

Задача 1. Кирпичная стена

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Петя и Вася нашли заброшенную кирпичную стену, представляющую собой несколько вертикальных рядов кирпичей. Рядом со стеной разбросаны кирпичи, которые не использовали при строительстве, и все они имеют одинаковый размер.

Стена оказалась недостроенной, высота от места к месту различалась. Ребята захотели достроить стену так, чтобы она целиком была одной высоты. Они решили **одновременно** класть каждый по одному кирпичу в разные места, пока высота везде не станет одинаковой. То есть, за одно действие Петя и Вася могут увеличить высоту двух различных рядов на размер кирпича.

В первый день они работали долго и упорно, но так и не добились желаемого результата, поэтому они решили написать программу, которая посчитает, как быстро они смогут достроить стену. За помощью они, естественно, обратились к вам.

Формат входных данных

В первой строке входного файла содержится одно целое число N — количество вертикальных рядов в стене ($2 \leq N \leq 10^5$).

В следующей строке заданы N целых неотрицательных чисел, каждое из которых не превышает 10^9 — количество кирпичей в соответствующем ряду.

Формат выходных данных

Если за несколько действий Петя и Вася могут сделать высоту всех рядов одинаковой, то в выходной файл необходимо вывести наименьшее число требуемых действий.

Если высоты рядов сделать равными нельзя, то нужно вывести число -1 .

Гарантируется, что ребята всегда смогут найти необходимое им количество кирпичей.

Система оценки

Пусть H — максимальное число кирпичей в вертикальном ряду.

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

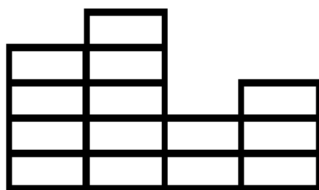
| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|----------------------------|-----------------------|
| 1 | 12 | $N \leq 3$ | |
| 2 | 23 | $N \cdot H \leq 10^6$ | |
| 3 | 30 | $N \leq 10^3$ | 1 |
| 4 | 35 | $N \leq 10^5, H \leq 10^9$ | 1, 2, 3 |

Примеры

| <code>input.txt</code> | <code>output.txt</code> |
|------------------------|-------------------------|
| 4 4 5 2 3 | 3 |
| 2 4 2 | -1 |

Замечание

В первом примере заброшенная стена изначально выглядит так:



Если доложить сначала в первый и третий ряды по кирпичу, а затем дважды в третий и четвертый, то получится стена, в которой в каждом ряду по 5 кирпичей.

Задача 2. Хитрый автомат

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

В современном кафе «А можно троечку?» недавно уволили всех официантов и поставили автомат по выдаче еды. Политика этого кафе заключается в том, что каждый посетитель может заказать только три блюда, ни больше ни меньше. Все блюда в кафе разбиты на категории: супы, десерты, напитки и т.п. Посетитель может выбрать только категорию каждого из трёх блюд, а сами блюда выбираются случайным образом. Отметим также, что нельзя заказать более одного блюда из какой-либо категории.

Новый автомат тоже подчиняется этой политике. На его стартовом экране демонстрируется список всех категорий блюд, в котором нужно отметить ровно три различных пункта. Далее на экране появляется число — количество различных комбинаций блюд, которые удовлетворяют заказу. Затем, если посетителя все устраивает, среди этих комбинаций случайным образом выбирается одна, которая и достаётся посетителю.

Школьник Ваня во время посещения кафе обратил внимание на одну интересную особенность: посетителю явным образом не сообщается, какое количество блюд каждой категории имеется в меню. Ваня задался целью добиться от автомата этой информации. Помогите ему!

Протокол взаимодействия

Это интерактивная задача, и в ней вам предстоит работать не с файловым вводом-выводом, а со специальной программой — интерактором. Взаимодействие с ней осуществляется через стандартные потоки ввода-вывода.

При старте вашей программе в стандартный поток ввода подаётся одно целое число N — количество категорий блюд, имеющихся в кафе ($4 \leq N \leq 10^5$).

Ваша программа должна отправлять запросы в стандартный поток вывода.

Запросы должны иметь следующий вид:

? $a b c$

Здесь a , b и c — различные числа ($1 \leq a, b, c \leq N$).

В ответ на запрос приходит одно целое число — количество различных комбинаций блюд, которые автомат может составить из блюд, относящихся к категориям, имеющим порядковые номера a , b и c в меню, и которые удовлетворяют политике кафе.

Если вы готовы назвать количество блюд каждой категории, то вместо очередного запроса выведите строку

! $k_1 k_2 \dots k_N$

Здесь k_i — количество блюд i -й категории. Гарантируется, что $0 \leq k_i \leq 100$.

После этого программа должна завершить работу.

Убедитесь, что вы выводите символ перевода строки и очищаете буфер потока вывода (команда `flush` языка) после каждого выведенного запроса. Иначе решение может получить вердикт `Timeout`.

В случае превышения лимита запросов решение получает вердикт `Wrong Answer`.

Система оценки

Обозначим за Z количество элементов k_i , которые равны 0, за Q — максимально допустимое количество запросов без учета вывода ответа.

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|---|-----------------------|
| 1 | 42 | $Z = 0, Q = N, N \geq 4$ | |
| 2 | 23 | $Z \leq 3, Q = N + 50, N \geq 7$ | 1 |
| 3 | 35 | $Z \leq \frac{2}{3}N, Q = N + 200, N \geq 50$ | 1, 2 |

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4 | ? 1 2 3 |
| 24 | ? 1 2 4 |
| 12 | ? 1 3 4 |
| 24 | ? 2 3 4 |
| 16 | ! 3 2 4 2 |
| 5 | ? 1 2 3 |
| 6 | ? 1 2 4 |
| 8 | ? 1 2 5 |
| 10 | ? 1 3 4 |
| 12 | ? 2 3 4 |
| 24 | ! 1 2 3 4 5 |

Задача 3. Иннокентий и путешествие

| | |
|-------------------------|--------------|
| Имя входного файла: | input.txt |
| Имя выходного файла: | output.txt |
| Ограничение по времени: | 4 секунды |
| Ограничение по памяти: | 256 мегабайт |

Перед тем как приехать в очередной город, Иннокентий отмечает на карте этого города интересные места, в которые он хочет попасть. Известно, что для любых двух мест на карте, с координатами (x_1, y_1) и (x_2, y_2) соответственно, расстояние между ними равно $|x_1 - x_2| + |y_1 - y_2|$.

Иннокентий хочет продумать маршрут для посещения интересных мест, но он недостаточно решителен для того, чтобы определиться с первой достопримечательностью в списке. А еще Иннокентий опытный путешественник и знает, что не стоит перенапрягаться в отпуске. Поэтому он хочет оценить, насколько далеко ему придется ходить в случае, если он начнет с определенного места и захочет сфотографировать некоторое количество достопримечательностей. Помогите ему и напишите программу, отвечающую на подобные запросы.

Формат входных данных

В первой строке входного файла заданы два числа N и Q – количество интересных мест и количество запросов ($1 \leq N \leq 10^5$, $1 \leq Q \leq 5 \cdot 10^4$).

В следующих N строках заданы пары целых чисел x, y – координаты очередного места ($|x|, |y| \leq 10^6$).

Затем в Q строках идут описания запросов, задающихся парой чисел u, k – номером стартовой достопримечательности в маршруте и количеством ближайших интересных мест для посещения ($1 \leq u \leq N, 1 \leq k \leq N - 1$).

Формат выходных данных

Для каждого запроса (u, k) в выходной файл необходимо вывести на отдельной строке одно число – наименьшее расстояние, при котором из места u можно попасть хотя бы в k других.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|--|-----------------------|
| 1 | 9 | $N \leq 100, Q \leq 100, x , y \leq 100$ | |
| 2 | 13 | $N \leq 2 \cdot 10^3, Q \leq 2 \cdot 10^3, x , y \leq 10^6$ | 1 |
| 3 | 35 | $N \leq 10^5, Q \leq 5 \cdot 10^4, x , y \leq 10^3$ | 1 |
| 4 | 43 | $N \leq 10^5, Q \leq 5 \cdot 10^4, x , y \leq 10^6$ | 1, 2, 3 |

Пример

| input.txt | output.txt |
|-----------|------------|
| 5 3 | 1 |
| 1 1 | 3 |
| 2 1 | 3 |
| 3 2 | |
| 2 3 | |
| 4 4 | |
| 2 1 | |
| 3 3 | |
| 4 4 | |

Замечание

На первом рисунке линией обозначена область, с расстоянием до положения Иннокентия не большим 1. Внутри этой области находится точка с номером 1, к которой Иннокентий может пойти.

На втором рисунке линией обозначена область, внутри которой находятся все места на расстоянии не более чем 3.

Пунктирной линией обозначена область, внутри которой находятся все точки на расстоянии не более чем 2. Заметьте, что эта область не подходит, так как в ней меньше 3 необходимых мест.

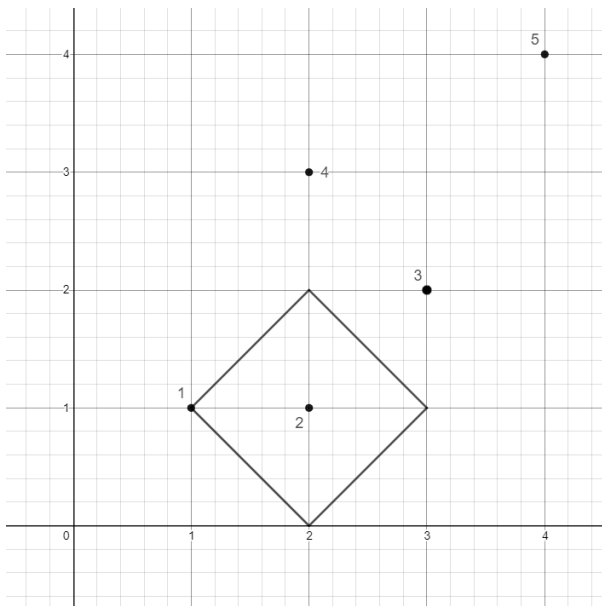


Рис. 1

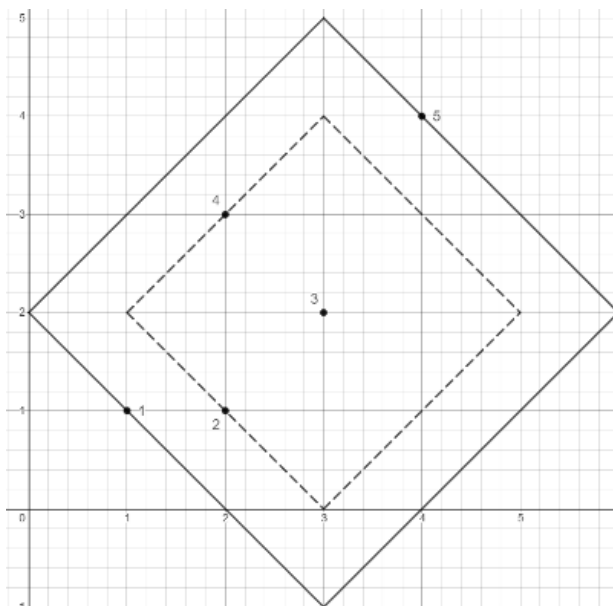


Рис. 2

Задача 4. Шпион в Берляндии

| | |
|-------------------------|--------------|
| Имя входного файла: | input.txt |
| Имя выходного файла: | output.txt |
| Ограничение по времени: | 1 секунда |
| Ограничение по памяти: | 256 мегабайт |

Шпион из соседней страны проник в Берляндию с целью изучить их железнодорожную систему. Берляндия — страна, в которой n городов, соединенных между собой железнодорожными путями так, что из любого города можно доехать до любого другого по железной дороге единственным способом. Назовем участком дороги железнодорожный перегон, соединяющий два города напрямую, не проходящий через другие города.

Шпиону необходимо собрать как можно более полную информацию о дорогах Берляндии. Все наблюдения он записывает в свой блокнот. Но он не хочет привлекать к себе внимание полицейских постов, которые есть повсюду. Поэтому он может на каждом участке дороги делать только одну запись. Кроме того, у него есть информация, что полицейские могут заподозрить что-то неладное, если он проедет по i -му участку железной дороги больше c_i раз.

Его путешествие по железной дороге начинается в городе s , а приехать ему нужно в город t , где его ждет связист, которому следует передать записи.

Посчитайте, какое наибольшее число записей шпион может собрать, и определите маршрут, с помощью которого он может этого достигнуть.

Формат входных данных

В первой строке входного файла записано число n — количество городов в Берляндии ($2 \leq n \leq 10^5$). Города пронумерованы числами от 1 до n .

В следующих $n - 1$ строках даны описания участков дороги. Очередной i -й участок дороги задаётся тремя целыми числами, разделёнными пробелами: a_i , b_i и c_i . Первые два числа a_i и b_i — это номера городов, которые соединяет i -й участок дороги ($1 \leq a_i, b_i \leq n$). Число c_i — наибольшее допустимое количество поездок по i -му участку дороги, которое не вызовет подозрений ($1 \leq c_i \leq 10^6$, $\sum_{i=1}^{n-1} c_i \leq 10^6$).

Наконец, в последней строке дано два целых числа s и t — номера начального города и города, в котором связист ожидает шпиона ($1 \leq s, t \leq n$).

Гарантируется, что между любыми двумя городами существует только один путь, не проходящий по одному и тому же участку дороги дважды.

Формат выходных данных

В первую строку выходного файла необходимо вывести целое число — наибольшее количество заметок, которое шпион сможет записать.

Во вторую строку требуется вывести через пробел номера городов, в том порядке, в котором их следует посещать.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|--|-----------------------|
| 1 | 25 | в Берляндии ровно два города, в которых заканчивается ровно 1 путь. | |
| 2 | 25 | $c_i = 2$ для любого i от 1 до $n - 1$ | |
| 3 | 25 | $n \leq 1000, \sum_{i=1}^{n-1} c_i \leq 1000$ | |
| 4 | 25 | без дополнительных ограничений | 1, 2, 3 |

Примеры

| input.txt | output.txt |
|----------------------------|--------------|
| 3 1 2 3 2 3 1 2 3 | 3 2 1 2 3 |
| 3 1 2 3 2 3 1 1 1 | 2 1 2 1 |
| 3 1 2 3 2 3 1 3 3 | 0 3 |

Задача 5. Предсказать игру

| | |
|-------------------------|--------------|
| Имя входного файла: | input.txt |
| Имя выходного файла: | output.txt |
| Ограничение по времени: | 4 секунды |
| Ограничение по памяти: | 256 мегабайт |

Станислав любит играть в компьютерные игры. А ещё больше он любит их ломать! Сейчас он исследует новый вид чата, который основывается на предсказании генерируемых игрой случайных чисел.

Как известно, в программировании почти отсутствуют истинно случайные числа. Когда в программе нужно сгенерировать случайное число, программист вызывает функцию вроде `rand()`. Подобные функции по заданному извне значению *seed* генерируют строго определённую последовательность чисел — такие числа обычно называют псевдослучайными. Несмотря на то, что последовательность выглядит как случайная, можно легко предсказать все числа, если определить закон генерации последовательности и значение *seed*.

Станислав узнал, что в интересующей его игре используется один глобальный генератор случайных чисел. У этого генератора в любой момент времени есть *состояние* — некоторое целое число X . Изначально состояние равно заданному программистом значению *seed*. Когда программист просит сгенерировать очередное число, состояние X изменяется по формуле:

$$X := (AX + B) \pmod M,$$

после чего новое значение X возвращается программисту в качестве случайного числа.

В программе на языке C такой генератор пишется так (константа A должна быть 64-битного типа):

```
int X; //глобальная переменная: состояние
int myrand() { //функция: возвращает следующее число
    X = (A * X + B) % M; //изменяем состояние по формуле
    return X; //возвращаем новое значение состояния
}
```

Станислав знает константы A , B и M в формуле, но *seed* в игре изменяется от запуска к запуску. Чтобы анализировать происходящее в игре, ему нужно научиться отвечать на вопросы двух видов:

1. Если генератор только что вернул число S , то какое число он вернёт через k вызовов?
2. Если генератор только что вернул число S , может ли он в будущем сгенерировать число T , и если может, то как скоро оно встретится впервые?

Формат входных данных

В первой строке входного файла задано три целых числа M , A и B — константы генератора псевдослучайных чисел ($2 \leq M \leq 10^9$, $0 \leq A, B \leq M - 1$, $A \neq 0$). Во второй строке записано целое число Q — количество вопросов ($1 \leq Q \leq 5 \cdot 10^4$).

В остальных Q строках записаны вопросы, по одному вопросу в строке. Формат вопроса зависит от типа:

- 1 $S k$ — нужно узнать, какое псевдослучайное число выдаст генератор через k вызовов.
- 2 $S T$ — нужно узнать, через сколько вызовов генератор впервые выдаст число T .

В обоих случаях второе число S — это последнее число, которое выдал генератор. Все числа целые, ограничения: $0 \leq S \neq T \leq M - 1$, $1 \leq k \leq 10^9$.

Формат выходных данных

В выходной файл требуется вывести Q целых чисел, по одному в строке — ответы на вопросы в порядке их задания. Для вопроса первого типа нужно вывести результат k -го по счёту вызова в будущем. Для вопроса второго типа нужно вывести, какой по счёту будущий вызов впервые вернёт число T . Если это не случится никогда, нужно вывести -1 вместо номера.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Исключение: подзадача 3 — по ней баллы начисляются отдельно за каждый пройденный тест, но лишь при условии прохождения всех необходимых подзадач.

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|---|-----------------------|
| 1 | 13 | $M \leq 10^4$, $Q \leq 100$, $k \leq 10^4$, M — простое | |
| 2 | 26 | $M \leq 10^9$, $Q \leq 100$, $k \leq 10^4$, M — простое | 1 |
| 3 | 24 | $M \leq 10^9$, $Q \leq 5 \cdot 10^4$, M — простое, нет вопросов первого типа | 1, 2 |
| 4 | 17 | $M \leq 10^9$, $Q \leq 5 \cdot 10^4$, нет вопросов второго типа | 1 |
| 5 | 20 | $M \leq 10^9$, $Q \leq 5 \cdot 10^4$ | 1, 2, 3, 4 |

Примеры

| input.txt | output.txt |
|--|---|
| 11 2 5 9 1 1 1 1 1 2 1 1 3 1 1 4 1 1 5 1 8 3 2 1 0 2 7 3 2 1 6 | 7 8 10 3 0 0 5 3 -1 |
| 16777216 16598013 12820163 5 1 1000000 1 1 1000000 2 1 1000000 4 2 1234567 7654321 1 1 1000000 | 7264259 13558458 16586500 14224986 489793 |