

Задача А. Интернет-банкинг

Имя входного файла: `bank.in`
Имя выходного файла: `bank.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Интернет-банкинг — это современная технология, которая позволяет клиентам банка получать доступ к информации о своих счетах с помощью сети Интернет из практически любой точки земного шара. Разумеется, при использовании интернет-банкинга большую роль играют вопросы безопасности. Поэтому для доступа к системе Интернет-банкинга пользователю необходимо ввести пароль.

Система Интернет-банкинга *Bank 2.0*, используемая одним *Очень Крупным Банком*, использует следующий способ ввода пароля. Серверной частью системы случайно генерируются n строк s_1, \dots, s_n , каждая из которых состоит из m строчных букв латинского алфавита (предполагается, что пароли состоят только из таких букв).

При вводе пароля пользователю разрешается выполнять такую операцию: выбрать из данных строк две (обозначим их как s_i и s_j , $1 \leq i, j \leq n$, $i \neq j$) и некоторую позицию k ($1 \leq k \leq m$) в них, после чего поменять местами k -е символы в s_i и s_j . Например, если $s_i = \text{«abcde»}$, $s_j = \text{«vwxyz»}$, $k = 3$, то после выполнения этой операции будут верны следующие равенства: $s_i = \text{«abxde»}$ и $s_j = \text{«vwcyz»}$. Для ввода пароля пользователю необходимо за *минимальное* число таких операций добиться состояния, в котором хотя бы одна из строк s_1, \dots, s_n совпадает с p .

Ваша задача состоит в том, чтобы написать программу, которая по заданному набору строк s_1, \dots, s_n и паролю пользователя p определит минимальное число операций указанного типа, которые необходимо выполнить для ввода пароля, а также найдет способ ввода пароля за такое число операций.

Формат входного файла

Первая строка входного файла содержит целое число n ($2 \leq n \leq 100$). Каждая из последующих n строк содержит строки s_1, \dots, s_n . Все они состоят только из строчных букв латинского алфавита и имеют одинаковую длину m ($2 \leq m \leq 100$).

Последняя строка входного файла содержит пароль пользователя p . Его длина равна m , и он состоит только из строчных букв латинского алфавита.

Формат выходного файла

Первая строка выходного файла должна содержать минимальное число c операций, необходимых для ввода пароля. Если с помощью описанных в условии операций пароль ввести нельзя, то выведите в первой строке «-1».

В случае существования решения следующие c строк должны содержать описания операций. Операции должны быть перечислены в порядке их применения, каждая из строк должна содержать три целых числа: i , j и k ($1 \leq i, j \leq n$, $i \neq j$, $1 \leq k \leq m$). Эти числа означают, что соответствующая операция состоит в обмене k -ых символов строк s_i и s_j .

Примеры

<code>bank.in</code>	<code>bank.out</code>
3 abc cab bca acb	2 1 3 2 1 2 3
3 abc cab bca acd	-1

Задача В. Большой огромный коллайдер

Имя входного файла: `collider.in`
Имя выходного файла: `collider.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Приехав в Хоббитанию, белый маг Гэндальф принялся рассказывать Бильбо последние новости из Средиземья. Больше всего впечатлительного хоббита поразил рассказ о *Большом огромном коллайдере* — только представить себе гигантских размеров кольцо, зарытое под землей!

Вдохновленный рассказом мага, Бильбо решил смастерить собственный коллайдер прямо у себя в норе. До начала строительства нора представляет собой множество комнат, причем некоторые пары комнат соединены коридорами. Как это принято у хоббитов, из любой комнаты в любую другую можно добраться по коридорам ровно одним способом.

Бильбо хочет прокопать новые коридоры в норе, но так как копать будут только Фродо и сам Бильбо, есть возможность прокопать только один или два новых коридора.

После этого последовательность из нескольких комнат будет названа коллайдером. При этом должны быть выполнены следующие условия: первая комната в этой последовательности соединена коридором со второй, вторая — с третьей, и так далее, наконец, последняя комната последовательности должна быть соединена с первой. Кроме того, каждая комната может входить в эту последовательность не более одного раза.

Помогите Бильбо выбрать такие один или два новых коридора, чтобы коллайдер имел максимальную возможную длину, то есть состоял из максимального возможного числа комнат.

Формат входного файла

В первой строке входного файла содержится целое число n ($3 \leq n \leq 100\,000$) — число комнат в норе Бильбо.

В следующих $n - 1$ строках содержатся по два целых числа — номера комнат, соединенных коридорами. Комнаты нумеруются от 1 до n .

Формат выходного файла

В первую строку выходного файла выведите максимально возможное число комнат в коллайдере.

На следующих одной или двух строках выведите пары номеров комнат, между которыми следует прокопать новые коридоры.

Если существует несколько возможных планов строительства коллайдера максимальной длины, выведите любой из них.

Примеры

<code>collider.in</code>	<code>collider.out</code>
4 1 2 2 3 3 4	4 1 4
4 1 2 1 3 1 4	4 2 3 2 4

В первом примере коллайдер состоит из комнат с номерами 1, 2, 3 и 4 (именно в этом порядке), во втором примере — 1, 3, 2, 4.

Задача С. Фитнесс-клуб

Имя входного файла: `fitness.in`
Имя выходного файла: `fitness.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В последнее время фитнес-клубы стали очень популярны среди жителей столицы Флатландии. В такие клубы люди ходят после работы для того, чтобы поддерживать себя в хорошей физической форме. В фитнес-клубе «Флат» каждому из посетителей на время посещения выделяется один из k шкафчиков, в который он может убрать свои вещи на время тренировки.

В течение дня в фитнес-клубе проходит n тренировок, причем каждый из посетителей приходит к началу какой-либо из этих тренировок (в этот момент он получает ключ от шкафчика) и уходит сразу после ее окончания (в этот момент он сдает ключ от шкафчика). При этом можно считать, что все посетители, закончившие тренировку, уходят раньше, чем все посетители, пришедшие на следующую тренировку.

Некоторые из посетителей уходя закрывают шкафчик, а другие — не закрывают. Так как все посетители фитнес-клуба посещают его уже достаточно давно, то про каждого из них персонал клуба знает, закроет ли он свой шкафчик, когда будет уходить. Таким образом, для каждой тренировки известно два числа: число a_i посетителей, которые закроют шкафчик, и число b_i посетителей, которые не закроют.

В начале дня все k шкафчиков закрыты. Разумеется, персоналу клуба хочется, чтобы в конце дня также как можно больше шкафчиков были закрыты — тогда надо будет меньше работать при подготовке к очередному дню. Для того, чтобы достичь этой цели, персонал может выдавать ключи от шкафчиков посетителям произвольным образом. Например, имеет смысл выдавать ключ от открытого шкафчика тому, кто его точно закроет.

Необходимо найти максимальное число шкафчиков, которые окажутся закрытыми, в случае оптимальных действий персонала фитнес-клуба.

Формат входного файла

Первая строка входного файла содержит два целых числа: n ($1 \leq n \leq 100$) и k ($1 \leq k \leq 1000$). Каждая из последующих n строк содержит по два целых числа a_i и b_i ($0 \leq a_i, b_i \leq k$, $a_i + b_i \leq k$).

Формат выходного файла

В выходной файл выведите одно число — ответ на задачу.

Примеры

<code>fitness.in</code>	<code>fitness.out</code>
2 4 1 2 1 1	3

Пронумеруем шкафчики числами от 1 до 4. Посетителям, пришедшим на первую тренировку выдадим ключи так: тому, кто закроет, — от шкафчика 1, тем, кто не закроет, — от 2 и 3. Посетителям, пришедшим на вторую тренировку, выдадим ключи так: тому, кто закроет, — номер 2, тому, кто не закроет, — номер 3. В итоге открытым после окончания дня останется только шкафчик номер 3.

Задача D. Электрик-ковбой Джо

Имя входного файла: joe.in
Имя выходного файла: joe.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

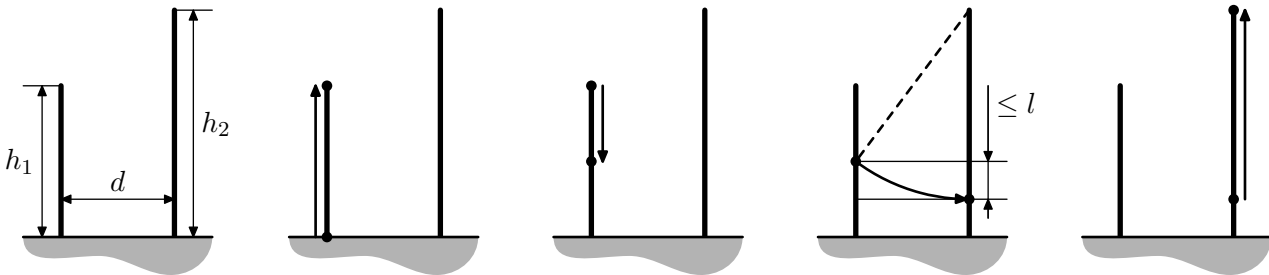
Джо — электрик-ковбой. Как у всех ковбоев у него есть лассо, как всем электрикам ему иногда приходится залезать на столбы, и как все он ленив.

Вот и сейчас ему поручили проверить два стоящих на расстоянии d друг от друга столба высоты h_1 и h_2 соответственно. Чтобы убедиться, что все хорошо, Джо должен побывать на вершинах обоих столбов.

Электрик-ковбой посещает столбы следующим образом: сначала он выбирает один из столбов и просто взбирается на него. Выполнив все работы на вершине, он спускается по этому столбу на некоторую высоту (возможно до самой земли), достает свое лассо и цепляется им за некоторую точку второго столба (это может быть произвольная точка). После этого Джо прыгает и движется вниз по дуге окружности с центром в точке, за которую зацепилось лассо, пока не достигнет либо другого столба, либо земли.

При этом если от начальной позиции электрика до конца его полета высота изменяется более чем на l , то ковбой набирает слишком большую скорость, больно ударяется и попадает в больницу, так и не выполнив работу. Поэтому Джо всегда аккуратно выбирает параметры прыжка.

Если в результате прыжка Джо оказался на земле, он подходит к другому столбу и взбирается на него. Если же Джо оказался на столбе, то он взбирается на вершину из той точки, в которой он оказался.



Джо просит вас помочь ему выполнить работу, сообщив какое минимальное расстояние ему придется лезть вверх по столбам.

Формат входного файла

Входной файл содержит четыре положительных целых числа: d , h_1 , h_2 и l — расстояние между столбами, высоту первого и второго столбов и максимальный допустимый перепад высот при прыжке, соответственно. Все числа во входном файле не превышают 10^6 .

Формат выходного файла

Выведите ответ с максимальной возможной точностью. Ответ будет проверяться с точностью до 10^{-5} .

Примеры

joe.in	joe.out
5 5 5 5	10.00000
4 5 8 5	10.00000
4 8 5 1	13.00000
3 4 6 1	9.00000

Задача Е. Счастливый билетик

Имя входного файла: `lucky.in`
Имя выходного файла: `lucky.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В стране Лаккиландии очень развит общественный транспорт. Проезд в нем бесплатный, но при этом каждому пассажиру при входе выдают билетик с уникальным номером. Особенно ценятся так называемые *счастливые билетики*.

Билетик называется *счастливым*, если сумма цифр на четных позициях в его номере равна сумме цифр на нечетных позициях.

Ване известно, что билеты выдаются подряд в порядке возрастания номеров. В очередной раз войдя в автобус Ваня получил свой очередной билет и тут ему стало интересно, какой существует минимальный счастливый билетик с номером, большим номера Ваниного билетика. Помогите Ване узнать ответ на этот вопрос.

Формат входного файла

В единственной строке входного файла задан номер Ваниного билетика — натуральное число, имеющее в своей десятичной записи не более 100 цифр.

Формат выходного файла

В выходной файл выведите минимальный номер счастливого билетика, который больше номера Ваниного билетика.

Примеры

<code>lucky.in</code>	<code>lucky.out</code>
123123	123134
99	110

Задача F. Циклические палиндромы

Имя входного файла: `palin.in`
Имя выходного файла: `palin.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Летит сорок свинок — как кони — в скорости тел!

Палиндромом называется слово, которое читается одинаково в обоих направлениях. Например, слово «шалаш» является палиндромом.

Для слова можно определить *циклический сдвиг* следующим образом: часть букв из конца слова (возможно ни одной) переставляются в начало с сохранением порядка относительно друг друга. Например, слово «нора» является циклическим сдвигом слова «рано» (надо перенести в начало буквы «но»).

Будем называть слово *циклическим палиндромом*, если у него есть циклический сдвиг, который является палиндромом. Например, слово «масса» является циклическим палиндромом: его циклический сдвиг «самас» является палиндромом.

Вам задано слово, состоящее не более чем из 100 букв латинского алфавита. Требуется проверить, является ли это слово циклическим палиндромом.

Формат входного файла

Входной файл содержит одно слово, содержащее от 1 до 100 строчных букв латинского алфавита.

Формат выходного файла

Если входной файл содержит циклический палиндром, выведите в выходной файл слово «yes». В противном случае выведите «no».

Примеры

<code>palin.in</code>	<code>palin.out</code>
<code>array</code>	<code>yes</code>
<code>computer</code>	<code>no</code>
<code>sis</code>	<code>yes</code>

Задача G. Строка

Имя входного файла: `string.in`
Имя выходного файла: `string.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В математике часто встречаются так называемые *рекуррентные соотношения*. Обычно они применяются для задания числовых последовательностей — очередное число в последовательности некоторым образом выражается через предыдущие. Примером такой последовательности являются числа Фибоначчи (в них очередное число равно сумме двух предыдущих).

С помощью соотношений такого типа можно задавать не только последовательности чисел, но и последовательности строк. В этой задаче рассматривается последовательность строк s_0, s_1, \dots , задаваемая следующим образом.

Строка s_0 пуста, а каждая строка s_i ($i \geq 1$) получается из s_{i-1} по следующему правилу: если десятичная запись числа i входит как подстрока в s_{i-1} , то $s_i = s_{i-1}$, иначе s_i получается приписыванием к s_{i-1} в конец десятичной записи числа i .

Задано число n . Необходимо найти строку s_n .

Формат входного файла

Входной файл содержит целое число n ($1 \leq n \leq 500$).

Формат выходного файла

В выходной файл выведите строку s_n .

Примеры

<code>string.in</code>	<code>string.out</code>
1	1
3	123
13	123456789101113

Задача Н. Театр

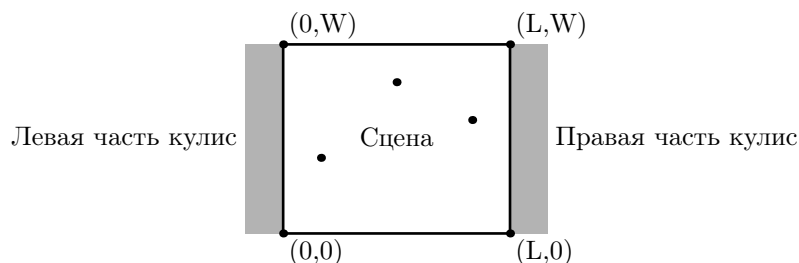
Имя входного файла: `theater.in`
Имя выходного файла: `theater.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Олег и Сергей — мастера по свету в одном из театров. В их задачу входит управление подсветкой сцены во время спектакля. Спектакль состоит из действий, во время каждого из которых некоторые лампы подсветки должны быть включены, а некоторые выключены. В перерывах между действиями занавес закрывается, и Олег с Сергеем должны включить на сцене набор ламп, необходимый для следующего действия.

Чтобы ничего не перепутать, мастера договорились, что Олег будет только включать лампы, а Сергей только выключать.

Театральная сцена представляет собой прямоугольник W на L метров, внутри которого расположено N ламп подсветки.

Кулисы состоят из двух не связанных между собой частей — левой и правой. Левая часть кулис целиком прилегает к левой стороне сцены, а правая — целиком к правой.



Олег может перемещаться по сцене с максимальной скоростью V_1 метров в секунду, а Сергей — V_2 метров в секунду. Мастера могут находиться на сцене только в перерывах между действиями. Во время действия они могут переместиться в любую точку в пределах той части кулис, в которой они оказались перед началом действия.

Перед началом спектакля Олег и Сергей получили подробный сценарий, в котором указано количество действий M и для каждого действия свой набор ламп подсветки, которые должны быть включены. Лампы, которые не входят в этот набор, должны быть выключены. Перед первым действием Олег должен находиться в левой части кулис, а Сергей — в правой. Изначально включены лампы, необходимые для первого действия.

Задача Олега и Сергея — организовать работу так, чтобы суммарное время всех перерывов между действиями было бы минимальным.

Формат входного файла

На первой строке входного файла находится пять чисел — W, L, V_1, V_2 и N ($1 \leq W, L \leq 50$, $1 \leq V_1, V_2 \leq 20$, $1 \leq N \leq 15$) — размеры сцены, максимальные скорости мастеров и число ламп подсветки соответственно. Далее идут N строк с координатами ламп подсветки в метрах x_i, y_i ($0 < x_i < L$, $0 < y_i < W$). Следующая строка содержит число M ($1 \leq M \leq 10\,000$) — число действий в спектакле. Далее идут M строк, каждая из которых содержит число ламп подсветки, которые должны быть включены в соответствующем действии, и номера ламп подсветки. Все числа во входном файле целые.

Формат выходного файла

В выходной файл выведите единственное число — минимальное суммарное время перерывов между действиями в секундах с точностью 10^{-5} .

Примеры

theater.in	theater.out
5 6 1 1 3 1 2 3 4 5 3 1 1 3	0.000000
5 6 1 1 3 1 2 3 4 5 3 3 1 3 2 1 2 3 1 2 3	8.828427

В первом примере только одно действие, поэтому суммарное время перерывов равно нулю.

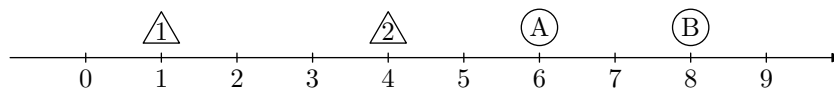
Задача I. Провода

Имя входного файла: `wires.in`
Имя выходного файла: `wires.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В 2222 году экспедиция на планету XZ-238 в звездной системе ню Малой Медведицы обнаружила в одной из пещер загадочное устройство. Вдоль стены по прямой расположены n треугольных и n круглых гнезд. Расшифровка инструкций, расположенных рядом на стене, показала, что эти гнезда следует соединить проводами: каждое треугольное гнездо должно быть соединено с каким-либо круглым, расположенным правее.

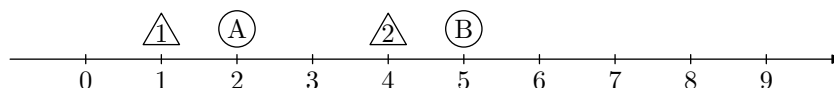
Были изготовлены n треугольных и n круглых штекеров, которые можно соединять проводами. Однако никакой маркировки нет, поэтому понять какое гнездо следует соединить с каким не удастся. Начальник экспедиции хочет заказать с Земли провода. Решено было заказать один большой моток кабеля, который затем нарезать на соединительные провода. Поскольку доставка грузов с Земли дорога, необходимо заказать как можно меньше кабеля. С другой стороны, поскольку неизвестно какое гнездо требуется соединить с каким, нужно заказать такое количество кабеля, чтобы при любом способе соединения длины кабеля хватило на изготовление соединительных проводов. Помогите руководству экспедиции по заданному расположению гнезд вычислить, кабель какой длины следует заказать. Гарантируется, что хотя бы один способ соединить гнезда описанным образом существует.

Например, рассмотрим расположение гнезд, изображенное на рисунке.



Можно соединить треугольное гнездо 1 с круглым гнездом А, а треугольное гнездо 2 — с круглым гнездом В. Для такого соединения необходимы провода длины 5 и 4, суммарная длина — 9. Можно также соединить 1 с В и 2 с А, в этом случае необходимы провода длины 7 и 2, суммарная длина также 9. Таким образом необходимо заказать с Земли кабель длиной 9 метров.

В случае же, показанном на следующем рисунке, есть лишь один способ соединить гнезда: 1 с А, а 2 с В. Гнездо 2 нельзя соединить с А, поскольку А расположено левее. Так что в этом случае необходимо 2 метра кабеля.



Формат входного файла

Первая строка входного файла содержит число n ($1 \leq n \leq 1000$). Вторая строка содержит n целых чисел, i -е число означает расстояние от начала стены до i -го треугольного гнезда. Третья строка также содержит n целых чисел, i -е число означает расстояние от начала стены до i -го круглого гнезда. Все числа во второй и третьей строках различны и лежат в диапазоне от 0 до 10^5 .

Формат выходного файла

Выведите одно число — кабель какой длины необходимо заказать с Земли.

Примеры

wires.in	wires.out
2 1 4 6 8	9
2 1 4 2 5	2