Задача 1. Беговая дорожка

Длина круговой беговой дорожки стадиона составляет s метров. Два бегуна выбегают из одной точки и бегут в противоположных направлениях. Скорости бегунов составляют u и v м/с. Определите, через сколько секунд бегуны встретятся во второй раз после старта.

Ответом на эту задачу является некоторое выражение, которое может содержать целые числа, переменные s, u и v (записываемые английскими буквами), операции сложения (обозначаются «+»), вычитания (обозначаются «−»), умножения (обозначаются «*»), деления (обозначаются «/») и круглые скобки для изменения порядка действий. Запись вида «2u» для обозначения произведения числа 2 и переменной u неверная, нужно писать «2 * u».

Пример правильной формы записи ответа.

$$u/2 + (s * u - v) * 2$$

Задача 2. Торты

Кондитерская фабрика выпускает несколько видов тортов. Сначала торт выпекает кондитер, после чего готовый продукт поступает к упаковщику. Кондитер и упаковщик одновременно могут работать только с одним тортом. Торт поступает к упаковщику только после завершения работы над ним кондитера. Для каждого торта известно время (в минутах), которое необходимо кондитеру и упаковщику для работы с этим тортом. Вам необходимо определить порядок, в котором должно производиться изготовление тортов (необходимо произвести по одному торту каждого вида), так, чтобы суммарное время производства было минимальным. Время считается от момента начала работы кондитера над первым тортом до окончания работы упаковщика над последним тортом.

Рассмотрим следующий пример из двух тортов.

Номер торта	Время работы кондитера	Время работы упаковщика
1	20	15
2	10	5

Если сначала начать изготавливать первый торт, потом второй, то через 20 минут после начала кондитер закончит работать над первым тортом и начнёт работать над вторым тортом, а упаковщик начнёт упаковывать первый торт. Упаковщик закончит упаковывать первый торт через $20+15 = 35$ минут. Кондитер закончит работать над вторым тортом через $20 + 10 = 30$ минут, но поскольку упаковщик в этот момент занят, то упаковщик начнёт упаковывать второй торт через 35 минут после начала и закончит через $35 + 5 = 40$ минут.

Если сначала изготавливать второй торт, то через 10 минут после начала кондитер закончит работать над вторым тортом. Упаковщик закончит упаковывать второй торт через $10 + 5 = 15$ минут, а кондитер закончит готовить первый торт через $10+20 = 30$ минут. В этот момент упаковщик свободен и сразу же начнёт упаковывать второй торт и закончит упаковку через $30 + 15 = 45$ минут. Таким образом, наиболее оптимальным порядком изготовления тортов будет «1 2».

Вам даны следующие 6 видов тортов.

Номер торта	Время работы кондитера	Время работы упаковщика
1	50	40
2	20	10
3	15	15
4	35	20
5	10	35
6	35	45

Определите порядок, в котором необходимо изготавливать торты для того, чтобы сделать это за наименьшее время.

Ответ на эту задачу необходимо записать в виде некоторой перестановки чисел 1, 2, 3, 4, 5, 6, записанных через пробел. Пример записи правильного ответа для приведённого выше примера для двух тортов: «1 2».

Задача 3. Много пирожных

На кондитерской фабрике есть некоторое количество пирожных нескольких разных видов. Пирожных разных видов может быть разное количество. Было принято решение отвезти пирожные на продажу на ярмарку, но директор фабрики решил, что кондитерские изделия на ярмарочной витрине должны быть выложены одинаковыми рядами, при этом пирожных каждого вида должно быть одинаковое количество. Необязательно отвозить на ярмарку все виды пирожных, можно выбрать некоторые виды и взять одинаковое число пирожных каждого выбранного вида.

Помогите директору отвезти на ярмарку наибольшее число пирожных — найдите, сколько видов пирожных и сколько пирожных каждого вида нужно отвезти на ярмарку.

Например, если на фабрике есть три вида пирожных количеством 4, 10 и 7 штук, то у директора фабрики есть следующие возможности.

а) Взять только один вид пирожных, тогда он сможет взять 10 пирожных одного вида.

б) Взять два вида пирожных, тогда он сможет взять по 7 пирожных второго и третьего вида, всего 14 пирожных.

в) Взять три вида пирожных, тогда он сможет взять только по 4 пирожных каждого вида, всего 12 пирожных.

Больше всего пирожных получится в случае б).

Вам необходимо решить эту задачу для следующих пяти примеров набора пирожных на фабрике. В примере указано количество пирожных каждого вида.

Пример 1: 1, 2, 3, 4, 10 (всего 5 видов пирожных).

Пример 2: 5, 4, 9, 7, 1, 3 (всего 6 видов пирожных).

Пример 3: 1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 5 (всего 15 видов пирожных).

Пример 4: 1, 2, 3, ..., 99 (все числа от 1 до 99, всего 99 видов пирожных).

Пример 5: 51, 53, 55, ..., 99 (все нечётные числа от 51 до 99, всего 25 видов пирожных).

Для каждого примера определите, какое количество видов пирожных необходимо выбрать и сколько пирожных каждого выбранного вида нужно выбрать, чтобы максимизировать общее число выбранных пирожных.

Ответ запишите в виде пяти строк, одна строка содержит ответ на один пример в виде двух чисел, записанных через пробел. Первое число — количество выбранных видов пирожных, второе число — количество пирожных каждого выбранного вида, которое необходимо взять. Например, для примера 4, 10, 7 ответ нужно записать в виде:

2 7

Вы должны записать в ответе ровно пять строк. Если Вы не можете найти ответ на какой-то из пяти примеров, запишите в этой строке два любых целых положительных числа.

Задача 4. Код, исправляющий ошибки

При двоичном кодировании буквам сопоставляются последовательности из символов «0» или «1». Например, рассмотрим код, в котором буква А кодируется последовательностью «000», а буква Б — последовательностью «111».

000
111

Этот код обладает следующим свойством: он исправляет одну ошибку, то есть при изменении не более одного переданного символа всё равно можно восстановить переданное кодовое слово. Например, если в последовательности «000» изменить один символ, то может получиться одна из следующих последовательностей: «100», «010», «001». А если изменить один символ в последовательности «111», то может получиться одна из следующих последовательностей «011», «101», «110».

Про такой код мы будем говорить, что это код длины 3 (все кодовые слова состоят из трёх символов) мощности 2 (мы построили два кодовых слова), исправляющий одну ошибку.

Нельзя составить код длины 2 мощности 2, исправляющий одну ошибку. Например, если взять кодовые слова «00» и «11», то при получении последовательности «01» непонятно, какая последовательность была передана: это могла быть как последовательность «00», так и последовательность «11». Несложно заметить, что код может исправлять одну ошибку, если любые два кодовых слова различаются как минимум в трёх позициях (то есть в приведённой таблице для любых двух выбранных строк верно свойство: найдётся три таких столбца, что в этих столбцах в двух выбранных строках записаны разные символы).

Приведём пример кода длины 5 мощности 3, исправляющий одну ошибку:

00000
00111
11100

В приведённой таблице для любых двух выбранных строк верно свойство: найдётся три таких столбца, что в этих столбцах в двух выбранных строках записаны разные символы. Поэтому этот код исправляет одну ошибку.

Вам необходимо построить код длины 6, исправляющий одну ошибку, при этом код должен содержать как можно больше кодовых слов.

В качестве ответа Вам нужно записать несколько кодовых слов, каждое кодовое слово в отдельной строке. Кодовое слово должно содержать ровно шесть символов «0» или «1».

Пример записи в ответе кода из двух кодовых слов:

000000
111111

Чем больше кодовых слов Вы сможете записать (при условии, что полученный код будет исправлять одну ошибку), тем больше баллов Вы получите. Ответ, в котором всего лишь два кодовых слова, будет оцениваться в 0 баллов.

Задача 5. Метро

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

На некоторых кросс-платформенных станциях метро (как, например, «Третьяковская») на разные стороны платформы приходят поезда разных направлений. Таня договорилась встретиться с подругой на такой станции, но поскольку подруга приехала из другого часового пояса, то из-за джетлага сильно проспала, и Тане пришлось долго её ждать. Поезда всегда ходят точно по расписанию, и Таня знает, что поезд стоит на платформе ровно одну минуту, а интервал между поездами (время, в течение которого поезда у платформы нет) составляет a минут для поездов на первом пути и b минут для поездов на втором пути. То есть на первый путь приезжает поезд и стоит одну минуту, затем в течение a минут поезда у платформы нет, затем в течение одной минуты у платформы стоит следующий поезд и т. д.

Пока Таня стояла на платформе, она насчитала n поездов на первом пути и m поездов на втором пути. Определите минимальное и максимальное время, которое Таня могла провести на платформе, или сообщите, что она точно сбилась со счёта.

Все поезда, которые видела Таня, она наблюдала в течение всей минуты, то есть Таня не приходит и не уходит с платформы посередине той минуты, когда поезд стоит на платформе.

Формат входных данных

Первая строка входных данных содержит число a — интервал между поездами на первом пути. Вторая строка содержит число b — интервал между поездами на втором пути. Третья строка содержит число n — количество поездов на первом пути, которые увидела Таня. Четвёртая строка содержит число m — количество поездов на втором пути, которые увидела Таня. Все числа — целые, от 1 до 1000.

Формат выходных данных

Программа должна вывести два числа: минимальное и максимальное время в минутах, которое Таня могла стоять на платформе, или одно число -1, если Таня точно ошиблась.

Примеры

стандартный ввод	стандартный вывод
1 3 3 2	5 7
1 5 1 2	-1

Замечание

В первом примере по первому пути поезда ходят через 1 минуту. По второму — через 3. Стоя на платформе 5, 6 или 7 минут, Таня могла насчитать 3 поезда на первом пути и 2 на втором.

Задача 6. Площадь

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Городская площадь имеет размер $n \times m$ и покрыта квадратной плиткой размером 1×1. При плановой замене плитки выяснилось, что новой плитки недостаточно для покрытия всей площади, поэтому было решено покрыть плиткой только дорожку по краю площади, а в центре площади разбить прямоугольную клумбу (см. рисунок к примеру). При этом дорожка должна иметь одинаковую ширину по всем сторонам площади. Определите максимальную ширину дорожки, которую можно выложить из имеющихся плиток.

Формат входных данных

Первая и вторая строки входных данных содержат по одному числу n и m ($3 \leqslant n \leqslant 1000$, $3 \leqslant m \leqslant 1000$) — размеры площади.

Третья строка содержит количество имеющихся плиток t, $1 \leqslant t < nm$.

Формат выходных данных

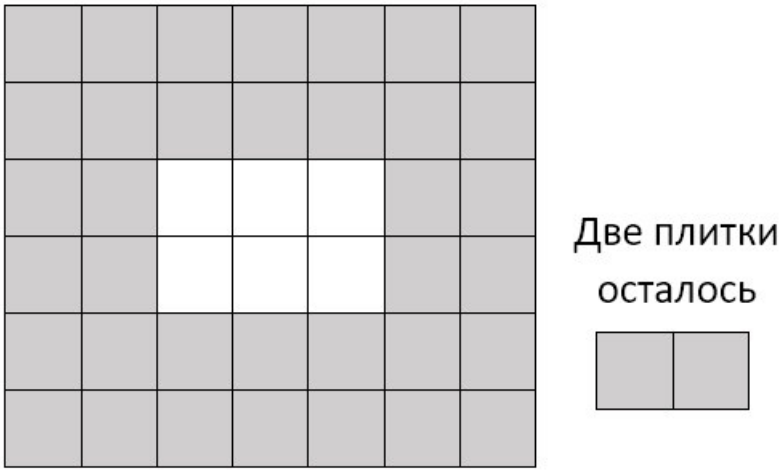
Программа должна вывести единственное число — максимальную ширину дорожки, которую можно выложить из имеющихся плиток.

Пример

стандартный ввод	стандартный вывод
6 7 38	2

Замечание

Пояснение к примеру. Площадь имеет размеры 6×7, из 38 плиток можно выложить дорожку шириной в 2 плитки.



Задача 7. Космические шахматы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

В космические шахматы играют на бесконечной доске, поэтому клетки нумеруют парой чисел (см. пример и рисунок к нему). Фигуры ходят по обычным правилам. Составьте маршрут шахматного коня из клетки $(0;0)$ в заданную клетку $(x;y)$.

Напомним, что конь за один ход перемещается на одну клетку по одной оси и на две по другой, то есть, например, из клетки $(0;0)$ он за один ход может попасть в клетки $(1;2)$, $(2;1)$, $(-1;2)$, $(2;-1)$, $(1;-2)$, $(-2;1)$, $(-1;-2)$ и $(-2;-1)$.

В качестве ответа Вам нужно вывести любой (не обязательно кратчайший) маршрут с началом в $(0;0)$ и концом в $(x;y)$, длина которого не больше 10^5 ходов.

Формат входных данных

Программа получает на вход два целых числа x и y, записанных в отдельных строках, — координаты конечной клетки маршрута коня. Клетка $(x;y)$ не совпадает с началом координат. $|x| \leqslant 100$, $|y| \leqslant 100$.

Формат выходных данных

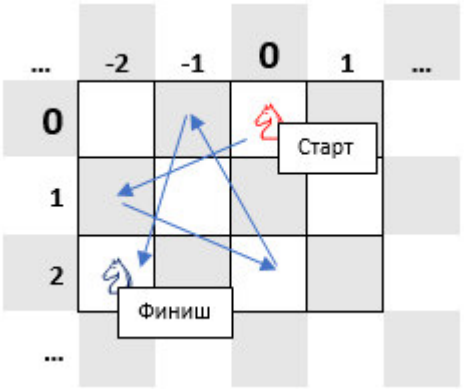
Программа должна вывести последовательность ходов, один ход в отдельной строке. В i-й строке должно быть выведено два числа x_i и y_i через пробел — координаты клетки, в которой окажется конь после i-го хода. Количество ходов не должно превышать 10^5. Последний ход должен вести в заданную клетку.

Пример

стандартный ввод	стандартный вывод
-2 2	-2 1 0 2 -1 0 -2 2

Замечание

Рисунок к примеру



Задача 1. Ограда

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Несколько столетий тому назад князь Гвидон в целях укрепления замка на острове Буян решил обнести его частоколом. Частокол - это деревянный забор с заострёнными концами.

По древнерусским государственным стандартам высота частокола должна была составлять N метров, а количество колов в частоколе должно быть не менее M штук. Также один кол в частоколе должен быть цельным, то есть нельзя взять низ от одного ствола, а верх от другого.

На острове растут только очень древние и очень высокие деревья, высотой H. Так как надо и забор строить, и древность почитать, было решено срубить минимально необходимое количество деревьев.

Князь Гвидон не силён в математике. Помогите ему посчитать, сколько деревьев ему надо приказать срубить.

Формат входных данных

В первой строке дано целое число N - высота частокола по древнерусским ГОСТам.

Во второй строке дано целое число M - количество кольев, необходимое для постройки частокола.

В третьей строке дано целое число H - высота древних деревьев острова Буян

$N \leqslant 100, M \leqslant 100, H \leqslant 100$

Формат выходных данных

Выведите одно число - количество деревьев, которые будут отданы под топор.

Примеры

стандартный ввод	стандартный вывод
10 5 25	3
3 6 9	2

Задача 2. Опасные Перекрёстки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Мэр города Снежногорск хочет сделать самые безопасные дороги в мире, для этого сначала он хочет понять, как обстоят дела с перекрестками и исправить самые опасные. Он попросил начальника строительной бригады Петра Семеновича провести обследование дорог и сообщить опасность каждого перекрестка.

Опасность каждого перекрестка определяется индексом опасности, который равен количеству пересекающихся дорог на перекрестке. Чем больше дорог пересекается на перекрестке, тем он опаснее. Все перекрестки пронумерованы от 1 до N.

Петр Семенович должен выписать все индексы опасности в упорядоченный список по номерам перекрестков и отправить его мэру. Так как мэр не очень хорошо разбирается в гродоустройстве, он будет думать, что перекресток очень опасен, если индекс его опасности больше индекса опасности соседних по списку перекрестков. Ваша задача по карте дорог города определить какие перекрестки мэр посчитает очень опасными. Если таких перекрёстков нет, выведите -1

Формат входных данных

Первая строка входных данных содержит количество перекрестков и количество дорог N и M соответственно, $1 \leqslant N \leqslant 2 \cdot 10^4$, $0 \leqslant M \leqslant 2 \cdot 10^5$. Следующие M строк содержат два числа U и V - номера перекрестков между которыми есть дорога, $1 \leqslant U, V \leqslant N$.

Формат выходных данных

Программа должна вывести через пробел номера всех перекрестков в упорядоченном по возрастанию порядке, которые мэр посчитает очень опасными.

Пример

стандартный ввод	стандартный вывод
6 5 3 4 2 5 6 5 1 2 4 5	2 5

Замечание

В тесте из условия Пётр Семёнович получит следующий список: $\{1, 2, 1, 2, 3, 1\}$ Мэр Снежногорска будет считать, что перекрёстки 2 и 5 - очень опасные

Задача 3. Странная лотерея

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Миша - обычный парень. Он любит математику, смотрит сериалы про пони, верит в чудеса, и в то, что на лотереях можно разбогатеть.

Каждую неделю Миша покупает у Аглаи Петровны газету с заветной лотереей и пять билетиков с четырёхзначным числом на каждом из них, в надежде выиграть сегодня большой куш.

В газете «Авантюрист», которую покупает Миша, каждую неделю проводится лотерейный розыгрыш. В газете публикуется какое-то число (не обязательно четырёхзначное, но не более), а в продажу поступают лотерейные билетики. Выигрывает тот лотерейный билетик, который будет удовлетворять следующим условиям:

- если друг за другом по невозрастанию записать сумму первых двух и последних двух чисел лотерейного билетика, то получается опубликованное в газете на этой неделе число.
- Число на билетике является максимальынм среди всех подобных чисел на лотерейных билетиках.

Миша так давно увлекается этим, что наладил все контакты и знает, какие билеты есть во всех лотерейных киосках в округе. Но Миша до сих пор не умеет быстро высчитывать абсолютное выигрышное значение. поэтому попросил Вас создать алгоритм, который выдавал бы номер счастливого билетика по числу в газете.

Формат входных данных

В единственной строке вводится число $0 \leqslant H < 10000$ - число, записанное в газете на этой неделе.

Формат выходных данных

Выведите единственное число - номер выигрышного билетика этой недели. Если такого не существует- выведите 0.

Пример

стандартный ввод	стандартный вывод
1412	9593

Задача 4. Пароль

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Вася, Коля и Петя — три друга, решили зарегистрировать в компьютерной игре общий аккаунт. Но вот досада, у каждого из ребят есть свой любимый пароль, а при регистрации можно указать только один. Чтобы не спорить чей пароль ставить на общем аккаунте, друзья решили разработать общий пароль, а чтобы запомнить его было проще, решили сделать новый пароль путем преобразований их любимых паролей и запомнить количество преобразований.

Существует всего две операции по преобразованию паролей:

- Заменить один символ пароля на два таких же символа (например, заменить символ «a» на «aa»).
- Заменить два подряд идущих одинаковых символа на один такой же символ.

Помогите друзьям придумать такой пароль, чтобы используя только эти две операции из него можно было получить пароль каждого из друзей. Ребята не хотят напрягаться и запоминать сложный пароль, поэтому просят сделать так, чтобы суммарное количество преобразований для каждого пароля было минимальным (от нового пароля до каждого пароля друзей).

Формат входных данных

Программа получает на вход три строки — любимые пароли Васи, Коли и Пети соответственно, состоящие из строчных букв латинского алфавита. Длина каждой строки не превышает 100 символов.

Формат выходных данных

Если при помощи указанных операций возможно придумать такой пароль, выведите такую строку S, что суммарное число операций, необходимых для преобразования нового пароля до каждого пароля друзей, будет минимальным. Если этого сделать нельзя, программа должна вывести одно слово IMPOSSIBLE (заглавными буквами).

Пример

стандартный ввод	стандартный вывод
bbuuuzzz buuzzzz bbbuzz	bbuuzzz
wabba awbba bwaaba	IMPOSSIBLE

Задача 5. Зоопарк Глеба

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Недавно Глеб открыл свой зоопарк. По лучшим мировым традициям он имеет форму круга, впрочем это не важно. Важно то, что он взял вас туда начальником охраны. Казалось бы все началось так хорошо, но именно в вашу первую смену кто-то открыл все клетки и животные разбежались по всему зоопарку. Перед вами встала задача поймать всех животных в ловушки, чтобы потом вернуть каждого в свою клетку.

В зоопарке n животных различных животных одного из 26 видов. Каждый вид обозначается своей буквой от 'a' до 'z'. Под каждый из них есть свой тип ловушки. Ловушки обозначаются латинскими заглавными буквами. К сожалению, почти все животные враждуют между собой в природе, поэтому ни одно животное не станет переходить дорогу животному своего или другого вида из-за инстинкта самосохранения.

Зоопарк по периметру обнесен колючей проволкой, поэтому животные не могут ходить вдоль забора.

С помощью камер, удалось выяснить, где находятся все животные. Умная система поддержки жизнедеятельности зоопарка уже просканировала зоопарк и вывела типы всех животных и ловушек в том порядке, в котором они видны из центра зоопарка против часовой стрелки. Получилось так, что все животные и все ловушки находятся около забора, то есть можно считать, что путь любого животного начинается в одной из точек окружности и заканчивается в точке, где находится ловушка для животных этого вида - тоже точка на окружности.

Вы хотите понять, могут ли животные придти в свою ловушку так, чтобы их путь не пересекался ни с одним другим. Если да, выведите какую-нибудь из схем поимки животных.

Формат входных данных

На вход подается строчка из $2 \cdot n, (n \leqslant 10000)$ символов латинского алфавита, где маленькая буква - животное, а большая - ловушка.

Гарантируется, что ловушек каждого типа столько же, сколько и представителей данного вида животных в зоопарке.

Формат выходных данных

Требуется вывести "Impossible если решения не существует или "Possible если можно загнать всех животных в свои ловушки так, чтобы их пути не пересекались.

Если это возможно, то для каждой ловушки в порядке обхода требуется вывести индекс животного, которое будет поймано в ней. Индексом животного называется его порядковый номер среди животных в общем списке животных и ловушек.

Примеры

стандартный ввод	стандартный вывод
ABba	Possible 2 1
aBAb	Impossible

Замечание

Первый пример: Животное b идёт в ловушку B, а животное a ловится в ловушку A. Их пути не пересекаются, поэтому их возможно поймать.

Второй пример: Пути животных в любом случае пересекаются, поэтому поймать их невозможно.