

А. Наибольшая цифра

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 киббайт**

Михаил – юный программист, и он решает одну задачку. Условие задачи простое: «Дано число, состоящее не более чем из 3 цифр (Михаил умеет считать только до 1000), без ведущих нулей, необходимо вывести наибольшую из крайних цифр».

Михаил уже написал программу и хочет с кем-нибудь сверить ответы. Напишите программу, выводящую наибольшую из крайних цифр заданного числа.

Входные данные

Одно целое положительное число меньше 1000.

Выходные данные

Одну цифру – ответ на задачу.

Примеры

Ввод	Вывод
160	1
19	9
3	3

В. Вода

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 кибибайт**

Василий живёт в деревне и ему время от времени нужно по хозяйству наполнить водой ёмкость (до краёв), представляющую собой прямоугольный параллелепипед со сторонами: L (длина), W (ширина) и H (высота). Для переноса воды из колодца в ёмкость он будет использовать ведро, имеющее форму усечённого конуса с диаметрами оснований d_1 (дно ведра) и d_2 (верх ведра) и высотой h .

Что бы оценить количество времени, которое Василий потратит на заполнение ёмкости и назначить встречу, на которую он придёт вовремя, ему нужно оценить, какое минимальное количество полных (до краёв) вёдер ему необходимо будет принести, чтобы заполнить ёмкость.

Помогите Василию и другим ребятам (в каждом доме в хозяйстве имеются различные по габаритам ёмкости и вёдра), напишите программу, которая по заданным L, W, H, d_1, d_2, h будет находить минимальное количество полных вёдер воды, которые нужно принести, чтобы до краёв заполнить ёмкость.

Входные данные

В первой строке через пробел даны целые числа L, W, H ($1 \leq L, W, H \leq 10^3$).

Во второй строке через пробел даны целые числа d_1, d_2, h ($1 \leq d_1, d_2, h \leq 10^3$).

Выходные данные

Одно целое число – ответ на задачу.

Пример

Ввод	Вывод
10 10 10 10 10 10	2

С. Счастливые билеты

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 киббайт**

«Счастливые билеты» – это такие билеты, номера которых имеют чётное количество цифр, и сумма половины цифр слева равна сумме половины цифр справа.

Иван очень добрый парень, и он, видя, как расстраивается его младшая сестрёнка, когда он приносит «несчастливый билет», который он купил при проезде в автобусе, решил исправлять номер так, чтобы «несчастливые билеты» становились «счастливыми».

Он так качественно исправляет номера, что его сестрёнка не может заметить, и она радуется каждый день новым «счастливым билетикам».

Дело в том, что на исправление одной цифры уходит не менее получаса, помогите Василию и напишите программу, которая будет указывать какой номер должен быть на билетике, при этом количество исправляемых цифр должно быть минимальным. Если билет и так является «счастливым», то выведите номер без исправлений.

Входные данные

Одно шестизначное число, возможно с ведущими нулями.

Выходные данные

Одно шестизначное число, возможно с ведущими нулями.

Если минимальных решений может быть несколько, выведите только одно любое из них.

Примеры

Ввод	Вывод
123600	123600
123500	023500

D. Штрих-код

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 кибибайт**

Марфуша и Феклуша прочитали статью про штрих-кодирование. Идея им очень понравилась, но им непонятны вычисления контрольных сумм, которые используются для помехоустойчивого кодирования. Поэтому они решили взять за основу одну из кодировок цифр используемых в стандарте UPC-A и не использовать какие-либо шаблоны.

Для передачи цифрового кода друг другу они берут листочек в клеточку и рисуют линии маркерами жёлтого и синего цветов в соответствии с кодировкой. Для кодирования нуля они используют жёлтый цвет (YELLOW), а для кодирования единицы – синий цвет (BLUE).

Ваша задача: по последовательности, состоящей из жёлтых и синих линий восстановить цифровое сообщение. Кодовые последовательности, которые используются при кодировании, приведены в таблице ниже.

Цифры	Последовательности	Цифры	Последовательности
0	Y Y V B V Y V	5	Y V B Y Y Y V
1	Y Y V B Y Y V	6	Y V Y V B B V
2	Y Y V Y Y V B	7	Y V B B Y V B
3	Y V B B V Y V	8	Y V B Y V B V
4	Y V Y Y Y V B	9	Y Y Y V Y V B

Входные данные

Одна строка, состоящая только из символов: «Y» и «V». Символ «Y» обозначает жёлтый цвет, а символ «V» – синий цвет. В «цветовом» штрих-коде закодировано не более 1000 цифр.

Выходные данные

Выведите одну строку – последовательность цифр, соответствующую закодированному посланию.

Примеры

Ввод	Вывод
Y Y V Y Y V B Y Y Y V Y V B	29
Y Y Y V B Y V Y Y Y V B Y V Y Y Y V B	009

Е. Мишень

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 киббайт**

Василий разрабатывает приложение «Мишень» для тренировки меткости стрельбы. Оно должно фиксировать попадание в цель, а при промахе, но попадании в стенд, давать корректирующие сообщения, указывать, куда нужно сместить прицел.

Стенд мишени представляет собой квадрат с длиной стороны равной 2000 мм. Цель имеет форму круга с радиусом R ($1 \leq R \leq 100$), центр круга (цели) совпадает с пересечением диагоналей квадрата (стенда).

Если стрелок попал в цель, то программа должна вывести «OK» (две латинские заглавные буквы без кавычек). Если промахнулся, то программа должна вывести информацию о том, в каком направлении следует сместить прицел: «UP» – вверх, «DOWN» – вниз, «LEFT» – влево, «RIGHT» – вправо (все надписи выводятся без кавычек). Если необходимо сместить одновременно в двух направлениях, то необходимо через пробел вывести оба направления, сначала по вертикали, затем по горизонтали.

Входные данные

Две строки, на первой одно целое положительное число – радиус цели, на второй два вещественных числа через пробел – координаты, куда попал стрелок, сначала координата по горизонтали, затем по вертикали. Координаты даны в миллиметрах с точностью до 6-ти знаков после запятой, по абсолютному значению все величины не превосходят одной тысячи.

Выходные данные

Одна строка – результат работы программы по вышеописанному алгоритму.

Примеры

Ввод	Вывод
10 10 0	OK
100 0 -100.000001	UP
100 -100.000001 0	RIGHT
5 4 3	OK
5 4.000001 3	DOWN LEFT
4 -3.141592 -2.718281	UP RIGHT

Ф. Делимость

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 кибибайт**

Василий, изучая историю вычислительной техники СССР, узнал, что были разработки компьютеров, работающих на физическом уровне на основе троичной логики. В этой системе счисления «круглыми числами» являются те, которые делятся на три (в десятичной системе «круглыми» называются те, которые делятся на 10, бывают и другие определения «круглых чисел», в это задаче следует руководствоваться тем, что дано в задаче).

В это же самое время Василий так же изучает очень большие числа, сопоставимые с гуголом – числом, равным 10^{100} . Ему интересно, является ли «круглым» в троичной системе счисления текущее изучаемое им число.

Входные данные

Одна строка – последовательно (от старших разрядов к младшим) расположенные через пробел десятичные цифры изучаемого Василием числа (количество цифр не более 100 и не менее 90, первые два теста имеют количество знаков менее 90, смотри примеры).

Выходные данные

Выведите «Round number», если число является «круглым» в троичной системе счисления, в противном случае выведите «number», кавычки выводить не надо.

Примеры

Ввод	Вывод
3 1 4 1 5 9 2 6 5	Round number
3 1 4	number