

## Задача А. Вася и коллекция изысканных массивов

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

Вася знаменит своей коллекцией изысканных массивов. Сегодня Вася хочет пополнить свою коллекцию массивом особого типа:

- количество элементов в массиве должно быть равно  $n$ ;
- сумма элементов в массиве должна равняться  $a$ ;
- разность между первым и вторым максимумами среди элементов должна равняться  $b$ ;
- все элементы в массивах должны быть целыми положительными числами.

Для заданных  $a$ ,  $b$  и  $n$  помогите Васе найти любой массив описанного типа или определите, что таких массивов не существует.

Первый максимум – максимальное значение в массиве. Второй максимум – максимум из оставшихся элементов в массиве после удаления первого максимума.

### Формат входных данных

В первой строке входного файла записаны три целых числа –  $a$ ,  $b$  и  $n$  ( $2 \leq a \leq 10^{18}$ ,  $0 \leq b \leq 10^{18}$ ,  $2 \leq n \leq 10^6$ ).

### Формат выходных данных

В выходной файл выведите  $n$  элементов массива через пробел, если такой массив существует. Если массивов, описанного типа не существует, выведите число  $-1$ .

### Система оценки

Решения, правильно работающие для  $n = 2$ , будут оцениваться в 10 баллов.

Решения, правильно работающие для  $n \leq 3$ , будут оцениваться в 20 баллов.

Решения, правильно работающие для  $n, a, b \leq 10^6$ , будут оцениваться в 50 баллов.

### Примеры

input.txt	output.txt
2 0 2	1 1
2 1 2	-1
7 4 3	1 1 5

## Задача В. Пробирки

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Неизвестный учёный Валентин имеет в своём распоряжении несколько пробирок, в каждую из которых насыпана смесь соли, сахара и соды. Смесь в  $i$ -й пробирке такова, что в каждом килограмме такой смеси находится  $a_i$  микрограммов соли,  $b_i$  микрограммов сахара. Всё остальное, конечно же, сода.

Сегодня к Валентину будут приходить его известные коллеги. Каждому посетителю нужно от Валентина лишь одно: смесь соли, сахара и соды в определённых пропорциях. Точнее,  $j$ -й посетитель хочет иметь смесь, содержащую  $x_j$  микрограммов соли и  $y_j$  микрограммов сахара на килограмм такой смеси. Чтобы угодить посетителю, Валентин сделает следующее:

1. выберет не больше трёх пробирок из имеющихся,
2. возьмёт некоторое количество смеси из каждой выбранной пробирки и поместит его в одну чашу,
3. смешает содержимое чаши и выдаст посетителю.

Конечно, Валентин постарается сделать так, чтобы в чаше была смесь, которую посетитель запросил. Ваша задача — написать программу, которая поможет Валентину выбрать не больше трёх пробирок для очередного посетителя или сообщить о том, что удовлетворить посетителя невозможно. Предполагается, что в каждой пробирке имеется достаточное количество смеси, которое может потребоваться Валентину.

### Формат входных данных

В первой строке входного файла записано целое число  $N$  — количество пробирок ( $1 \leq N \leq 10^5$ ). Далее следует  $N$  строк, в  $i$ -й из которых даны целые числа  $a_i$  и  $b_i$  ( $0 \leq a_i, b_i \leq 10^9$ ,  $a_i + b_i \leq 10^9$ ).

В следующей строке записано целое число  $M$  — количество посетителей ( $1 \leq M \leq 10^5$ ). Затем следует  $M$  строк, в  $j$ -й из которых даны целые числа  $x_j$  и  $y_j$  ( $0 \leq x_j, y_j \leq 10^9$ ,  $x_j + y_j \leq 10^9$ ,  $1 \leq j \leq M$ ).

### Формат выходных данных

В выходной файл необходимо вывести  $M$  строк,  $j$ -я из которых должна содержать ответ на задачу для  $j$ -го посетителя. Если для  $j$ -го посетителя получить желаемую смесь не удастся, то в ответ следует вывести единственное число  $-1$ . Если получить желаемую смесь можно, то ответ должен начинаться с числа выбранных пробирок  $k$  ( $1 \leq k \leq 3$ ), а затем, через пробел, должны быть выписаны номера этих пробирок.

### Система оценки

Решения, правильно работающие для ( $M, N \leq 80$ ), будут оцениваться в 20 баллов.

Решения, правильно работающие для ( $M, N \leq 250$ ), будут оцениваться в 40 баллов.

Решения, правильно работающие для ( $M \leq 100$ ), будут оцениваться в 60 баллов.

### Пример

input.txt	output.txt
1	1 1
1 1	-1
2	
1 1	
1 2	

## Задача С. Сплошные проблемы

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

Как всем известно, в клубе любителей проблем любят решать проблемы. У каждого участника клуба есть свой непустой набор проблем. За время существования клуба суммарно проблем накопилось столько, что пора бы уже начать их решать.

Участники клуба собрали все свои проблемы в одну кучу и теперь хотят понять, кто какие проблемы из этой кучи возьмёт для решения. Но тут есть одна небольшая проблема — чтобы никто не был в обиде, каждый должен получить одинаковое количество проблем для решения, а это, как вы уже догадались, возможно не получится обеспечить. Поэтому члены клуба решили пригласить в свой клуб новых участников. Слишком проблемных людей клуб приглашать не хочет, как следствие, каждому новому человеку разрешено приходить ровно с одной новой проблемой. Количество людей — не проблема, оно может быть любым, главное, чтобы общее число проблем из кучи можно было разделить между участниками поровну.

К сожалению, с описанной небольшой проблемой никто из клуба справиться не может, поэтому они попросили Вас помочь определить, сколько человек нужно пригласить для успешного распределения общей кучи проблем.

### Формат входных данных

В единственной строке входного файла записано два числа  $a$  и  $b$  — общее количество проблем в исходной куче и количество людей в клубе ( $1 \leq b \leq a \leq 10^{17}$ ).

### Формат выходных данных

В выходной файл необходимо вывести одно целое неотрицательное число — сколько ещё людей нужно пригласить, чтобы клубу удалось разделить проблемы между участниками поровну. Ответ не должен превосходить число  $10^{18}$ . Если подходящих ответов существует несколько, то выведите любой из них. Если же ответа с такими ограничениями не существует, то выведите число  $-1$ .

### Система оценки

Решения, которые находят ответы с ограничением до  $10^8$  набирают 30 баллов.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
11 5	1
12 10	-1

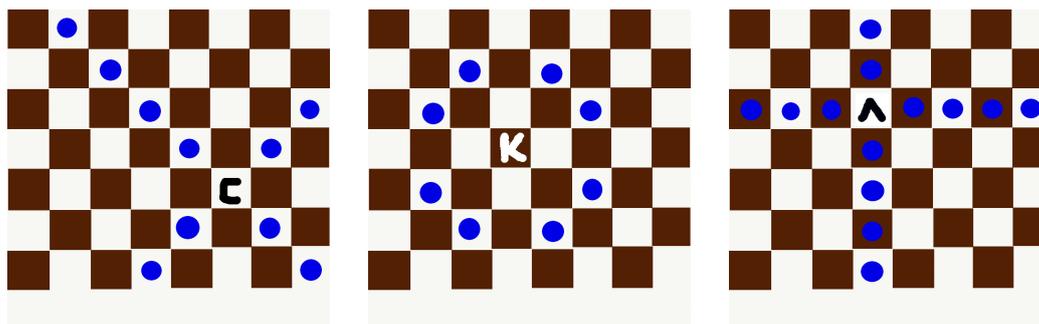
## Задача D. Бить или не бить?

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

На весенние каникулы, как обычно, Петя поехал к бабушке в Диманово. На второй день он открыл коробку со своими старыми «сокровищами». Среди прочих полезных вещей он нашёл шахматную доску. Однако почти все шахматные фигуры оказались потерянными, кроме одной *ладьи*, одного *коня* и одного *слона*. Вспоминая, как ходят эти фигуры, Петя начал расставлять их на разные клетки доски. В итоге ему удалось расставить их так, чтобы каждая фигура не била по шахматным правилам никакую другую из двух оставшихся. После этого он переключился на другие вещи из коробки.

Приехав домой после каникул, Петя вспомнил про ту расстановку фигур и решил написать программу, которая находит количество всевозможных таких расстановок на заданном прямоугольном шахматном поле. Помогите ему написать такую программу.

Напомним, что ладья бьёт любую из клеток доски, расположенных на ее горизонтали, либо вертикали; слон бьёт любую из клеток, которые находятся на одной из двух диагоналей, проходящих через его клетку; конь бьёт любую из клеток, в которые можно попасть, перейдя из своей клетки на одну клетку в одном направлении и на две клетки в другом (направления – горизонтальное и вертикальное). На рисунках ниже приведены примеры для каждой из фигур.



### Формат входных данных

В первой строке входного файла записаны два числа  $n$  и  $m$  – количество строк и столбцов у шахматного поля ( $n, m \geq 2, n \cdot m \leq 10^6$ ).

### Формат выходных данных

В выходной файл необходимо вывести одно число – количество расстановок одной ладьи, одного коня и одного слона на шахматном поле  $n \times m$  таких, что все фигуры стоят в разных клетках и каждая из фигур не находится под ударом никакой из двух других фигур.

### Система оценки

- Решение, правильно работающее для полей размера  $n \cdot m \leq 400$ , набирает 50 баллов
- Решение, правильно работающее для полей размера  $n \cdot m \leq 1000$ , набирает 60 баллов
- Решение, правильно работающее для полей размера  $n \cdot m \leq 4000$ , набирает 70 баллов
- Решение, правильно работающее для полей размера  $n \cdot m \leq 40000$ , набирает 80 баллов

### Примеры

<code>input.txt</code>	<code>output.txt</code>
2 2	0
3 2	4

## Задача Е. Розовая страна

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

В Розовой стране  $N$  городов, каждый из которых покрашен в голубой цвет. Конечно, такая покраска не соответствует названию страны, и поэтому города начали перекрашивать. К году номер 1 власти страны организовали перекраску столицы в розовый цвет. Каждый последующий год власти страны будут выбирать очередной город, покрашенный в голубой цвет, и перекрасят его в розовый. Ваша задача — выбрать порядок, в котором следует перекрашивать города так, чтобы минимизировать суммарное неудобство жителей страны.

Перекраска происходит очень быстро в начале года, поэтому для покраски  $N$  городов потребуется  $N - 1$  полных лет. Суммарное неудобство жителей — это сумма годовых неудобств по всем  $N - 1$  годам перекрашивания. Как посчитать годовое неудобство, будет рассказано далее.

В стране существует  $M$  транспортных связей. Каждая связь — это возможность уехать из города  $u$  и приехать в город  $v$ , или наоборот из  $v$  в  $u$ . Для любого из таких переездов нужно купить билет.

Жители страны любят менять своё цветное окружение, и поэтому будут ездить из одного города в другой. С наступлением года номер  $i$ , ( $1 \leq i \leq N - 1$ ) число розовых городов станет равным  $i$ , и тогда годовое неудобство за этот год можно будет выразить величиной, равной стоимости самого дорогого билета за переезд из какого-нибудь голубого города в какой-нибудь розовый.

### Формат входных данных

В первой строке даны целые числа  $N$  и  $M$  ( $1 \leq N, M, \leq 10^5$ ) — число городов и число транспортных связей.

Далее следует  $M$  строк, в  $i$ -й из которых дано описание очередной транспортной связи. Описание связи — это, разделённые пробелом, три целых числа  $u_i, v_i, e_i$  ( $1 \leq u_i, v_i \leq N, 1 \leq e_i \leq 10^9$ ), — номера городов и стоимость билета соответственно. Между двумя городами может быть несколько транспортных связей. Могут быть и транспортные связи ведущие из города в него же.

Гарантируется, что по транспортным связям можно добраться от столицы до любого другого города.

### Формат выходных данных

Выведите номера городов в порядке, в котором их следует перекрашивать. Не забудьте вывести и номер столицы, которая перекрашена первой. Номер столицы всегда равен 1.

### Примеры

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

## Задача F. Поиск коробок

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

Ваня – талантливый изобретатель. Однажды он создал робота, с помощью которого хотел автоматизировать работу почты в Диманово. По задумке Вани роботу загружается некоторое количество посылок, которые он пронумеровывает, начиная с номера один, а затем выгружает на ленту, замкнутую в круг. После этого робот используется для поиска посылок: он выдает номер посылки, стоящей на месте  $x$ , считая от начального положения. В случае, если  $x$  больше, чем количество посылок на ленте, робот промотает ленту на несколько кругов и определит номер посылки, стоящей на позиции  $(x \bmod N) + 1$ , где  $N$  – количество коробок на данный момент на ленте, а  $\bmod$  – взятие остатка от деления. После каждого запроса лента возвращается в исходное положение на посылку под номером один.

Алгоритм поиска посылки, написанный Ваней, работал замечательно до того момента, пока он не обнаружил, что его коллега Василий забрал на доставку несколько подряд идущих коробок. Теперь результат выполнения команд не соответствует ожиданию Вани, ведь номера посылок больше не совпадают с их реальной позицией на ленте. В своих попытках доработать алгоритм Ваня заметил, что Василий точно не забрал первую и последнюю коробки, однако решение всё еще далеко от завершения. Поэтому Ваня просит вас написать программу, которая поможет ему узнать, сколько посылок изначально загрузили роботу, сколько коробок забрал Василий и номер первой коробки, которую он забрал.

### Формат входных данных

В данной задаче вам предстоит работать не с файловым вводом-выводом, а со специальной программой – интерактором. Взаимодействие с ней осуществляется через стандартные потоки ввода-вывода.

Каждый запрос вашей программы должен состоять из одной строки следующего вида: ?  $x$ , где  $x$  – целое число, обозначающее позицию ( $1 \leq x \leq 10^{18}$ ). Ответом за запрос будет номер коробки, стоящей на позиции  $x$ .

Запрос, в котором вы сообщаете интерактору ответ на задачу, должен иметь вид !  $N$   $y$   $l$ , где  $N$  – количество коробок в начале,  $y$  – номер первой коробки, которую забрал Василий,  $l$  – количество коробок, которое забрал Василий. После данного запроса ваша программа должна завершиться и ничего больше не выводить в стандартный поток вывода ( $1 < y < N \leq 10^9$ ).

Учтите, что использование робота – занятие недешевое, поэтому Ваня хочет, чтобы вы узнали ответ **не более чем за 90 запросов**, иначе решение получит вердикт **Wrong Answer**.

Убедитесь, что вы выводите символ перевода строки и очищаете буфер потока вывода (команда `flush` языка) после каждой выведенной команды. Иначе решение может получить вердикт **Timeout**.

### Система оценки

Решения, правильно работающие для  $y \leq 90$ , будут оцениваться в 10 баллов.

Решения, правильно работающие для  $y \leq 180$ , будут оцениваться в 30 баллов.

Решения, правильно работающие для  $y \leq 900$ , будут оцениваться в 70 баллов.

### Пример

стандартный ввод	стандартный вывод
1	? 1
1	? 10
5	? 6
1	? 4
2	? 5
	! 5 3 2