

Задача С. Отношение чисел Фибоначчи (сложная версия)

Вход: `stdin`
 Выход: `stdout`
 Ограничение по времени: 2 с
 Ограничение по памяти: 512 Мб

У чисел Фибоначчи много замечательных свойств. В частности, отношение двух последовательных чисел Фибоначчи стремится к золотому сечению:

$$\lim_{n \rightarrow \infty} \frac{F_{n+1}}{F_n} = \frac{1 + \sqrt{5}}{2}.$$

Вам предстоит проверить это свойство для обобщенных чисел Фибоначчи

$$G_n = aG_{n-1} + bG_{n-2}, \quad n \geq 2.$$

По заданным a , b , G_0 и G_1 вычислите предел

$$\lim_{n \rightarrow \infty} \frac{G_{n+1}}{G_n},$$

в случае если он существует.

Формат входного файла

В первой строке заданы четыре целых числа a , b , G_0 , G_1 от -1000 до 1000 .

Формат выходного файла

В первой строке выведите “YES”, если предел существует, и “NO” в противном случае. В случае положительного ответа во второй строке выведите сам предел. Абсолютная или относительная погрешность ответа не должна превышать 10^{-6} .

Примеры

stdin	stdout
1 1 1 1	YES 1.6180339887
0 0 1 1	NO

Примечание

Если последовательность G_n содержит бесконечное количество нулей, считается, что предела не существует.

Задача D. Обобщенная последовательность Фибоначчи

Вход: `stdin`
Выход: `stdout`
Ограничение по времени: 2 с
Ограничение по памяти: 512 Мб

Назовем обобщенной последовательностью Фибоначчи последовательность, первые два элемента которой равны a и b (a и b — положительные целые числа), а каждый следующий элемент равен сумме двух предыдущих. Таким образом, каждая такая последовательность однозначно определяется её первыми двумя членами. Например, если $a = 3$, $b = 4$, то первые члены последовательности: 3, 4, 7, 11, 18, 29, ... Стандартная последовательность чисел Фибоначчи получается при $a = b = 1$.

Вам дано число n . Требуется найти количество обобщенных последовательностей Фибоначчи, в которых n встречается в качестве i -ого элемента, где $i > 2$. Элементы обобщенной последовательности нумеруются с единицы.

Формат входного файла

Входные данные содержат единственное число n ($1 \leq n \leq 10^6$).

Формат выходного файла

Выведите единственное число — ответ на задачу.

Примеры

<code>stdin</code>	<code>stdout</code>
3	3
1	0

Задача Е. Сложение в фибоначчиевой системе счисления

Вход: `stdin`
 Выход: `stdout`
 Ограничение по времени: 2 с
 Ограничение по памяти: 512 Мб

Числа Фибоначчи определяются рекуррентным соотношением:

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2}, \text{ для } n > 1.$$

Фибоначчиева система счисления (ФСС) основана на теореме Цекендорфа, утверждающей, что любое целое положительное число имеет единственное представление вида

$$n = F_{k_1} + F_{k_2} + \dots + F_{k_r},$$

где F_{k_i} — числа Фибоначчи, а $k_1 \gg k_2 \gg \dots \gg k_r \gg 0$. Здесь $i \gg j$ означает, что $i \geq j + 2$. Целое неотрицательное число можно записать в виде последовательности нулей и единиц:

$$n = (b_m b_{m-1} \dots b_2)_F \Leftrightarrow n = \sum_{k=2}^m b_k F_k,$$

где $b_i = 1$, если F_i входит в представление, и 0 в противном случае. Например, $1000000 = 832040 + 121393 + 46368 + 144 + 55 = F_{30} + F_{26} + F_{24} + F_{12} + F_{10}$, или $(1000000)_{10} = (100010100000000000010100000000)_F$. Эта система счисления напоминает двоичное представление, за исключением того, что в ней никогда не встречаются две единицы подряд.

При прибавлении единицы к числу в ФСС возникают два случая. Если младший разряд есть нуль, он заменяется на единицу (так как $F_2 = 1$), в противном случае в двух младших разрядах будет 01, и они заменяются на 10 (так как $F_3 = F_2 + 1$). Затем мы должны заменять набор цифр 011 на 100 (так как $F_n = F_{n-1} + F_{n-2}$), до тех пор, пока в строке цифр имеются две рядом стоящие единицы.

Напишите программу для сложения двух чисел в ФСС.

Формат входного файла

Вам даны два неотрицательных целых числа в ФСС, по одному числу в строке. Количество разрядов в числах может быть различным и не превосходит 5000.

Формат выходного файла

Выведите результат сложения этих двух чисел также в ФСС.

Пример

<code>stdin</code>	<code>stdout</code>
1010 100	10010

Задача F. Игра с монетами

Вход: `stdin`
Выход: `stdout`
Ограничение по времени: 2 с
Ограничение по памяти: 512 Мб

Петя и Вася придумали игру с монетами. У них есть n монет достоинствами a_i . В наборе могут быть монеты одинаковых достоинств. Сначала все монеты лежат на столе. Игроки по очереди берут по одной монете со стола. Первым ходит Петя. Он выигрывает, если после очередного хода у него окажется наперед заданная и известная обоим игрокам сумма s . Возможно, в этот момент на столе еще будут оставаться другие монеты. Вася выигрывает, если ему удастся помешать Пете набрать заданную сумму.

Петя считает игру нечестной, потому что сумм s , при которых он может выиграть, достаточно мало. Помогите ребятам определить все такие суммы s , которые Петя может набрать независимо от действий Васи.

Формат входного файла

В первой строке содержится целое число n ($1 \leq n \leq 100$) — количество монет на столе. В следующей строке заданы n целых положительных чисел — достоинства монет a_i . Их общая сумма не превосходит 100.

Формат выходного файла

В первой строке выведите количество сумм s , выигрышных для Пети. В следующей строке через пробел выведите сами суммы в порядке возрастания.

Примеры

<code>stdin</code>	<code>stdout</code>
3	3
2 2 3	2 3 5

Задача G. Цзяньшицзы с тремя кучками

Вход: `stdin`
Выход: `stdout`
Ограничение по времени: 2 с
Ограничение по памяти: 512 Мб

Имеются три кучки камней. Двое по очереди берут камни из этих кучек, причем при каждом ходе игрок может взять или произвольное количество камней из одной кучки, или поровну из всех трех. Выигрывает тот, кто заберет последний камень. Определите, кто выиграет при оптимальной игре.

Формат входного файла

Входные данные содержат несколько тестов. В первой строке содержится их количество t ($1 \leq t \leq 1000$). Следующие t строк описывают сами тесты. Каждая из них содержит тройку чисел a, b, c ($1 \leq a, b, c \leq 10^{12}$) — количества камней в кучках.

Формат выходного файла

Для каждого теста в отдельной строке выведите номер игрока (1 или 2), который выиграет при правильной игре. В случае если выиграет первый игрок, в той же строке через пробел выведите первый ход, который приведет его к победе. Ход описывается двумя целыми числами: номером кучки и строго положительным количеством камней, которые нужно из нее взять. Кучки нумеруются числами 1, 2, 3. Если предполагается взять поровну камней из всех трех кучек, в качестве номера кучки выведите 0. Если вариантов выигрышного хода несколько, выведите любой. Если выиграет второй игрок, ход выводить не нужно.

Примеры

<code>stdin</code>	<code>stdout</code>
3	1 0 1
1 1 1	2
1 2 3	1 1 4
5 2 3	

Задача Н. Спички детям не игрушки

Вход: `stdin`
Выход: `stdout`
Ограничение по времени: 2 с
Ограничение по памяти: 512 Мб

Петя и Вася снова играют в игру со спичками. У них имеется кучка спичек, из которой игроки по очереди берут спички по определенным правилам. Проигрывает тот, кто не может сделать ход.

Варианты игры, в которых за ход разрешается брать не более фиксированного количества спичек, извечным персонажам олимпиадных задач уже давно надоели. Поэтому сегодня они договорились, что количество спичек, которое игрок может брать за ход — от одной до $\lfloor \log_2 n \rfloor$ (округленного вниз двоичного логарифма числа спичек n *перед текущим ходом*).

Несмотря на то, что игра обещает быть интересной, у Пети с Васей нет возможности ее доиграть. С минуты на минуту придет бабушка, которая игры со спичками категорически не одобряет. Помогите ребятам определить, кто из них выиграет при оптимальной игре.

Формат входного файла

Входные данные содержат несколько тестов. В первой строке содержится их количество t ($1 \leq t \leq 1000$). Следующие t строк описывают сами тесты. Каждая из них содержит целое число n ($1 \leq n \leq 10^{18}$) — количество спичек в кучке в начале игры.

Формат выходного файла

Для каждого теста в отдельной строке выведите номер игрока (1 или 2), который выиграет при правильной игре. В случае если выиграет первый игрок, в той же строке через пробел выведите первый ход, который приведет его к победе — число спичек, которое нужно взять из кучки. Если вариантов выигрышного хода несколько, выведите любой. Если выиграет второй игрок, ход выводить не нужно.

Примеры

<code>stdin</code>	<code>stdout</code>
5	2
1	1 1
2	2
3	1 1
4	1 2
5	

Примечание

$\log_2 1 = 0$, поэтому если в кучке останется одна спичка, игра заканчивается.

Задача I. Ферзь в угол!

Вход: `stdin`
Выход: `stdout`
Ограничение по времени: 2 с
Ограничение по памяти: 512 Мб

В левом верхнем углу шахматной доски $n \times m$ стоит ферзь. Двое играют в следующую игру. Ходят по очереди. Каждым ходом разрешается передвинуть ферзя по вертикали вниз, по горизонтали вправо или по диагонали вправо-вниз на любое ненулевое число клеток. Проигрывает тот, после чьего хода ферзь окажется в правом нижнем углу доски.

Как вы уже догадались, ваша задача определить, кто выиграет при правильной игре.

Формат входного файла

Входные данные содержат несколько тестов. В первой строке содержится их количество t ($1 \leq t \leq 1000$). Следующие t строк описывают сами тесты. Каждая из них содержит пару чисел n и m ($2 \leq n, m \leq 10^{12}$), n — высота доски, m — ширина доски.

Формат выходного файла

Для каждого теста в отдельной строке выведите номер игрока (1 или 2), который выиграет при правильной игре. В случае если выиграет первый игрок, в той же строке через пробел выведите первый ход, который приведет его к победе. Ход описывается двумя целыми неотрицательными числами — смещением по вертикали a и смещением по горизонтали b . Заметим, что у корректного хода ферзя $a = 0$, или $b = 0$, или $a = b$, при этом a и b не могут быть нулями одновременно. Если вариантов выигрышного хода несколько, выведите любой. Если выиграет второй игрок, ход выводить не нужно.

Пример

<code>stdin</code>	<code>stdout</code>
3	2
3 3	1 1 1
5 7	1 9 0
10 2	

Задача J. (p, q)-коня в угол!

Вход: `stdin`
 Выход: `stdout`
 Ограничение по времени: 2 с
 Ограничение по памяти: 512 Мб

(p, q) -конь — это фигура, которая за один ход перемещается на p клеток в одном направлении (по вертикали или по горизонтали) и на q клеток в перпендикулярном ему. Для обычного шахматного коня $p = 1, q = 2$.

(p, q) -конь стоит в левом верхнем углу $(1, 1)$ шахматной доски $n \times m$ и ему нужно добраться до правого нижнего угла (n, m) . В этой задаче конь может совершать только такие ходы, которые приближают его к цели и в горизонтальном, и в вертикальном направлении. То есть он может пойти либо на p клеток вправо и на q клеток вниз, либо наоборот: на q клеток вправо и на p вниз. Разумеется, конь не может выходить за пределы доски. Кроме того, на доске есть запрещенные клетки, по которым также нельзя ходить (но через которые можно перепрыгивать). Клетки $(1, 1)$ и (n, m) не могут быть запрещенными.

Определите количество способов добраться из клетки $(1, 1)$ в клетку (n, m) , не посещая запрещенные клетки. Поскольку это количество может быть очень большим, выведите его по данному простому модулю d .

Формат входного файла

В первой строке содержатся параметры коня p и q и простой модуль d ($1 \leq p < q \leq 20, 2 \leq d \leq 10^6$). Во второй строке заданы размеры доски n и m ($2 \leq n, m \leq 10^9$). В следующей строке содержится одно число k — количество запрещенных клеток ($0 \leq k \leq 10$). В следующих k строках перечислены запрещенные клетки парами координат " $x_i y_i$ " ($1 \leq x_i \leq n, 1 \leq y_i \leq m$). Все запрещенные клетки различны между собой и не совпадают с начальной и конечной клетками $(1, 1)$ и (n, m) .

Формат выходного файла

Выведите одно целое число — ответ на задачу по модулю d .

Примеры

stdin	stdout
1 2 23 4 4 0	2
1 2 23 4 4 1 3 2	1

Задача К. Игра

Вход: `stdin`
 Выход: `stdout`
 Ограничение по времени: 2 с
 Ограничение по памяти: 512 Мб

В аристократических кругах Берляндии получила распространение следующая игра. Поле для этой игры представляет собой l ячеек, расположенных по кругу. В некоторых ячейках расположены фишки белого и черного цвета, причем в одной ячейке не может быть более одной фишки. Фишками белого цвета играет первый игрок, фишками черного цвета — второй. Игроки делают ходы по очереди. Первым ходит игрок, фишки которого белые. За один ход игрок может взять любую фишку своего цвета и подвинуть ее на ненулевое количество клеток по или против часовой стрелки. При этом при перемещении не разрешается перепрыгивать ни через какие другие фишки или занимать ячейку, в которой уже находится какая-то фишка. Проигрывает в игре тот игрок, кто не может сделать очередной ход. Учитывая, что оба противника играют оптимально, определите исход игры. Игра может завершиться ничьей (т.е. может продолжаться бесконечно).

Формат входного файла

Входной файл состоит из нескольких тестовых примеров. В первой строке записано целое число t , обозначающее количество этих примеров. Описание каждого теста начинается с тройки целых чисел l , n и m ($1 \leq l \leq 10^9, 1 \leq n, m \leq 10^6$) — количество ячеек, количество белых фишек и количество черных фишек соответственно. Далее следует возрастающая последовательность из n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq l$) — номера ячеек, в которых стоят белые фишки. Затем следует возрастающая последовательность из m целых чисел $b_1, b_2, \dots, b_m$ ($1 \leq b_i \leq l$) — номера ячеек, в которых стоят черные фишки. Ячейки пронумерованы от 1 до l последовательно по часовой стрелке. Все числа a_i и b_j различны. Суммы значений n и m по всем тестовым примерам не превосходят 10^6 .

Формат выходного файла

Для каждого тестового примера выведите ответ на отдельной строке. Если победит игрок,двигающий белые фишки, выведите “white”. В случае, если победит игрок,двигающий черные фишки, выведите “black”. Если же игра завершится ничьей, выведите “draw”.

Примеры

stdin	stdout
4	black
2 1 1	draw
1	white
2	white
4 1 1	
1	
2	
4 2 1	
1 4	
2	
6 2 2	
5 6	
2 4	

Задача L. Игра со строкой

Вход: `stdin`
 Выход: `stdout`
 Ограничение по времени: 2 с
 Ограничение по памяти: 512 Мб

Это интерактивная задача. Ваша программа должна читать данные из стандартного входного потока и выводить данные в стандартный выходной поток. После вывода каждой строки используйте flush-операции, например, `fflush(stdout)` в C++, `System.out.flush()` в Java, `flush(output)` в Pascal, `sys.stdout.flush()` в Python, `Console.Out.Flush()` в C#.

Представьте, что на доске написана строка и двое играют с ней в следующую игру. Игроки делают ходы по очереди. Ход состоит в стирании некоторого непустого подмножества символов строки. При этом не разрешается стирать одним ходом два одинаковых символа. Например, из строки “abacaba” можно стереть символы с номерами 1, 2 и 4, но нельзя стереть символы с номерами 1, 3, 5, 6, потому что, например, первый и третий символы — одинаковые (“a”). Проигрывает тот, кто не может сделать ход, т.е. перед чьим ходом на доске не остается ни одного символа.

Ваша задача состоит в том, чтобы выбрать, за какого игрока играть (за первого или за второго) и обыграть программу жюри (интерактор) в эту игру.

Формат входного файла

Входные данные содержат непустую строку s длины до 500, состоящую из строчных латинских букв. Символы строки нумеруются целыми числами от 1 до ее длины. При стирании символов номера остальных символов *не меняются*.

При обмене информацией с интерактором на вход подаются ходы оппонента, по одному в строке. Описание хода содержит целое положительное число — количество элементов в стираемом подмножестве. Далее в той же строке должны быть перечислены номера стираемых символов в любом порядке. Числа в описании хода разделяются пробелами. Гарантируется, что ходы интерактора корректны — все стираемые символы различны, каждый номер выводится только один раз и не совпадает с номером никакого уже стертого ранее символа. Ваши ходы тоже должны удовлетворять этим условиям.

Если вы решаете играть за первого игрока, выведите свой первый ход, дальше прочитайте ход второго игрока, потом опять выведите свой ход и т.д. Если вы играете за второго игрока, сначала прочитайте ход оппонента, потом выведите свой ход, потом снова прочитайте ход оппонента и т.д.

Когда оппонент проигрывает, он вместо своего хода выводит одно число “-1”, после считывания которого ваша программа должна завершить свое выполнение.

Формат выходного файла

В начале взаимодействия с интерактором выведите номер игрока (1 или 2), за которого вы хотите играть. Учтите, что интерактор всегда играет оптимально и для решения задачи вам необходимо у него выиграть.

Далее выводите описания ходов по одному в строке (последовательность взаимодействия с интерактором описана в формате входных данных). После вывода каждой строки не забывайте делать flush.

Примеры

stdin	stdout
abacaba	1 1 4 2 2 5 2 1 6 1 7 1 3 -1
pp	2 1 2 1 1 -1

Примечание

Выходные данные к примерам содержат полный протокол взаимодействия решения с интерактором, т.е. первое число — номер игрока, далее по очереди идут ходы обоих игроков.