

Задача Е. Верное равенство

Имя входного файла: e.in
Имя выходного файла: e.out
Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 мегабайта

Возвращаясь из школы домой, Петя каждый раз обращал внимание на надпись на заборе « $1 + 1 = 10$ » и удивлялся очевидной его неправоте. Но однажды его осенило, что это равенство верное, если рассматривать его в двоичной системе счисления. Его настолько поразила эта идея, что он решил непременно придумать свои три числа так, чтобы сумма первых двух была равна третьему в некоторой системе счисления.

Теперь он перебирает тройки чисел, которые, на его взгляд, достойны находиться на заборе. Петя выбирает числа A , B , C , записывающиеся десятичными цифрами, и дальше пытается найти основание системы счисления K , в которой равенство $A + B = C$ оказалось бы верным. Петя рассматривает системы счисления с основанием от 2 до бесконечности.

Поскольку проверка каждой тройки — занятие трудоемкое, в помощь Пете необходимо написать программу, облегчающую расчеты.

Формат входных данных

В первой строке содержится число A , состоящее из цифр от 0 до 9 длины не более 200. В следующих двух строках в таком же формате записаны числа B и C .

Все числа неотрицательные и без ведущих нулей.

Формат выходных данных

Выведите минимальное основание системы счисления, в которой выполняется равенство $A + B = C$. Если такого не существует, то выведите 0.

Примеры

e.in	e.out
9 8 17	10
9 8 11	16
5 5 1010	0
0 0 0	2

Частичные ограничения

Первая группа состоит из тестов, в которых у всех трех чисел количество цифр не превышает 5, а при сложении их «столбиком» в искомой системе счисления не происходит переноса в следующий разряд.

Вторая группа состоит из чисел, при переводе которых из искомой системы счисления в десятичную они не будут превышать 10^9 .

Задача F. Симпатичные прямоугольники

Имя входного файла: `f.in`
Имя выходного файла: `f.out`
Максимальное время работы на одном тесте: 2 секунды
Максимальный объем используемой памяти: 64 мегабайта

Знаменитый художник Вася только что закончил работу над своим новым шедевром и хочет знать, сколько он сможет получить за свой труд.

Картина представляет собой прямоугольник N на M сантиметров, разделенный на маленькие квадратики 1 на 1 сантиметр со сторонами, параллельными сторонам картины. Для достижения гармонии каждый из этих квадратиков Вася покрасил одним из 26 особых цветов, обозначаемых маленькими латинскими буквами.

Стоимость картины в точности равна количеству «симпатичных» частей в ней. Частью картины называется любой прямоугольник, который может быть вырезан из нее по границам квадратиков. Часть называется «симпатичной», если при выполнении симметрии относительно ее центра получается прямоугольник, раскрашенный также, как и исходная часть. Например, в картине, раскрашенной так:

abc
acb

симпатичными являются все части, состоящие из одного квадратика (их 6), а также части

bc и a
cb a

Напишите программу, которая по информации о шедевре Васи определит его стоимость.

Формат входных данных

В первой строке содержатся два числа N и M ($1 \leq N, M \leq 100$). В следующих N строках идут строки, состоящие из M маленьких латинских символов. Символ в i -й строке j -м столбце определяет цвет соответствующего квадратика картины.

Формат выходных данных

Выведите стоимость шедевра — количество частей, симметричных относительно своего центра.

Примеры

<code>f.in</code>	<code>f.out</code>	Комментарии
2 3 abc acb	8	Этот пример разобран в условии.
3 2 ab cc ba	8	Симпатичными являются шесть частей 1×1 , одна часть 1×2 и сама картина.

Частичные ограничения

Первая группа состоит из тестов, в которых $N, M \leq 15$.

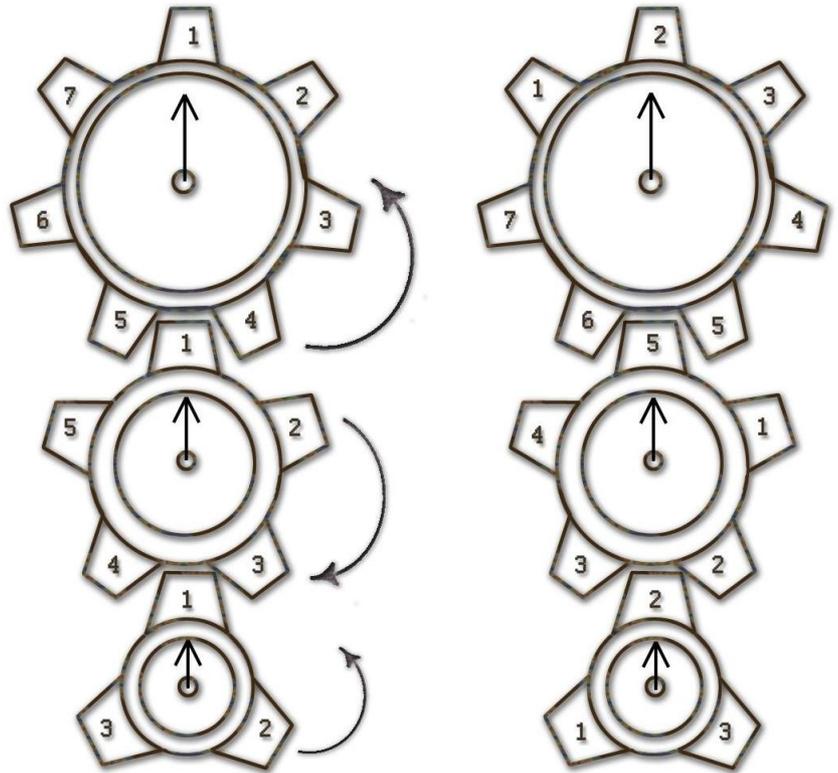
Вторая группа состоит из тестов, в которых $N, M \leq 50$.

Задача G. Шестеренки

Имя входного файла: `g.in`
Имя выходного файла: `g.out`
Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 мегабайта

На каждой из трех осей установлено по одной вращающейся шестеренке и неподвижной стрелке. Шестеренки соединены последовательно. На первой шестеренке n зубцов, на второй — m , на третьей — k .

На каждом зубце первой, второй и третьей шестеренок по часовой стрелке написаны в порядке возрастания числа от 1 до n , от 1 до m и от 1 до k соответственно. Стрелки зафиксировали таким образом, что когда стрелка первой оси указывает на число, стрелки двух других осей также указывают на числа. Витя записывает три числа (a_1, a_2, a_3) , на которые указывают стрелки. После этого он поворачивает первую шестеренку на угол $360^\circ/n$ против часовой стрелки, чтобы напротив стрелки на первой оси оказался следующий (по часовой стрелке) зубец. При этом вторая шестеренка поворачивается на угол $360^\circ/m$ по часовой стрелке (размеры зубцов у шестеренок одинаковые, поэтому размеры самих шестеренок разные, чтобы на границе шестеренок равномерно уложилось разное число одинаковых по размеру зубцов), а третья шестеренка поворачивается на угол $360^\circ/k$ против часовой стрелки. Витя опять записывает три числа, на которые указывают стрелки.



Поступая и далее таким образом, Витя заметил, что после некоторого количества таких действий стрелки показывают на три первоначальных числа.

Теперь его очень интересует, как по двум данным тройкам чисел определить, принадлежат ли они к одной последовательности, то есть можно ли некоторым количеством поворотов перейти от первой тройки ко второй. Он попросил вас написать программу, которая отвечает на этот вопрос.

Формат входных данных

В первой строке содержатся четыре числа T, n, m, k ($1 \leq T \leq 10, 1 \leq n, m, k \leq 10^{18}$) — количество пар троек, которые хочет проверить Витя и количества зубцов соответственно на первой, второй и третьей шестеренках.

В следующих T строках записаны шесть натуральных чисел $a_1, a_2, a_3, b_1, b_2, b_3$ ($1 \leq a_1, b_1 \leq n, 1 \leq a_2, b_2 \leq m, 1 \leq a_3, b_3 \leq k$).

Формат выходных данных

Для каждой пары троек в выходной файл выведите YES, если обе тройки принадлежат одной последовательности, и NO иначе. Каждое слово должно быть в отдельной строке, в порядке, соответствующем входному файлу.

Примеры

g.in	g.out	Комментарии
3 11 13 15 5 5 5 6 4 6 11 13 15 1 12 1 2 13 2 1 1 1	YES YES YES	В первой и второй парах вторая тройка получается из первой за один поворот первой шестеренки против часовой стрелки. В третьем случае из второй конфигурации просто получить первую опять же одним поворотом первой шестеренки против часовой стрелки. Очевидно, что тогда из первой можно каким-то образом получить вторую.
2 2 2 2 1 1 1 1 1 2 1 1 1 2 2 2	NO YES	В первой паре тройки нельзя перевести друг в друга. Во второй тройки переходят друг в друга при одном повороте.
1 7 5 3 1 1 1 2 1 1	YES	(1, 1, 1) — (семь поворотов первой против часовой стрелки шестеренки) — (1, 4, 2) — (еще семь таких же поворотов) — (1, 2, 3) — (один поворот) — (2, 1, 1)

Частичные ограничения

Первая группа состоит из тестов, в которых произведение $nmk \leq 10^6$.

Вторая группа состоит из тестов, в которых $n, m, k \leq 10^9$.

Примечание

Не гарантируется, что описанную в условии задачи систему шестеренок и стрелок можно реально построить.

Задача Н. Великий Флатландский Забор

Имя входного файла: `h.in`
Имя выходного файла: `h.out`
Максимальное время работы на одном тесте: 2 секунды
Максимальный объем используемой памяти: 64 мегабайта

С незапамятных времен граница страны Флатландия имеет форму N -угольника без самопересечений и самокасаний (необязательно выпуклого). В каждой из вершин этого многоугольника король построил по башне.

В целях увеличения обороноспособности государства на его территории король задумал построить Великий Флатландский Забор. По причине многолетней войны ресурсов хватает на строительство ровно K стен (неважно, какой длины). Каждая стена должна соединять ровно две башни по прямой и не должна даже частично выходить за пределы Флатландии. К тому же, Забор должен представлять собой не более чем K -угольник также без самопересечений и самокасаний (углов может оказаться и меньше, чем K , так как некоторые соседние стены могут лежать на одной прямой). Военный советник настаивает на том, что площадь защищенной области должна быть как можно больше.

Ваша задача — помочь спроектировать такой Забор.

Формат входных данных

В первой строке записаны два целых числа N и K ($3 \leq K \leq N \leq 230$). В следующих N строках содержатся пары целых чисел — координаты башен в порядке обхода границы против часовой стрелки. Гарантируется, что никакие три последовательные башни не лежат на одной прямой. Все координаты не превосходят 1000 по абсолютной величине.

Формат выходных данных

В первую строку запишите максимальную площадь, которую можно защитить Забором, с точностью до пяти знаков после десятичной точки. Во второй строке должно быть Q — количество выбранных башен.

Будем считать, что башни занумерованы числами от 1 до N в порядке перечисления их во входном файле. Во третью строку через пробел выведите номера Q башен, которые будут вершинами Забора, в порядке его обхода против часовой стрелки.

Примеры

<code>h.in</code>	<code>h.out</code>
3 3 0 0 1 0 0 1	0.50000 3 2 3 1
6 5 0 0 7 0 4 3 4 4 3 4 0 7	24.500000 3 1 2 6
4 3 0 0 -2 -2 1 3 0 2	2.000000 3 1 3 4

Частичные ограничения

Первая группа состоит из тестов, в которых $N \leq 20$ и граница Флатландии представляет собой выпуклый многоугольник.

Вторая группа состоит из тестов, в которых $N \leq 20$, но граница может быть уже любой.