

Задача 01. Робот

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Варвара программирует своего учебного робота. Пока что ей доступны две команды: «один шаг вперёд» и «один шаг назад». Шаги робота в обоих направлениях имеют одну и ту же длину.

Варвара написала программу из этих команд, выполнив которую от начала до конца один раз, робот возвращается в исходную точку. Программа робота циклическая: каждый раз после выполнения последней команды робот переходит к началу программы и снова выполняет её, начиная с первой. Это продолжается, пока робота не выключат.

Варвара проверяла работу робота в поле, где вокруг него не было никаких препятствий. Однако теперь робот стоит спиной к забору, да так близко, что шаг назад из этого положения он сделать не сможет. Пространство же для движения вперёд можно считать не ограниченным.

Варвара решила начать выполнение программы с такой команды, чтобы робот никогда не врезался в забор. Помогите ей выбрать такое начало.

Формат входных данных

В первой строке входных данных записано целое число n — длина программы ($2 \leq n \leq 100$). Во второй строке задана сама программа — строка из n символов, после которых следует перевод строки. Символ «+» (плюс, ASCII-код 43) соответствует шагу робота вперёд, символ «-» (минус, ASCII-код 45) — шагу назад. Гарантируется, что программа не содержит других символов, а также что в результате однократного выполнения программы робот возвращается в исходную точку.

Формат выходных данных

В первой строке выведите одно число: с какой по счёту команды нужно начать выполнение циклической программы, чтобы робот не врезался в забор. Если правильных ответов несколько, выведите минимальный из них. Если же не врезаться в забор невозможно, выведите число -1 .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
6 +---+-	4
4 -+-+	2

Пояснения к примерам

В первом примере единственный возможный вариант — выполнять программу с четвёртой командой: «+-+---+-+---+-...».

Во втором примере можно было бы начать со второй или с четвёртой команды, при этом последовательность команд будет такой: «+-+---+-+---+-...». Из этих двух ответов следует вывести минимальный.

Задача 02. Сообщение об ошибке

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Правильные скобочно-палочные последовательности (ПСПП) определяются рекурсивно следующим образом:

1. Пустая строка является ПСПП.
2. Если A и B — две ПСПП, то AB (строка, полученная приписыванием строки B справа к строке A) — тоже ПСПП.
3. Если A и B — две ПСПП, то строка $(A|B)$ также является ПСПП.

Например, «(|)» является ПСПП, так как получается по пункту 3 определения для пустых строк A и B , которые являются ПСПП по пункту 1. Строка «(|)|(|)» также является ПСПП: по пункту 2 её можно составить из строк $A = (|)|$ и $B = (|)$. В свою очередь, строка «(|)|)» является ПСПП по пункту 3 определения для $A = (|)$ и пустой строки B .

Задана строка, состоящая из символов «(», «|» и «)». Если эта строка является ПСПП, выведите сообщение «correct, length = x », где x — длина строки. В противном случае выведите сообщение об ошибке вида «at position p : expected c_1 [or c_2 [or c_3]], found e ». Такое сообщение означает, что первые $(p - 1)$ символов строки являются ПСПП или началом какой-либо ПСПП, а на следующей позиции должен оказаться один из вариантов c_1 , c_2 или c_3 , чтобы строка была ПСПП или началом какой-либо ПСПП. Вместо этого на p -й позиции строки оказался вариант e . Позиции в строке нумеруются с единицы.

Во втором сообщении вместо каждого из обозначений вариантов — c_1 , c_2 , c_3 и e — стоит либо один из трёх возможных символов «(», «|» и «)», либо строка «END», обозначающая конец строки. Возможные значения упорядочены следующим образом: сначала «(», затем «|», далее «)» и, наконец, «END». Значения в списке c_i должны быть перечислены в соответствующем этому порядке.

Квадратные скобки означают, что часть сообщения, заключённая в них, может присутствовать, а может отсутствовать: например, если в какой-то позиции есть ровно два правильных варианта, сообщение выглядит как «at position p : expected c_1 or c_2 , found e ».

Формат входных данных

Единственная строка ввода имеет длину от 1 до 60 символов и состоит исключительно из символов «(», «|» и «)» (ASCII-коды 40, 124 и 41).

Обратите внимание: после всех этих символов следует перевод строки.

Формат выходных данных

Выведите сообщение о первой неправильной позиции в строке или о том, что строка является ПСПП. Сообщение должно иметь формат, указанный в условии. Пожалуйста, соблюдайте его как можно точнее.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
(())	correct, length = 6
())	at position 4: expected (or END, found)
((at position 3: expected (or , found END

Пояснения к примерам

В первом примере строка «(|(|))» целиком является ПСПП. Она получается по пункту 3 рекурсивного определения для пустой строки A и $B = (|)$.

Во втором примере строка «(|)|)» или, например, «(|)(|)» была бы ПСПП. Однако нет ПСПП, начинающейся на «(|)». Из возможных вариантов открывающая скобка «(» должна быть выведена раньше конца строки «END».

В третьем примере строка могла бы продолжаться как «(|)|)» или «(|)|)|)». Вместо этого на позиции 3 оказался конец строки, обозначаемый при выводе сообщения как «END». Из возможных вариантов открывающая скобка «(» должна быть выведена раньше вертикальной черты «|».

Задача G. Игра с окружностью

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На листе бумаги нарисована окружность, на которой отмечено n различных точек. Гидран и Диореп играют в игру, делая ходы по очереди. Ход состоит в том, чтобы выбрать две различные отмеченные точки и провести между ними хорду; это можно сделать, только если новая хорда не имеет общих точек (в том числе и концов) со всеми ранее проведёнными хордами. Проигрывает тот, кто не может сделать ход. Первым ходит Гидран. Кто выигрывает при правильной игре?

Формат входных данных

На ввод подаётся несколько строк. Первая из них содержит целое число t — количество тестовых случаев ($1 \leq t \leq 50$). Каждая из следующих t строк описывает один тестовый случай. Описание тестового случая состоит из одного целого числа n — количества отмеченных точек на окружности ($3 \leq n \leq 10^9$).

Формат выходных данных

Для каждого заданного n выведите на отдельной строке «53», если выигрывает Гидран, и «34», если выигрывает Диореп. Ответы должны быть выведены в том же порядке, в котором числа n заданы во вводе.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2	53
4	34
5	

Задача 03. Десятичная дробь

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В этой задаче требуется найти оптимальный период для бесконечной десятичной дроби.

Рассмотрим бесконечную десятичную дробь $x_0.x_1x_2x_3\dots$, которая является записью некоторого вещественного числа x от 0 до 1 включительно: $x = x_0 + x_1 \cdot 10^{-1} + x_2 \cdot 10^{-2} + x_3 \cdot 10^{-3} + \dots$. Здесь x_i — это десятичные цифры от 0 до 9. В этой задаче нет никаких ограничений на дробь, кроме приведённых выше. В частности, это означает, что, например, 0.999999... и 1.000000... — корректные бесконечные десятичные дроби, являющиеся записью одного и того же вещественного числа 1.

Периодическая десятичная дробь — это способ записи бесконечной десятичной дроби в виде $y_0.y_1y_2y_3\dots y_r(y_{r+1}y_{r+2}\dots y_s)$, где $r \geq 0$ и $s > r$. Эту запись можно *раскрыть* в бесконечную десятичную дробь $y_0.y_1y_2y_3\dots y_r y_{r+1}y_{r+2}\dots y_s y_{r+1}y_{r+2}\dots y_s y_{r+1}y_{r+2}\dots y_s \dots$, то есть бесконечную дробь, начинающуюся с $y_0.y_1y_2y_3\dots y_r$ и затем повторяющую последовательность цифр $y_{r+1}y_{r+2}\dots y_s$ в бесконечном цикле. Будем говорить, что r — это длина *предпериода*, а $s - r$ — это длина *периода*. Не всякую бесконечную десятичную дробь можно записать как периодическую. На самом деле такое представление существует тогда и только тогда, когда вещественное число x является рациональным.

Нам заданы несколько первых цифр бесконечной десятичной дроби, оставшиеся цифры просто отброшены (никакого округления не происходит). Теперь мы хотим записать какую-нибудь периодическую десятичную дробь, раскрыв которую, мы получим дробь, начинающуюся с заданной конечной части. Среди таких бесконечных десятичных дробей найдите ту, у которой сумма длин предпериода и периода минимально возможная.

Формат входных данных

Первая строка ввода содержит начало бесконечной десятичной дроби в формате $x_0.x_1x_2x_3\dots x_n$ ($1 \leq n \leq 1\,000\,000$). Здесь x_i — десятичные цифры от 0 до 9, а вещественное число x , записью которого является дробь, лежит между 0 и 1 включительно.

Формат выходных данных

Выведите одну строку, содержащую периодическую десятичную дробь в формате $y_0.y_1y_2y_3\dots y_r(y_{r+1}y_{r+2}\dots y_s)$, где $r \geq 0$ и $s > r$. Здесь y_i — десятичные цифры от 0 до 9. Раскрыв период, мы должны получить бесконечную десятичную дробь, начинающуюся с $x_0.x_1x_2x_3\dots x_n$ (это начало задано во вводе), а сумма длин предпериода и периода должна быть минимально возможной. Если возможных ответов несколько, выведите один любой из них. Гарантируется, что хотя бы один ответ существует.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
0.9999999	0.(9)
0.63573573	0.6(357)
0.123456789	0.12345(6789)

Пояснения к примерам

В первом примере периодическая десятичная дробь 0.(9) раскрывается в бесконечную десятичную дробь 0.999..., которая начинается с 0.9999999. Здесь длина предпериода равна 0, а длина периода равна 1. Другие ответы, например, 0.9(9) или даже 0.99999998(7), также раскрываются в дробь, начинающуюся с 0.9999999, но они не оптимальны. Заметим, что, хотя $0.9999999\dots = 1$ как вещественное число, ответ 1.(0) **не** является корректным, так как он раскрывается в дробь, которая не начинается на 0.9999999.

Во втором примере ответ 0.6(357) раскрывается в 0.6357357357357.... Здесь длина предпериода равна 1, а длина периода равна 3. Первые несколько цифр соответствуют заданному началу.

В третьем примере возможные ответы таковы: 0.(123456789), 0.1(23456789), ..., 0.12345678(9). Помните, что длина предпериода должна быть неотрицательна, а длина периода — положительна.

Задача 04. Мыльные пузыри

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

У Хексы день рождения!

Она решила порадовать себя совершенно необычным образом: надуванием мыльных пузырей. Мыльный пузырь представляет собой окружность на плоскости.

Для начала выбирается точка на оси Ox — центр надувательства. Да-да, все мыльные пузыри должны иметь один общий центр!

Затем последовательно надуваются пузыри: i -й пузырь надувается до тех пор, пока не коснётся точки (x_i, y_i) , либо пока не упрётся в другой — уже надутый — пузырь. Если один мыльный пузырь коснётся другого, то вся конструкция взорвётся мириадами фантазмагорически иллюминирующих брызг, чего никак нельзя допустить!

Интересно, а сколько существует порядков надувания мыльных шаров, при которых не происходит то, чего никак нельзя допустить?

Формат входных данных

В первой строке содержится n — количество надуваемых мыльных пузырей ($1 \leq n \leq 1000$). В последующих n строках записаны разделённые пробелами пары целых чисел: i -я пара содержит координаты i -й точки (x_i, y_i) ($1 \leq x_i, y_i \leq 100$).

Формат выходных данных

Выведите одно число: количество допустимых порядков надувания мыльных шаров.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3 3 7 3	2
3 1 1 2 4 9 2	4

Пояснения к примерам

В первом примере можно надувать пузыри как в порядке $(1, 2)$ (например, находясь в координате $x = 10$), так и в порядке $(2, 1)$ (например, находясь в координате $x = 0$).

Во втором примере возможные порядки — $(1, 2, 3)$, $(2, 1, 3)$, $(2, 3, 1)$ и $(3, 2, 1)$.

Задача 05. Сообщение на сервере

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Марк — шпион страны Бенджаминия, отправленный в страну Алевтинию с секретным заданием. Глубоко на территории чужой страны он пробрался в исследовательский центр и заполучил информацию, которую записал как сообщение из n двоичных цифр.

У Марка есть передатчик, который может загрузить на специальный сервер в интернете пакет ровно из $3n$ двоичных цифр. К сожалению, передатчик барахлит: известно, что он передаст только $2n + 2$ цифры, потеряв сколько-то (возможно, несколько) цифр в начале пакета и сколько-то (возможно, несколько) цифр в его конце. Сколько именно цифр передатчик потеряет в начале, а сколько в конце — неизвестно.

Позже, когда Марк вернётся в родную страну, он сможет скачать получившиеся $2n + 2$ цифры с сервера. Помогите ему придумать, какой сформировать пакет из $3n$ цифр, чтобы потом по оставшимся от него $2n + 2$ идущим подряд цифрам Марк смог в точности восстановить исходное сообщение из n двоичных цифр.

Протокол взаимодействия

В этой задаче ваше решение будет запущено на каждом тесте два раза.

При первом запуске в первой строке будет записано слово «upload». Вторая строка будет содержать целое число n ($2 \leq n \leq 50$). Третья строка будет содержать исходное сообщение из n двоичных цифр: символов «0» и «1». Вывести нужно одну строку из $3n$ двоичных цифр — пакет, передаваемый на сервер.

При втором запуске в первой строке будет записано слово «download». Вторая строка будет содержать целое число n ($2 \leq n \leq 50$) — то же, что и при первом запуске. Третья строка будет содержать $2n + 2$ символа — ту часть пакета, которую удалось передать на сервер при первом запуске. Нужно восстановить и вывести исходное сообщение из n двоичных цифр.

При всех запусках каждая строка входных данных, включая последнюю, завершается переводом строки.

Примеры

На каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске. В примерах мы рассмотрим решение, которое просто повторяет каждую цифру сообщения трижды. К сожалению, при $n > 4$ такое решение не всегда сможет восстановить сообщение.

Далее приведены два запуска этого решения на первом тесте. В этом тесте $n = 2$, поэтому $2n + 2 = 6$, и все шесть цифр пакета передаются на сервер.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
upload 2 01	000111
download 2 000111	01

Ниже показаны два запуска того же решения на втором тесте. В этом тесте $n = 4$, так что $2n + 2 = 10$. Тест устроен так, что при передаче теряются первая и последняя цифры.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
upload 4 1001	111000000111
download 4 1100000011	1001

Задача 06. Поездка

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В городе Ровный изобретатель Виктор испытывает беспилотный автомобиль «Астра» собственной конструкции. Город имеет форму прямоугольника, ровно расчерченного на $m \times n$ квадратных кварталов улицами, идущими с запада на восток и с севера на юг. Другими словами, в городе $(m + 1) \times (n + 1)$ перекрёстков. Каждый перекрёсток либо работает, либо закрыт на ремонт. При этом для каждого работающего перекрёстка среди его соседей есть хотя бы два, которые тоже работают.

У «Астры» есть вся информация о состоянии перекрёстков. Виктор знает, что поворот налево зачастую сложнее, чем поворот направо, так что он разработал следующий алгоритм для испытаний «Астры». Оказавшись на каком-то перекрёстке, автомобиль сначала проверяет, можно ли ехать вперёд: для этого впереди должен быть работающий перекрёсток. Если ответ утвердительный, «Астра» просто движется вперёд. В случае, если движение вперёд невозможно — либо «Астра» попала на границу города, либо следующий перекрёсток в этом направлении закрыт на ремонт — автомобиль аналогичным образом проверяет возможность повернуть направо. Если же и направо повернуть нельзя, «Астра» поворачивает налево. Поскольку у каждого работающего перекрёстка есть хотя бы два работающих соседа, в одном из этих трёх направлений обязательно можно проехать.

На финальной стадии испытаний автомобиль Виктора движется не только без водителя, но и без пассажиров. Виктор довёл автомобиль до перекрёстка r_0 -й улицы, идущей с запада на восток (эти улицы нумеруются от 0 до m с севера на юг), и c_0 -й улицы, идущей с севера на юг (эти улицы нумеруются от 0 до n с запада на восток). После этого Виктор развернул «Астру» так, чтобы впереди был восток, вышел и включил автопилот. Сам Виктор пошёл пить чай и освободится через t минут.

Известно, что в городе Ровный автомобиль проезжает от перекрёстка до перекрёстка за одну минуту. Подскажите Виктору, на каком перекрёстке должен оказаться его автомобиль через t минут.

Формат входных данных

В первой строке входных данных записаны два целых числа m и n — размеры города Ровный ($1 \leq m, n \leq 50$). Во второй строке записаны два целых числа r_0 и c_0 — координаты перекрёстка, где Виктор включил автопилот ($0 \leq r_0 \leq m, 0 \leq c_0 \leq n$). Соседние числа в строках разделены пробелом. В третьей строке записано целое число t — через сколько минут освободится Виктор ($1 \leq t \leq 1\,000\,000\,000$).

В следующих $m + 1$ строках задано состояние перекрёстков. Каждая строка содержит $n + 1$ символ и соответствует улице, идущей с востока на запад. Эти улицы перечислены в порядке с севера на юг. Работающий перекрёсток обозначается символом «+» (плюс, ASCII-код 43), а закрытый на ремонт — символом «x» (маленькая английская буква икс, ASCII-код 120). Последняя строка, как и все остальные, заканчивается переводом строки.

Гарантируется, что «Астра» начинает движение на работающем перекрёстке, а также что у каждого работающего перекрёстка есть хотя бы два работающих соседа.

Формат выходных данных

В первой строке выведите два числа r_t и c_t , разделив их пробелом: координаты перекрёстка, на котором окажется «Астра» через t минут.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>	Пояснение
2 1 1 0 3 ++ ++ ++	2 0	
2 3 2 3 8 ++xx ++++ x+++	2 1	

Задача 07. Поиск уникального элемента

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Это интерактивная задача.

А вы слышали о такой задаче: «Дан массив чисел, в котором все числа кроме одного, встречаются два раза, а оставшееся — один раз. Найти это число.»?

Вам предстоит решить почти такую же. В этой задаче массив чисел отсортирован, содержит все элементы кроме одного по два раза, оставшийся элемент содержит один раз, но он вам не дан! Вместо этого можно по одному запрашивать его элементы.

Протокол взаимодействия

В начале взаимодействия на вход вашей программе будет подано нечётное число n ($1 \leq n \leq 199\,999$) — длина массива.

После этого вы можете делать два типа запросов.

- «? x ». Запросить элемент массива на позиции x ($1 \leq x \leq n$). Разрешено сделать не более 40 таких запросов. При превышении этого лимита решение получит вердикт «Wrong Answer».
- «! v ». Ответить на задачу. Число v должно быть равно единственному элементу массива, который встречается один раз.

В ответ на запрос первого типа будет получена одна строка, в которой содержится соответствующий элемент массива. Это положительное целое число, не превосходящее 10^9 .

После выполнения запроса второго типа решение должно корректно завершиться.

Каждый запрос следует выводить на отдельной строке. Чтобы предотвратить буферизацию вывода, после каждого выведенного запроса следует вставить команду очистки буфера вывода: например, это может быть `fflush (stdout)` в C или C++, `System.out.flush ()` в Java, `flush (output)` в Pascal или `sys.stdout.flush ()` в Python.

Пример

запросы участника	ответы проверяющей программы	Загаданный массив
? 1	5	1 1 2 3 3
? 5	1	
? 3	3	
! 2	2	

Задача 08. Гирлянда

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Новогодняя гирлянда, установленная за окном Васиного дома, представляет из себя n лампочек, расположенных на одной прямой и пронумерованных подряд числами от 0 до $(n-1)$. Вася, глядя в окно, заинтересовался порядком включения лампочек. После внимательных наблюдений он установил следующее.

Изначально все лампочки выключены. Далее каждую секунду включается одна лампочка. В первую секунду включается лампочка с номером a , во вторую — с номером $a^2 \bmod n$, в третью — с номером $a^3 \bmod n$, ..., в k -ю — лампочка с номером $a^k \bmod n$.

Вася выяснил значения чисел a и n , и теперь его заинтересовал следующий вопрос. Рассмотрим лампочку, включённую в r -ю секунду (она имеет номер $a^r \bmod n$). Сколько лампочек слева от неё, то есть с меньшими номерами, уже включены? Вася считает, что именно свойства этих чисел для разных значений r и обеспечивают гирлянде особую красоту. Однако сам он будет слишком долго вручную их вычислять. Помогите Васе — в ответ на каждый заданный им вопрос о числе r выведите количество лампочек, зажжённых до r -й зажжённой лампочки и находящихся слева от неё.

Формат входных данных

В первой строке входных данных записаны целые числа a , n и q через пробел ($1 \leq a < n \leq 1\,000\,000$, $1 \leq q \leq 10\,000$). Следующие q строк содержат по одному целому числу r_i каждая ($1 \leq r_i \leq 100\,000$) и описывают Васины вопросы. Вася не знает, что будет, когда по этому алгоритму надо будет зажечь уже зажжённую лампочку, поэтому его вопросы таковы, что для любого i за первые r_i секунд никакая лампочка не должна будет загореться дважды.

Формат выходных данных

Выведите q строк, по одному числу в каждой строке. В i -й строке выведите количество лампочек, зажжённых раньше r_i -й зажжённой лампочки и имеющих меньшие номера.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3 2 1 2	0 0
5 16 4 1 3 4 2	0 2 0 1
2 8 2 2 2	1 1

Пояснения к примерам

В первом примере порядок зажигания лампочек — 2, 1.
Во втором примере — 5, 9, 13, 1.
В третьем примере — 2, 4, 0.

Задача 09. Трёхцветная улица

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Жители Трёхцветной улицы решили раскрасить свои дома, согласно названию, в три цвета — красный, зелёный и синий, каждый в свой цвет. При этом из эстетических соображений жители не хотят, чтобы какие-либо два стоящих рядом дома были покрашены в один и тот же цвет. Для дома номер i стоящими рядом с ним считаются дома $i - 1$ и $i + 1$; первый и последний дома имеют по одному соседнему дому.

Недавно состоялось собрание жильцов всех домов этой улицы, на котором было выяснено, во сколько обойдётся покраска каждого дома в каждый из трёх возможных цветов. Теперь жильцы обратились за помощью к вам, чтобы вы, располагая этой информацией, выбрали, в какой цвет красить каждый дом так, чтобы учесть эстетические соображения жильцов и при этом заплатить за покраску как можно меньшую сумму.

Формат входных данных

В первой строке ввода находится целое число N ($1 \leq N \leq 20$) — количество домов на Трёхцветной улице. В последующих N строках записано в каждой по три целых числа R_i , G_i и B_i через пробел — стоимость покраски i -го дома в красный, зелёный и синий цвета, соответственно. Известно, что $1 \leq R_i, G_i, B_i \leq 1000$.

Формат выходных данных

Выведите в единственной строке одно число — минимальную стоимость требуемой покраски.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 1 100 100 100 1 100 100 100 1	3
3 1 100 100 100 100 100 1 100 100	102
3 26 40 83 49 60 57 13 89 99	96
6 30 19 5 64 77 64 15 19 97 4 71 57 90 86 84 93 32 91	208
8 71 39 44 32 83 55 51 37 63 89 29 100 83 58 11 65 13 15 47 25 29 60 66 19	253

Пояснения к примерам

В первом примере дешевле всего будет покрасить первый дом в красный, второй — в зелёный и третий — в синий цвет.

Во втором примере первый и третий дома надо красить в красный цвет, а второй — в зелёный или синий.

Задача 10. И снова сумма...

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Реализуйте структуру данных, которая поддерживает множество S целых чисел, с которым разрешается производить следующие операции:

- `add (i)` — добавить в множество S число i (если оно там уже есть, то множество не меняется);
- `sum (l, r)` — вывести сумму всех элементов x из S , которые удовлетворяют неравенству $l \leq x \leq r$.

Исходно множество S пусто.

Формат входных данных

Первая строка входных данных содержит целое число n — количество операций ($1 \leq n \leq 300\,000$). Следующие n строк содержат операции. Каждая операция записывается либо как «+ i », либо как «? l r ».

Операция «? l r » задаёт запрос `sum (l, r)`.

Если операция «+ i » появляется в начале входных данных или после другой операции «+», то она задаёт операцию добавления `add (i)`. Если же она следует сразу после запроса «?», и результат этого запроса равен y , то следует выполнить операцию добавления `add ((i + y) mod 109)`.

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до 10^9 включительно.

Формат выходных данных

Для каждого запроса выведите одно число — ответ на этот запрос.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
6	3
+ 1	7
+ 3	
+ 3	
? 2 4	
+ 1	
? 2 4	

Задача 11. Бублики

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В сказочной стране пекарей деньги традиционно называются «бубликами». В этой стране в ходу n различных монет номиналом в v_1, v_2, \dots, v_n бубликов. Номиналы таковы, что для любых двух монет достоинство одной из них делится нацело на достоинство другой.

Вас попросили написать программу для автомата, выдающего суммы бубликов монетами. Для удобства пользователей автомат должен выдавать любую сумму при помощи минимально возможного для этой суммы количества монет.

Считая, что количество монет каждого достоинства в автомате не ограничено, выясните, как ему выдать w бубликов минимальным количеством монет.

Формат входных данных

В первой строке заданы два целых числа n и w через пробел — количество различных достоинств монет и сумма, которую необходимо набрать, соответственно ($1 \leq n \leq 30, 1 \leq w \leq 10^9$). Во второй строке заданы n различных целых чисел v_1, v_2, \dots, v_n через пробел — достоинства монет ($1 \leq v_i \leq 10^9$). Известно, что для любых двух различных достоинств одно из них делится на другое. Одно из заданных достоинств равно единице.

Формат выходных данных

В первой строке выведите одно число k — минимальное количество монет, при помощи которого можно набрать необходимую сумму. Во второй строке перечислите достоинства этих k монет в любом порядке; сумма этих достоинств должна быть равна w . Если оптимальных ответов несколько, можно вывести любой из них.

Тесты в этой задаче таковы, что в оптимальном ответе число k не превысит 10 000.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 90 1 2 10 50	5 10 10 10 10 50
4 3 1 2 10 50	2 2 1

Задача 12. Кротовья нора

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Кротовья нора состоит из пещерок, соединённых коридорами, причём из каждой пещерки выходит ровно три коридора. Коридорами и пещерками пользуется не только крот – в одной из пещерок обосновался жук-носорог, а в другую только что свалился жук-олень. Жук-олень чует своего соперника и спешит встретиться, чтобы выяснить, кто из них будет главным в кротовьей норе!

Выясните, каким минимальным количеством коридоров потребуется пройти жуку-оленю, чтобы попасть в пещерку, где его поджидает жук-носорог.

Формат входных данных

В первой строке заданы три числа: N ($4 \leq N \leq 1000$) – количество пещерок в кротовьей норе, X ($1 \leq X \leq N$) – номер пещерки жука-носорога и Y ($1 \leq Y \leq N$) – номер пещерки жука-оленя. В последующих N строках стоит по три числа в каждой: в $(i + 1)$ -й строке стоят номера пещерок, в которые из i -й пещерки ведут коридоры. Каждый коридор упоминается в строках обеих вершин, которые он соединяет. Никакой коридор не соединяет пещерку с самой собой, и никакие две пещерки не соединены более чем одним коридором. Гарантируется, что $X \neq Y$. Все числа во входных данных целые.

Формат выходных данных

Выведите одно целое число – минимальное количество коридоров, которыми требуется пройти, чтобы попасть из пещерки жука-оленя в пещерку жука-носорога. Если это оказалось невозможно, выведите число -1 .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 1 4 2 3 4 3 4 1 4 1 2 1 2 3	1
6 2 3 4 5 6 4 5 6 4 5 6 1 2 3 1 2 3 1 2 3	2
8 5 4 2 3 4 1 3 4 1 2 4 1 2 3 6 7 8 5 7 8 5 6 8 5 6 7	-1

Задача 13. Игра

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

— Но как он это делает! Он забирается на самую высокую сосну и оттуда планирует.
— Ага. Простите, что планирует?

«День радио»

Девочка Наташа готовит поле для очень интересной игры. В ней примут участие k команд, каждая из которых должна получить в своё распоряжение одно или несколько деревьев и верёвку. При этом у каждой из команд должна быть возможность с помощью верёвки добраться от любого своего дерева до любого другого своего дерева, не используя чужие деревья, но, возможно, используя другие свои деревья как промежуточные. Будем считать, что с помощью верёвки можно перебраться с одного дерева на другое напрямую, если её длина не меньше расстояния между ними.

Длина всех верёвок должна быть одинаковой, чтобы поставить команды в равные условия. Разделите все доступные n деревьев на k наборов так, чтобы необходимая длина верёвок оказалась как можно меньшей.

Формат входных данных

В первой строке записаны целые числа n и k — количество деревьев и команд, соответственно ($1 \leq k \leq n \leq 1000$).

В каждой из следующих n строк записано по два целых числа x_i и y_i — координаты i -го дерева ($-10^4 \leq x_i, y_i \leq 10^4$).

Формат выходных данных

В первой строке выведите одно вещественное число с шестью или более точными знаками после десятичной точки — минимально возможную длину верёвок. Во второй строке выведите n целых чисел от 1 до k — номера команд, которым следует присвоить соответствующие деревья.

Если решений несколько, выведите любое.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 2	1.00000000
1 0	1 2 1 2
0 1	
1 1	
0 0	