).

Програма *fixcount* виводить на екран (стандартний пристрій виведення) єдине число – кількість цілих чисел $x \in [N, M]$, що мають вказану властивість.

Приклад вхідних та вихідних даних

Введення Виведення

10 30 1

Математика – цариця наук (*fixolymp*)

Перемога на олімпіаді з усного рахунку була здобута! Дім Діміч сидів і з подивом дивився на першу отриману медаль. В інтернеті він знайшов інформацію про те, що рівно через рік відбудуться дуже престижні змагання у місті Вінниці – XXV Всеукраїнський Турнір Чемпіонів. Треба готуватися. Але як?

І тут до нього прийшов на допомогу професор Геній Євгенович. Він пообіцяв регулярно давати Діму Дімічу завдання та постійно контролювати їх виконання. І ось перше завдання.

Дано два натуральних числа A і B . Треба знайти два натуральних числа X і Y такі, що

$$\text{НОД}(A, B) = \text{НОД}(X, Y)$$

$$\text{НОК}(A, B) = \text{НОК}(X, Y)$$

$$X \leq Y$$

$$Y - X \rightarrow \min$$

Формат введення-виведення:

Програма *fixolymp* зчитує з клавіатури (стандартного пристрою введення) два цілих числа A та B ($1 \leq A, B \leq 10^9$).

Програма *fixolymp* виводить на екран (стандартний пристрій виведення) два цілих числа, що задовольняють умові задачі.

Приклад вхідних та вихідних даних

Введення Виведення

1 12 3 4

4 3 3 4

Музыкальное шоу (Fixshow)

Поскольку Дим Димыч вновь забыл поздравить бабушку с днем рождения, он решил устроить для нее грандиозное музыкальное шоу. Для этого он нашел плоскую площадку, в определенных координатах которой расставил N новейших электронных синтезаторов звука. По замыслу Дим Димыча, в некоторые (возможно различные) моменты времени, они должны начать издавать звуки (гарантированно различные), и звучание должно быть длительным. Поскольку Дим Димыч хочет, чтобы бабушка услышала все звуки как можно скорее, он решил найти такую точку на площадке, в которой впервые будут слышны все N звуков одновременно. Решить такую сложную задачу самостоятельно он не смог, а поэтому прибег к вашей помощи.

Напишите программу, которая вычисляет координаты точки и время, в которое звуки от всех синтезаторов будут в ней впервые слышимы, при условии, что звук распространяется только по воздуху со скоростью C , одинаковой для всех синтезаторов и направлений. Остальными задержками (например, конечной скоростью распространения электрических сигналов) можно пренебречь.

Формат ввода-вывода:

Программа **fixshow** читает с клавиатуры (стандартного устройства ввода) сначала число N ($1 \leq N \leq 555$) – количество синтезаторов, затем N троек чисел – координаты очередного синтезатора на площадке и время начала его звучания и, наконец, число C ($0.1 \leq C \leq 1000$) – скорость распространения звука.

Значения координат не превышают по модулю 10^6 и не обязательно целые.

Программа **fixshow** выводит на экран (стандартное устройства вывода) три числа через пробел: координаты точки, в которой впервые будут слышимы все звуки одновременно, и время, когда это произойдет.

Результат засчитывается, если время отличается от правильного не более чем на 10^{-3} и в указанное время в указанной точке будут слышны все синтезаторы.

Пример входных и выходных данных:

Ввод Вывод

4

2 8 6

16 4 0 5 4 11

5 1 8

3 6 2

1

Профессор и помощник (helper)

У фиксиков в скором времени ожидаются престижные соревнования по ремонту сложнейшего оборудования. Как известно, для устранения особо сложных поломок оборудования фиксик используют помощника, однако им надо уметь пользоваться в совершенстве!

Профессор Гений Евгеньевич изобрел новую экспериментальную модель помогатора и все хотят с ней срочно ознакомиться. Но... у профессора отпуск! Симка начала уговаривать его дать помогатор ученикам и после долгих уговоров профессор согласился, но с некоторыми условиями:

- Гений Евгеньевич отдаст устройство фиксикам X и хочет забрать его после отпуска у фиксика Y .
- Каждый день помогатор **должен** менять своего владельца.

Всего в классе M учеников и отпуск профессора заканчивается через N дней.

Дим Димычу сразу стало интересно, какие могут быть варианты «путешествия» помогатора по ученикам, а Симке – сколько всего существует таких вариантов.

Формат ввода-вывода:

Программа **helper** читает с клавиатуры (стандартного устройства ввода) четыре целых числа N , M ($1 \leq N, M \leq 10^9$), X , Y ($1 \leq X, Y \leq M$).

Программа **helper** выводит на экран (стандартное устройства вывода) одно число – количество возможных вариантов «путешествия» помогатора по ученикам, по заданным правилам.

Поскольку число может быть очень большим, выведите остаток от деления этого числа на $10^9 + 7$.

Пример входных и выходных данных:

Ввод **Вывод**

5

4 3 1 2

3 5 2 2 12

4 5 1 5 51

Примечание. В первом примере существуют такие варианты «путешествия» помогатора по ученикам:

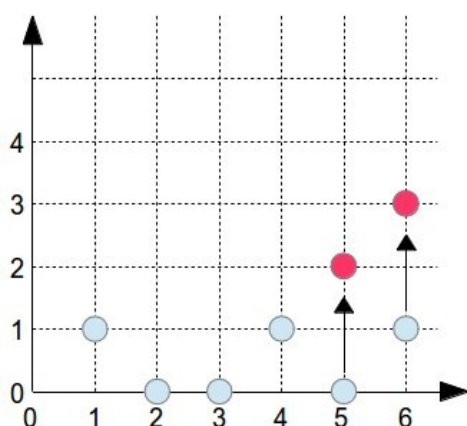
1 2 1 3 2

1 2 3 1 2

1 3 1 3 2

1 3 2 3 2

1 3 2 1 2



Копир (copier)

Фиксики настолько любопытные существа, что неприятности Нолика во время его знакомства с копиром ничему их не научили. Теперь уже целых N фиксиков забрались под крышку устройства и оказались в разных местах стеклянной панели ксерокса. Тайна их существования оказалась на грани раскрытия, ибо

Лизонька, как всегда не вовремя, решила сделать копию какого-то объявления о пропаже очередных вещей профессора Геня Евгеньевича. Чтобы никто не заинтересовался существами на полученной копии, надо получить изображение, на котором ровно k фиксиков (точек) расположено по диагонали снизу вверх. Перепуганные фиксиксы совсем растерялись и недолго было начаться панике, если бы Симка не предложила такой выход

Представим сканируемую поверхность ксерокса координатной плоскостью с началом координат в левом нижнем углу панели. Тогда расположение фиксиков можно задать координатами точек (i, y_i) , $i \in [1..N]$, $0 \leq y_i \leq N \leq 200000$.

По команде `fixup#i` фиксик с номером i может переместиться на одну клеточку вверх, т.е. из точки с координатами (i, y_i) в точку с координатами (i, y_i+1) . Теперь надо добиться, чтобы за минимальное количество команд k фиксиков оказались в точках с координатами $(x_i, 1)$, $(x_i+1, 2)$, ..., (x_i+k-1, k) (см. рисунок) или определить, что это сделать невозможно.

Формат ввода-вывода:

Программа **copier** сначала читает с клавиатуры (стандартного устройства ввода) два целых числа N и k ($1 \leq k \leq N \leq 200000$). Затем считывается N целых чисел y_1, y_2, \dots, y_N – начальные положения фиксиков.

Программа **copier** выводит на экран (стандартное устройства вывода) одно число – минимальное количество команд, необходимое для получения нужной конфигурации фиксиков, или «-1» (без кавычек), если это сделать невозможно.

Пример входных и выходных данных:

Ввод	Вывод
6 3	
1 0 0 1 0 1	4
3 3	
1 3 2	-1

Фиксиксы считают (*fixcount*)

И опять Дим Димыч с Васькой готовятся к олимпиаде по устному счету. Только теперь им необходимо решать примеры не только на сложение, но и на умножение. Пришлось привлечь не только всех фиксиков, а и профессора Геня Евгеньевича для составления примеров. Это было настолько утомительное занятие, что ребятам начали давать одну последовательность чисел, которую надо было сначала сложить, а потом умножить. Оказалось, что в некоторых случаях результат получался совершенно одинаковый! Например, для последовательности чисел 1, 2 и 3 выполняется свойство $1+2+3=1*2*3=6$. Чтобы немного отвлечь ребят от рутинной работы, Симка предложила решить обратную задачу: найти число, которое может быть представлено в виде набора натуральных чисел, сумма и произведение которых давало бы одно и то же число. И такое число нашлось быстро – 4, ведь $4=2+2=2*2$. Через некоторое время всеобщими усилиями было найдено число, у которого было РОВНО ДВА различных набора натуральных чисел $\{a_1, a_2, \dots, a_p\}$ и $\{b_1, b_2, \dots, b_q\}$ ($2 \leq p, 2 \leq q$) таких, что

$$x = a_1 + a_2 + \dots + a_p = b_1 + b_2 + \dots + b_q = a_1 \cdot a_2 \cdot \dots \cdot a_p = b_1 \cdot b_2 \cdot \dots \cdot b_q$$

Но профессор Гений Евгеньевич не остановился на достигнутом! Он окончательно усложнил задачу, предложив посчитать количество целых чисел на заданном интервале, для которых выполняется указанное свойство, причем наборы, отличающиеся только порядком входящих элементов, надо считать одинаковыми. Например, 1+2+3 и 2+3+1 – одинаковые наборы.

Формат ввода-вывода:

Программа **fixcount** читает с клавиатуры (стандартного устройства ввода) два целых числа N и M ($1 \leq N \leq M \leq 10^{18}$).

Программа **fixcount** выводит на экран (стандартное устройства вывода) одно число – количество целых чисел $x \in [N, M]$, обладающих указанным свойством.

Пример входных и выходных данных:

Ввод **Вывод**

10 30 1

Математика – царица наук (**fixolymp**)

Победа на олимпиаде по устному счету была одержана! Дим Димыч сидел и с удивлением смотрел на первую полученную медаль. В интернете он нашел информацию о том, что ровно через год состоятся очень престижные соревнования в городе Виннице – XXV Всеукраинский Турнир Чемпионов. Надо готовиться. Но как?

И тут на помощь к нему пришел профессор Гений Евгеньевич. Он пообещал регулярно давать Дим Димычу задачи и постоянно контролировать их выполнение. И вот первое задание.

Дано два натуральных числа A и B . Надо найти два натуральных числа X и Y такие, что

$$\text{НОД}(A, B) = \text{НОД}(X, Y)$$

$$\text{НОК}(A, B) = \text{НОК}(X, Y)$$

$$X \leq Y$$

$$Y - X \rightarrow \min$$

Формат ввода-вывода:

Программа **fixolymp** читает с клавиатуры (стандартного устройства ввода) два целых числа A и B ($1 \leq A, B \leq 10^9$).

Программа **fixolymp** выводит на экран (стандартное устройства вывода) два целых числа, удовлетворяющих условию задачи.

Пример входных и выходных данных:

Ввод **Вывод**

1 12 3 4

4 3 3 4

