

Задача А. Интернет-банкинг

Имя входного файла: `bank.in`
Имя выходного файла: `bank.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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 s 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 s 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

bank.in	bank.out
3 abc cab bca acb	2 1 3 2 1 2 3
3 abc cab bca acd	-1

Задача В. Рисование

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дано N точек на плоскости, их надо покрасить в черный и белый цвета.

При этом должны присутствовать точки обоих цветов и должна существовать прямая, по одну сторону от которой все точки черные, по другую белые.

Требуется посчитать количество раскрасок, удовлетворяющих этим условиям.

Формат входного файла

В первой строке задано число $2 \leq N \leq 300$. В следующих N строках заданы координаты точек.

Формат выходного файла

Выведите единственное число — количество различных раскрасок.

Примеры

input.txt	output.txt
4 0 0 1 0 1 1 0 1	12

Задача С. Жизнь бактерий

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На плоскости живут N бактерий, они находятся в точках с целочисленными координатами. Каждый день бактерии размножаются «делением пополам»: берутся все возможные пары бактерий и на середине отрезка их соединяющего рождается новая бактерия. Все старые бактерии при этом умирают. Определить первый день, когда найдутся две бактерии, которые родятся в одном и том же месте или сообщите, что этого никогда не произойдет.

Формат входного файла

Первая строка входных данных содержит число бактерий N ($1 \leq N \leq 1000$). Каждая из следующих N строк содержит 2 целых числа x_i, y_i – координаты i -й бактерии ($-10^9 \leq x_i, y_i \leq 10^9$). Никакие 2 бактерии не располагаются в одной точке.

Формат выходного файла

Выведите 0, если 2 бактерии никогда не родятся в одном месте. В противном случае выведите минимальное количество дней, через которое это случится.

Примеры

input.txt	output.txt
3 0 0 1 0 0 1	0
4 0 0 1 0 0 1 1 1	1
4 1 1 5 3 7 4 9 5	2

Задача D. Электрик-ковбой Джо

Имя входного файла: `joe.in`
Имя выходного файла: `joe.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

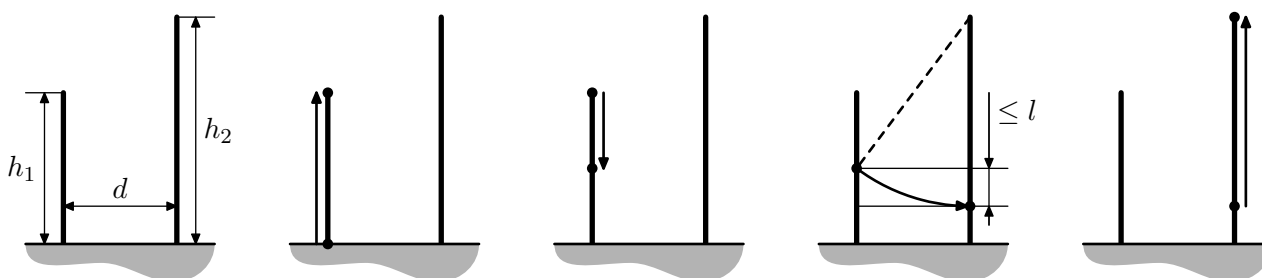
Джо — электрик-ковбой. Как у всех ковбоев у него есть лассо, как всем электрикам ему иногда приходится залезать на столбы, и как все он ленив.

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: 64 мегабайта

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. Дерево Фенвика

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Дерево Фенвика — это структура данных, эффективно поддерживающая запросы о сумме префикса числового массива. Для числа t обозначим $h(t)$ максимальное значение k , такое что t делится на 2^k . Например, $h(24) = 3$, $h(5) = 0$. Обозначим $l(t) = 2^{h(t)}$, например, $l(24) = 8$, $l(5) = 1$.

Рассмотрим массив $a[1], a[2], \dots, a[n]$ целых чисел. Дерево Фенвика для этого массива — это массив $b[1], b[2], \dots, b[n]$, такой что

$$\sum_{j=i-l(i)+1}^i a[j]$$

Таким образом: $b[1] = a[1]$,

$b[2] = a[1] + a[2]$,

$b[3] = a[3]$,

$b[4] = a[1] + a[2] + a[3] + a[4]$,

$b[5] = a[5]$,

$b[6] = a[5] + a[6]$,

Например, дерево Фенвика для массива $a = (3, -1, 4, 1, -5, 9)$ есть массив $b = (3, 2, 4, 7, -5, 4)$.

Назовем массив само-фенвиковским, если он совпадает со своим деревом Фенвика. Например, массив $a = (0, -1, 1, 1, 0, 9)$ таковым является.

Вам дан массив . Вам разрешается заменять в нем некоторые элементы, не меняя их порядка, чтобы сделать из исходного массива само-фенвиковский. Количество изменений при этом должно быть минимально возможным.

Формат входного файла

В первой строке входных данных содержится количество чисел в массиве n ($1 \leq n \leq 100\,000$). Во второй строке находятся сами n целых чисел. Все числа по модулю не превосходят 10^9 .

Формат выходного файла

Выведите n чисел — элементы видоизмененного массива. Если решений несколько — выведите любое из них.

Примеры

input.txt	output.txt
6	0 -1 1 1 0 9
3 -1 4 1 -5 9	

Задача G. Контроль доступа

Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мегабайта

Вася разрабатывает новый веб-сервер. В настоящее время он работает над функцией, обеспечивающей поддержку списков контроля доступа. Список контроля доступа позволяет ограничить доступ к некоторым ресурсам веб-сайта, основывая на основании IP-адреса запрашивающей стороны.

Каждый IP-адрес — это 4-байтный номер, который записан байт за байтом в десятичной записи. Байты разделены точками следующим образом: «byte0.byte1.byte2.byte3» (кавычки добавлены для ясности). Каждый байт записывается как десятичное число от 0 до 255 (включительно) без ведущих нулей. IP-адреса организованы в IP-сети. IP сети описывается двумя 4-байтовыми числами - сетевым адресом и маской сети. И сетевой адрес и маска сети записаны в той же форме, что и IP-адреса. Для того чтобы понять смысл сетевого адреса и маски сети рассмотрим их двоичное представление. Двоичное представление IP адреса, сетевого адреса и маски сети состоит из 32 бит: 8 бит для byte0 (от старших к младшим), затем по 8 бит для byte1, 8 бит для byte2 и 8 бит для byte3.

IP сеть содержит $2N$ IP-адресов, где $0 \leq N \leq 32$. В маске сети первые $32-N$ бита установлены в единицы, и последние N бит установлены в ноль. Сетевой адрес имеет произвольные $32-N$ первых бит, а последние N бит установлен в ноль. IP сеть содержит все IP-адреса, первые $32-N$ бит которых равны $32-N$ первых бит сетевого адреса с произвольными N последними битами. Например, IP сеть с сетевым адресом 194.85.160.176 и сетевая маска 255.255.255.248 содержит 8 IP-адресов, с 194.85.160.176 по 194.85.160.183 (включительно).

IP сети, как правило, обозначается как «byte0.byte1.byte2.byte3/S», где «byte0.byte1.byte2.byte3» — сетевой адрес, S — это число бит, установленных в единицу в маске сети. Например, IP сети из предыдущего абзаца обозначается как 194.85.160.176/29. Список контроля доступа содержит упорядоченный список правил. Каждое правило имеет одну из следующих форм:

deny from <IP network> — запрещает доступ к ресурсу для любого IP из заданной сети.

deny from <IP address> — запрещает доступ к ресурсу для указанного IP-адреса.

allow from <IP network> — разрешает доступ к ресурсу для любого IP из заданной сети.

allow from <IP address> — разрешает доступ к ресурсу для указанного IP-адреса.

Когда кто-нибудь обращается к какому-либо ресурсу, первым делом проверяется IP-адрес обращающегося по списку контроля доступа. Правила проверяются в том порядке, в котором они перечислены, и применяется первое выполняющееся правило. Если ни одно из правил не соответствует IP-адресу запрашивающей стороны, то доступ предоставляется.

По известному списку контроля доступа и списку запросов от IP-адресов, необходимо выяснить для каждого из запросов будет ли предоставлен доступ к ресурсу.

Формат входного файла

Первая строка ввода содержит число N — количество правил в списке контроля доступа ($0 \leq N \leq 100000$). Следующие N строк содержат правила по одному в строке. IP сеть всегда записывается как «byte0.byte1.byte2.byte3/S». Следующая строка содержит число M — количество IP адресов, которые следует проверить ($0 \leq M \leq 100000$). Следующие M строк содержат по одному IP адресу в строке.

Формат выходного файла

Для каждого из M IP-адресов выведите «A», если доступ будет предоставлен, и «D» иначе. Все символы следует выводить слитно, не разделяя пробелами.

Примеры

input.txt	output.txt
4 allow from 10.0.0.1 deny from 10.0.0.0/8 allow from 192.168.0.0/16 deny from 192.168.0.1 5 10.0.0.1 10.0.0.2 194.85.160.133 192.168.0.1 192.168.0.2	ADAAA

Задача Н. Театр

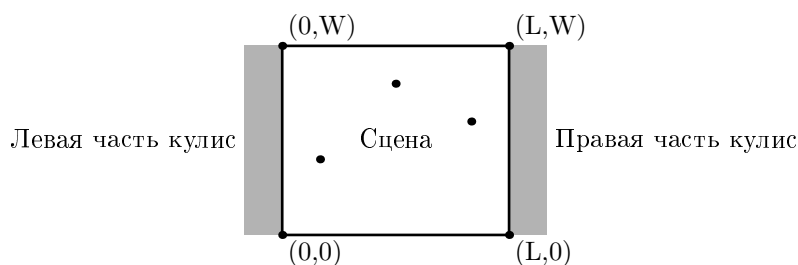
Имя входного файла: `theater.in`
Имя выходного файла: `theater.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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.

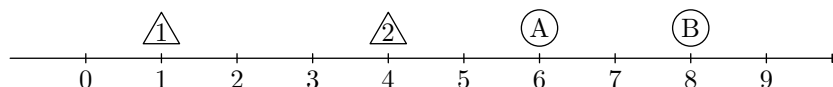
Задача I. Провода

Имя входного файла: `wires.in`
Имя выходного файла: `wires.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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.

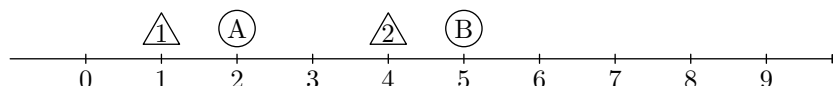
Были изготовлены n треугольных и n круглых штекеров, которые можно соединять проводами. Однако никакой маркировки нет, поэтому понять какое гнездо следует соединить с каким не удастся. Начальник экспедиции хочет заказать с Земли провода. Решено было заказать один большой моток кабеля, который затем нарезать 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 A, а треугольное гнездо 2 — с круглым гнездом B. Для такого соединения необходимы провода длины 5 и 4, суммарная длина — 9. Можно также соединить 1 с B и 2 с A, 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 A, а 2 с B. Гнездо 2 нельзя соединить с A, поскольку A 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

Задача J. Преобразование матрицы

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Введем следующие операции над прямоугольной таблицей символов. Пусть нам дана матрица, состоящая из m строк (первый индекс) и n столбцов (второй индекс). Определим результирующую матрицу для каждой из операций следующим образом:

- Транспозиция относительно главной диагонали (код операции '1'): $B_{j,i} = A_{i,j}$
- Транспозиция относительно другой диагонали (код операции '2'): $B_{n-j+1,m-i+1} = A_{i,j}$
- Горизонтальное отражение (код операции 'H'): $B_{m-i+1,j} = A_{i,j}$
- Горизонтальное отражение (код операции 'V'): $B_{i,n-j+1} = A_{i,j}$
- Поворот на 90 ('A'), 180 ('B'), или 270 ('C') градусов по часовой стрелке: $B_{j,m-i+1} = A_{i,j}$ (для 90°)
- Поворот на 90 ('X'), 180 ('Y'), или 270 ('Z') градусов против часовой стрелки: $B_{n-j+1,i} = A_{i,j}$ (для 90°)

Вам дана последовательность не более 100000 операций над исходной матрицей. Выведите полученную в результате этих операций матрицу.

Формат входного файла

В первой строке входных данных находятся значения m и n ($0 < m, n \leq 300$). В каждой из следующих m строк находятся по n печатных символов (т.е. не используются символы с кодами от 33 до 126).

Последняя строка содержит описание операций, которые были выполнены над этой матрицей, путем записи без пробелов из кодов. Операции выполняются по порядку слева направо.

Формат выходного файла

Выведите сначала два целых числа — число строк и столбцов в результирующей матрице. Затем выведите саму матрицу в том же формате, что и во входных данных.

Примеры

input.txt	output.txt
3 4 0000 a0b0 cdef A1	3 4 cdef a0b0 0000