

Задача А. Любители Кошек

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В университетском клубе любителей кошек зарегистрировано n членов. Естественно, что некоторые из членов клуба знакомы друг с другом. Нужно сосчитать, сколькими способами можно выбрать из них троих, которые могли бы свободно общаться (то есть, любые два из которых знакомы между собой).

Формат входных данных

В первой строке ввода заданы числа n и m ($1 \leq n \leq 1000$, $1 \leq m \leq 30\,000$), где m обозначает общее число знакомств. В последующих m строках идут пары чисел a_i b_i , обозначающие, что a_i знаком с b_i . Информация об одном знакомстве может быть записана несколько раз, причём даже в разном порядке (как (x, y) , так и (y, x)).

Формат выходных данных

Выведите одно число — количество способов выбрать троих попарно знакомых друг с другом людей из клуба.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 3 1 2 2 3 3 1	1

Задача В. Волновой обход графа

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Пусть *расстояние* от вершины u до вершины v — это минимальное количество рёбер в пути между u и v ; так, расстояние между u и u — 0, а расстояние между любыми двумя различными соседними вершинами — 1.

Волновым обходом графа из вершины v назовём такую последовательность вершин u_1, u_2, \dots, u_r , что:

- $u_1 = v$,
- Каждая вершина графа, достижимая из v , встречается в ней хотя бы один раз, и
- Каждая следующая вершина последовательности удалена от вершины v не меньше, чем предыдущая.

Задан связный неориентированный граф и его вершина v . Выведите любой волновой обход этого графа.

Формат входных данных

В первой строке заданы числа N , M и v через пробел — количество вершин и рёбер в графе и начальная вершина обхода ($1 \leq N \leq 100$, $0 \leq M \leq 10\,000$, $1 \leq v \leq N$). Следующие M строк содержат по два числа u_i и v_i через пробел ($1 \leq u_i, v_i \leq N$); каждая такая строка означает, что в графе существует ребро между вершинами u_i и v_i .

Формат выходных данных

В первой строке выведите число r — количество вершин в найденном волновом обходе ($1 \leq r \leq 10\,000$; гарантируется, что обход, удовлетворяющий этим ограничениям, существует). Во второй строке выведите сами числа u_1, u_2, \dots, u_r через пробел.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2 1 1 2 2 3	3 1 2 3
4 4 1 1 2 2 3 3 4 4 1	4 1 2 4 3

Задача С. Пошаговый обход графа

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Пошаговым обходом графа из вершины v назовём такую последовательность вершин u_1, u_2, \dots, u_r , что:

- $u_1 = u_r = v$,
- Каждая вершина графа, достижимая из v , встречается в ней хотя бы один раз, и
- Между любыми двумя соседними вершинами последовательности в графе существует ребро.

Задан связный неориентированный граф и его вершина v . Выведите любой пошаговый обход этого графа.

Формат входных данных

В первой строке заданы числа N , M и v через пробел — количество вершин и рёбер в графе и начальная вершина обхода ($1 \leq N \leq 100$, $0 \leq M \leq 10\,000$, $1 \leq v \leq N$). Следующие M строк содержат по два числа u_i и v_i через пробел ($1 \leq u_i, v_i \leq N$); каждая такая строка означает, что в графе существует ребро между вершинами u_i и v_i .

Формат выходных данных

В первой строке выведите число r — количество вершин в найденном пошаговом обходе ($1 \leq r \leq 10\,000$; гарантируется, что обход, удовлетворяющий этим ограничениям, существует). Во второй строке выведите сами числа u_1, u_2, \dots, u_r через пробел.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2 1 1 2 2 3	5 1 2 3 2 1
4 4 1 1 2 2 3 3 4 4 1	5 1 2 3 4 1

Задача D. Связность

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В этой задаче требуется проверить, что граф является *связным*, то есть что из любой вершины можно по рёбрам этого графа попасть в любую другую.

Формат входных данных

В первой строке заданы числа N и M через пробел — количество вершин и рёбер в графе, соответственно ($1 \leq N \leq 100$, $0 \leq M \leq 10\,000$). Следующие M строк содержат по два числа u_i и v_i через пробел ($1 \leq u_i, v_i \leq N$); каждая такая строка означает, что в графе существует ребро между вершинами u_i и v_i .

Формат выходных данных

Выведите «YES», если граф является связным, и «NO» в противном случае.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2 1 2 3 2	YES
3 1 1 3	NO

Задача Е. Дерево

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф. Проверьте, является ли он деревом.

Формат входных данных

В первой строке входных данных заданы через пробел два целых числа n и m — количество вершин и рёбер в графе, соответственно ($1 \leq n \leq 100$). В следующих m строках заданы рёбра; i -я из этих строк содержит два целых числа u_i и v_i через пробел — номера концов i -го ребра ($1 \leq u_i, v_i \leq n$). Граф не содержит петель и кратных рёбер.

Формат выходных данных

В первой строке выведите «YES», если граф является деревом, и «NO» в противном случае.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2 1 2 1 3	YES
3 3 1 2 2 3 3 1	NO

Задача F. Расстояние от корня

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В заданном корневом дереве найдите вершины, максимально удалённые от корня. Расстоянием между вершинами считается количество рёбер в пути.

Формат входных данных

В первой строке задано n — количество вершин в дереве ($1 \leq n \leq 100$). В следующих $n - 1$ строках заданы вершины, являющиеся предками вершин $2, 3, \dots, n$. Вершина 1 является корнем дерева.

Формат выходных данных

В первой строке выведите максимальное расстояние от корня до остальных вершин дерева. Во второй строке выведите, сколько вершин дерева находятся от корня на таком расстоянии. В третьей строке выведите номера этих вершин через пробел в порядке возрастания.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	1
1	2
1	2 3
3	2
1	1
2	3

Задача G. Кротовья нора

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Кротовья нора состоит из пещерок, соединённых коридорами, причём из каждой пещерки выходит ровно три коридора. Коридорами и пещерками пользуется не только крот – в одной из пещерок обосновался жук-носорог, а в другую только что свалился жук-олень. Жук-олень чует своего соперника и спешит встретиться, чтобы выяснить, кто из них будет главным в кротовьей норе!

Выясните, каким минимальным количеством коридоров потребуется пройти жуку-оленю, чтобы попасть в пещерку, где его поджидает жук-носорог.

Формат входных данных

В первой строке заданы три числа: N ($4 \leq N \leq 1000$) – количество пещерок в кротовьей норе, X ($1 \leq X \leq N$) – номер пещерки жука-носорога и Y ($1 \leq Y \leq N$) – номер пещерки жука-оленя. В последующих N строках стоит по три числа в каждой: в $(i + 1)$ -й строке стоят номера пещерок, в которые из i -й пещерки ведут коридоры. Каждый коридор упоминается в строках обеих вершин, которые он соединяет. Никакой коридор не соединяет пещерку с самой собой, и никакие две пещерки не соединены более чем одним коридором. Гарантируется, что $X \neq Y$. Все числа во входных данных целые.

Формат выходных данных

Выведите одно целое число – минимальное количество коридоров, которыми требуется пройти, чтобы попасть из пещерки жука-оленя в пещерку жука-носорога. Если это оказалось невозможно, выведите число -1 .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 1 4 2 3 4 3 4 1 4 1 2 1 2 3	1
6 2 3 4 5 6 4 5 6 4 5 6 1 2 3 1 2 3 1 2 3	2
8 5 4 2 3 4 1 3 4 1 2 4 1 2 3 6 7 8 5 7 8 5 6 8 5 6 7	-1

Задача Н. Лего

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Рассмотрим трёхмерную конструкцию из блоков, аналогичных блокам известного конструктора «Лего». Конструкция состоит из нескольких прямоугольных слоёв одинакового размера, расположенных один над другим. Каждый слой состоит из блоков. Каждый блок — это плоская фигура из кубиков, связанная по стороне. Соседние слои скреплены между собой таким образом, что нижний кубик в каждом столбике скреплен с верхним. Различные же блоки в одном слое никак не скреплены между собой.

Два блока A и B считаются связанными, если существует такая цепочка блоков $A = C_1, C_2, C_3, \dots, C_{n-1}, C_n = B$, что каждые два соседних блока C_i и C_{i+1} в ней скреплены. Очевидно, всю конструкцию можно поделить на компоненты связности. Найдите их количество.

Формат входных данных

В первой строке ввода записаны через пробел три числа x, y и z — длина, ширина и высота конструкции, соответственно ($1 \leq x, y, z \leq 100$). Далее размещены z блоков по y строк, описывающие конструкцию. В каждой из этих строк ровно x символов. Перед каждым блоком расположена дополнительно одна пустая строка. Каждый кубик описывается одной заглавной буквой английского алфавита. Два кубика в одном слое принадлежат одному блоку, если они соседние по стороне и обозначены одной и той же буквой.

Формат выходных данных

Выведите одно число — количество компонент связности в заданной конструкции.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 2 2 AA AA AB CD	1
3 3 2 AAB BCB BAA CAA CEB DDB	2
4 3 1 AABB ACCB AABB	3

Пояснения к примерам

В первом примере четыре отдельных кубика второго слоя крепятся к одному блоку, занимающему весь первый слой. В конструкции одна компонента связности.

Во втором примере блоки из двух кубиков образуют связный «колодец». Два блока в центре связаны друг с другом, но не связаны с «колодцем». Количество компонент связности в этой конструкции равно двум.

В третьем примере в конструкции всего один слой, состоящий из трёх блоков. Отметим, что блоки не обязательно имеют прямоугольную форму. В этой конструкции три компоненты связности.

Задача I. (p, q) -лошадь

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

(p, q) -лошадь — это обобщение обычного шахматного коня. (p, q) -лошадь своим ходом перемещается на p клеток в одном направлении и на q — в другом (перпендикулярном). Например, $(3, 4)$ -лошадь может переместиться с клетки $(5, 6)$ на клетки $(1, 3)$, $(2, 2)$, $(2, 10)$, $(1, 9)$, $(8, 10)$, $(9, 9)$, $(8, 2)$ и $(9, 3)$. Очевидно, что обычный шахматный конь — это $(2, 1)$ -лошадь.

Ваша задача — определить минимальное число ходов, которое требуется (p, q) -лошади, чтобы добраться от одной клетки шахматной доски $M \times N$ до другой. За пределы доски выходить запрещается.

Формат входных данных

Единственная строка входных данных содержит восемь целых чисел $M, N, p, q, x_1, y_1, x_2, y_2$ ($1 \leq x_1, x_2 \leq M \leq 100, 1 \leq y_1, y_2 \leq N \leq 100, 0 \leq p \leq 100, 0 \leq q \leq 100$).

Формат выходных данных

В первой строке выведите целое число k — минимальное число ходов, которое требуется (p, q) -лошади, чтобы добраться из клетки (x_1, y_1) в клетку (x_2, y_2) . Далее должны следовать $k + 1$ строк, в которых должны быть записаны последовательные положения (p, q) -лошади на этом пути.

Если (p, q) -лошадь не может добраться из (x_1, y_1) в (x_2, y_2) , выведите единственное число -1 .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 3 1 1 1 1 3 3	2 1 1 2 2 3 3
2 2 1 1 1 1 1 2	-1

Задача J. Топологическая сортировка

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать: найти такую перестановку вершин, чтобы для каждого ребра графа его начало встречалось в этой перестановке раньше, чем конец.

Формат входных данных

В первой строке даны два натуральных числа N и M — количество вершин и рёбер в графе соответственно ($1 \leq N \leq 100\,000$, $0 \leq M \leq 100\,000$). Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой: номерами начальной и конечной вершин соответственно.

Формат выходных данных

Выведите любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, выведите одно число -1 . Если возможных перестановок несколько, выведите любую из них.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5
3 3 1 2 2 3 3 1	-1

Задача К. Друзья и враги

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

На острове Дженту живут n крокодилов. Каждый крокодил имеет ровно одного друга и ровно одного врага среди остальных крокодилов на острове. Отношения дружбы и вражды симметричны: если крокодил A — друг крокодила B , то крокодил B — друг крокодила A ; если же крокодил A — враг крокодила B , то и крокодил B — враг крокодила A . Никакой крокодил не является одновременно врагом и другом другого крокодила. Кроме того, никакой крокодил не может быть ни другом, ни врагом самому себе.

Крокодилы острова Дженту весьма эмоциональны. Когда встречаются два врага, они со страшной силой колотят по земле хвостами, а затем происходит Битва Крокодилов. Когда встречаются два друга, они исполняют разрушительный Танец Дружбы Крокодилов.

Недавно танцы и битвы разбудили вулкан, находящийся под островом, и остров раскололся на две части. Вулкан успокоился, но крокодилы опасаются, что дальнейшие танцы и битвы разбудят его снова, и обе половины острова зальёт лава.

Теперь крокодилы хотят расселиться по двум половинам острова таким образом, чтобы на каждой половине оказалось *нейтральное* множество крокодилов — такое множество, что никакие два крокодила в нём не являются ни друзьями, ни врагами. Расселившись так, крокодилы не будут устраивать танцы и битвы, а значит, можно надеяться, что вулкан не будет извергаться снова.

Крокодилы острова Дженту мудры, но непрактичны. Они понимают, что при таком устройстве дружбы и вражды разделение крокодилов на два нейтральных множества всегда возможно, но не знают, как именно разделиться на такие множества.

Помогите им! Найдите такое разбиение крокодилов на два множества, что у каждого крокодила в своём множестве нет ни друзей, ни врагов.

Формат входных данных

В первой строке задано натуральное число n — количество крокодилов ($4 \leq n \leq 100$). Следующие n строк описывают крокодилов. В первой из них записаны два числа f_1 и e_1 через пробел — номер друга первого крокодила и номер его врага. Во второй записаны числа f_2 и e_2 — номера друга и врага второго крокодила, и так далее. В последней из этих строк записаны f_n и e_n — друг и враг крокодила с номером n . Крокодилы пронумерованы числами от 1 до n в том порядке, в котором они описываются во входных данных.

Все числа f_k и e_k целые и лежат в пределах от 1 до n , включительно. Отношения дружбы и вражды симметричны. Друг и враг каждого крокодила различны. Никакой крокодил не является ни другом, ни врагом самому себе.

Формат выходных данных

В первой строке выведите n чисел через пробел. Каждое из этих чисел должно быть равно либо 1, либо 2. Если i -е и j -е числа равны, это означает, что крокодилы с номерами i и j оказались на одной половине острова. Если же эти числа различны, значит, крокодилы i и j оказались на разных половинах острова.

Если правильных ответов несколько, можно вывести любой из них.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4	1 2 1 2
2 4	
1 3	
4 2	
3 1	

Задача I. Мосты

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф. Требуется найти все мосты в нём.

Формат входных данных

Первая строка содержит два натуральных числа n и m — количество вершин и рёбер графа соответственно ($n \leq 20\,000$, $m \leq 200\,000$).

Следующие m строк содержат описание рёбер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i и e_i — номерами концов ребра ($1 \leq b_i, e_i \leq n$).

Формат выходных данных

Первая строка должна содержать одно целое число b — количество мостов в заданном графе. В следующей строке выведите b целых чисел — номера рёбер, которые являются мостами, в возрастающем порядке. Рёбра нумеруются с единицы в том порядке, в котором они заданы во входных данных.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
6 7	1
1 2	3
2 3	
3 4	
1 3	
4 5	
4 6	
5 6	

Задача М. Точки сочленения

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам задан неориентированный связный граф с N вершинами и M рёбрами ($1 \leq N \leq 20\,000$, $1 \leq M \leq 200\,000$). В графе отсутствуют петли и кратные рёбра.

Найдите все точки сочленения в заданном графе.

Формат входных данных

Граф задан следующим образом: первая строка содержит числа N и M . Каждая из следующих M строк содержит описание ребра: два целых числа из диапазона от 1 до N — номера концов ребра.

Формат выходных данных

В первой строке выведите число C — количество точек сочленения в заданном графе. В следующей строке выведите C целых чисел — номера вершин, которые являются точками сочленения, в возрастающем порядке.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 6	1
1 2	3
2 3	
1 3	
3 4	
3 5	
4 5	

Задача N. Кодовый замок

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Петя опоздал на тренировку по программированию! Поскольку тренировка проходит в воскресенье, главный вход в учебный корпус, где она проходит, оказался закрыт, а вахтёр ушёл куда-то по своим делам. К счастью, есть другой способ проникнуть в здание — открыть снаружи боковую дверь, на которой установлен кодовый замок.

На пульте замка есть d кнопок с цифрами $0, 1, \dots, d - 1$. Известно, что код, открывающий замок, состоит из k цифр. Замок открывается, если последние k нажатий кнопок образуют код.

Поскольку Петя не имеет понятия, какой код открывает замок, ему придётся перебрать все возможные коды из k цифр. Но, чтобы как можно скорее попасть на тренировку, нужно минимизировать количество нажатий на кнопки. Помогите Пете придумать такую последовательность нажатий на кнопки, при которой все возможные коды были бы проверены, а количество нажатий при этом оказалось бы минимально возможным.

Формат входных данных

В первой строке записаны через пробел два целых числа d и k — количество кнопок на пульте и размер кода, соответственно ($2 \leq d \leq 10$, $1 \leq k \leq 20$).

Формат выходных данных

В первой строке выведите искомую последовательность. Если последовательностей минимальной длины, перебирающих все возможные коды, несколько, можно выводить любую из них. Гарантируется, что d и k таковы, что минимальная длина последовательности не превосходит одного мегабайта.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3	0001011100

Пояснение к примеру

Последовательность в примере перебирает все коды длины 3 в следующем порядке: 000, 001, 010, 101, 011, 111, 110, 100.

Задача О. Жизнь на Марсе

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Майор Том бороздил Марс в поисках жизни, когда марсоход начал его подводить — система питания почти вышла из строя! К счастью, в этот момент на Красной планете бушевала магнитная буря.

Для простоты представим Марс как прямоугольное клетчатое поле размера $n \times m$. В каждой клетке определён некоторый вектор с целыми координатами, не превосходящими по модулю 1 — на такой вектор солнечный ветер сносит марсоход из этой клетки. Ветер, в том числе, может сносить марсоход за пределы Марса, но нельзя этого допустить — вернуться обратно невозможно, и, оказавшись там однажды, майору Тому придется вечно странствовать в открытом космосе.

Несмотря на неисправность, двигатели марсохода всё ещё работают и могут развивать любую целую мощность по любой координате. Вектор мощности суммируется с вектором магнитного поля в клетке, в которой находится марсоход, и он перемещается на получившийся вектор. Пусть, например, майор Том находится в клетке (i, j) с силой поля (r_{ij}, c_{ij}) и включил двигатель на вектор (u, v) . Тогда в следующий момент времени он окажется в клетке с координатами $(i + r_{ij} + u, j + c_{ij} + v)$, после чего двигатель будет снова выключен. Но для того, чтобы развить мощность (u, v) , потребуется $|u| + |v|$ единиц заряда.

Майор Том вызывает Землю и просит сказать, какое минимальное количество заряда потребуется, чтобы добраться до базы.

Формат входных данных

Первая строка входных данных содержит два целых числа n и m — количество строк и столбцов соответственно ($1 \leq n, m \leq 1000$).

Вторая строка содержит четыре целых числа — координаты майора Тома и базы: A_r, A_c, B_r, B_c ($1 \leq A_r, B_r \leq n; 1 \leq A_c, B_c \leq m$).

Каждая из последующих n строк содержит по m пар целых чисел (r_{ij}, c_{ij}) — векторы поля в каждой клетке ($-1 \leq r_{ij}, c_{ij} \leq 1$).

Формат выходных данных

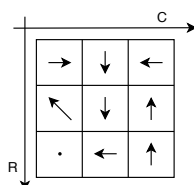
Выведите одно число — минимальное количество заряда, которое потребуется, чтобы добраться из начальной точки в конечную.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 3 1 1 3 3 0 1 1 0 0 -1 -1 -1 1 0 -1 0 0 0 0 -1 -1 0	1
3 5 1 1 2 5 0 -1 0 -1 0 -1 0 -1 0 -1 0 -1 0 -1 0 -1 0 -1 0 -1 0 1 0 1 0 1 0 1 0 1	4

Пояснения к примерам

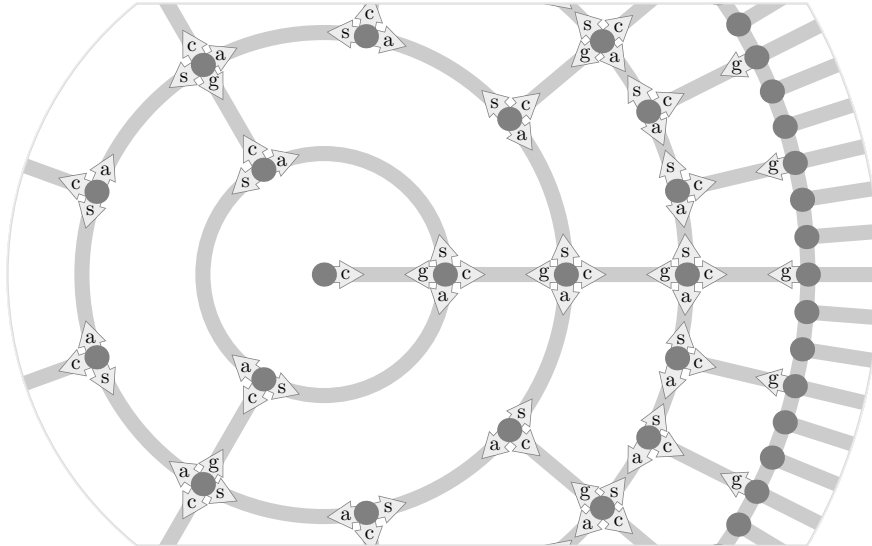
В первом примере можно один раз включить двигатель на вектор $(0, 1)$ в клетке $(2, 2)$.



Задача Р. Галактический Центр

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Эрнест и Селестина прибыли в Галактический Центр, чтобы зарегистрировать свой домик на краю Галактики. Центр — это огромная система залов и коридоров. Эрнест и Селестина очутились в центральном зале и посмотрели на висящую на стене карту ближайших окрестностей.



Залы Центра делятся на уровни: на уровне 0 лишь один зал — центральный, на уровне 1 три зала, на уровне 2 — девять залов, на третьем уровне целых 27 залов, и так далее. Галактический центр поистине огромен, так что количество уровней можно считать бесконечным. Залы каждого уровня, кроме нулевого, соединены коридорами в кольцо. Кроме того, каждый зал соединён коридором с залом следующего уровня. Соответственно, каждый третий зал соединяется с залом предыдущего уровня. Коридоры образуют регулярную структуру, как показано на карте выше. По всем коридорам можно перемещаться в обе стороны. Удивительно, но перемещение по любому коридору в любую сторону занимает одно и то же фиксированное время.

Маршруты перемещений в Центре записываются как строки из маленьких английских букв. Каждая буква означает движение по коридору до соседнего зала. Буквы соответствуют следующим направлениям на карте: «s» (спин) — против часовой стрелки, «a» (антиспин) — по часовой стрелке, «c» (центробежное) — от центра, «g» (гравитационное) — к центру. Адресом зала считается любой маршрут, начинающийся в центральном зале и ведущий в этот зал.

Эрнесту и Селестине нужно сначала поставить подпись в зале по адресу s , а затем — печать в зале по адресу t . Они уже справились с первой половиной этой задачи, и теперь находятся в зале по адресу s . Помогите им найти кратчайший маршрут в зал по адресу t . Длиной маршрута считается количество букв в нём.

Формат входных данных

В первой строке задан адрес s , а во второй — адрес t . Каждый из них содержит от 1 до 35 букв. Гарантируется, что оба адреса являются корректными маршрутами из центрального зала, а также что холлы по этим двум адресам различны.

Формат выходных данных

Выведите кратчайший маршрут из зала по адресу s в зал по адресу t . Если возможных ответов несколько, выведите любой из них.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
ccc csss	gg
cccsca cccacs	aaaa