

Задача А. Валлетский баскетбол

Имя входного файла: `basketball.in`
Имя выходного файла: `basketball.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Два кольца, два конца. . .

баскетбольный судья Гвоздик

Как-то раз в штаб-квартире Корпорации МИФ появился Квингли, бывший охотник на демонов, на Валлете ставший магом государства Та-Хо.

— О, Квингли! Рад тебя видеть, — поприветствовал его Скив.

— Интересно, что здесь делает тахойский маг? — Ааз не был столь учтив.

— Объединённая команда Та-Хо и Вейгаса вызывает вас на матч-реванш. Мы нашли пункт в правилах, согласно которому после изменения правил возможен вызов команды-победителя на матч-реванш за Приз.

— Мы ничего не знаем про этот пункт, да и время, отведённое на реванш, давно прошло, — Аазу явно не хотелось рисковать Призом.

— Гильдия букмекеров уже установила призовой фонд, который будет разделён между участниками Игры в пропорции 2:1.

После того, как Квингли назвал сумму, Ааз неожиданно проявил интерес к новым правилам. Они напоминали распространённый на Пенте баскетбол, но с тем отличием, что мяч в случае соприкосновения с кольцом взрывается, и в случае взрыва попадание в кольцо не засчитывается.

— Мы согласны, — ответил за всех Ааз, уже прикидывая в уме общую прибыль. Но с одним условием — команда будет называться не «Демоны», а «М. И. Ф.»

И начались тренировки.

Напишите программу, которая определяет, будет ли брошенный игроком мяч засчитан.

Кольцо представляет собой окружность, висящую над полом на высоте z_1 в точке с координатами $x = 0, y = 0$, и имеющую радиус r_1 . В начальный момент игрок находится в точке пола (x_0, y_0) и держит в руках на высоте z_0 мяч, который является замкнутым шаром радиуса r_2 ($r_2 < z_0$). Игрок кидает мяч со скоростью $v = (v_x, v_y, v_z)$ и далее он летит под действием силы тяжести. На Пенте ускорение свободного падения равно 10 м/с^2 .

Мяч засчитывается, если в какой-то момент его полёта центр мяча прошёл через плоскость кольца внутри самого кольца со строго отрицательной скоростью по оси Oz , и за время всего полёта мяч не взорвался (взрыв после попадания в кольцо также аннулирует мяч).

Формат входного файла

Во входном файле заданы девять чисел — координаты игрока x_0, y_0 , положение центра мяча z_0 и радиус мяча r_2 , высота и радиус кольца z_1 и r_1 , а также начальная скорость полёта мяча — v_x, v_y и v_z . Координаты и радиусы заданы в метрах, скорость — в метрах в секунду. Все числа вещественные и по модулю не превосходят 100. Кроме того, выполнены условия $0 < r_2 < z_0, r_2 < z_1/2, 0 < r_1$.

Формат выходного файла

Выведите YES, если мяч будет засчитан, и NO в противном случае. Кинув мяч, игрок сразу же отбегает, и полёту мяча не мешает. Как только мяч коснется пола, его полёт прекращается.

Пример

<code>basketball.in</code>	<code>basketball.out</code>
0 0 5 1 3 3 0 0 -1	YES
0 0 5 1 3 3 0 -3.5 0	NO

Задача В. Куб

Имя входного файла: `cube.in`
Имя выходного файла: `cube.out`
Ограничение по времени: 10 секунд
Ограничение по памяти: 256 мегабайт

Главное в жизни — найти цель.

полковник Кольт

Генерал Хью Плохсекир выглядел озабоченным.

— Видите ли, Скив... Я понимаю Ваше решение уйти в отставку с поста придворного мага Поссилтума, более того, после того, что Вы сказали королю, это было единственным достойным решением. Я узнал, что Вы возглавили корпорацию МИФ, и у меня есть заказ для Вашей корпорации...

Оказалось, что генерал не был уверен в том, что Поссилтуму впредь не придётся сталкиваться с магами, и хотел, чтобы Скив придумал для армии Поссилтума какое-то магическое оружие. Более того, генерал каким-то образом сумел выбить из скряги Гримбла финансирование всего этого проекта.

Узнав про выгодное дело, Ааз с Танандой тут же отправились на Базар-На-Деве, где и приобрели «по сходной цене» магическое оружие с огневой характеристикой $100d100$. Генерал скептически осмотрел образцы и заметил, что «испытания покажут». При этом испытывать оружие генерал собирался на «идеальной трёхмерной модели армии».

«Идеальная трёхмерная модель армии» представляла из себя стальной куб со стороной S , состоящий из множества маленьких кубиков со стороной 1. В каждый момент времени Хью может подойти к одной из граней куба снаружи и выстрелить в центр какого-то из маленьких кубиков по направлению, перпендикулярному этой грани. При этом оружие является настолько мощным, что выбивает из большого куба все маленькие кубики, находящиеся на пути его следования.

По совету Ааза испытание проводится в невесомости, так что кубики, оставшиеся без опоры, но еще не выбитые снарядом, остаются на своих местах внутри большого куба.

Испытания оказались успешными, но дотошный Гримбл потребовал детального отчёта об испытаниях. Для генерации отчёта Вам поручено написать программу, которая в некоторые заданные моменты времени узнаёт, сколько маленьких кубиков ещё осталось в большом кубе.

Формат входного файла

В первой строке входного файла задана сторона куба S ($1 \leq S \leq 10^6$) и число операций N , которые проделывает генерал ($1 \leq N \leq 40\,000$). Далее следуют N строк, описывающих операции. Каждая из операций — это либо выстрел, либо запрос на количество оставшихся кубиков. Выстрел описывается в формате $1\ x_i\ y_i\ z_i\ dx_i\ dy_i\ dz_i$, где (x_i, y_i, z_i) — координаты левого нижнего угла маленького кубика, который первым окажется на пути выстрела ($0 \leq x_i, y_i, z_i < S$), а (dx_i, dy_i, dz_i) — направление выстрела (из этих чисел два — нули, а оставшееся равно 1 или -1). Запрос описывается строкой из единственного числа 2.

Формат выходного файла

Для каждого из запросов выведите количество маленьких кубиков, находящихся в данный момент в кубе.

Пример

<code>cube.in</code>	<code>cube.out</code>
3 5	27
2	24
1 1 1 0 0 0 1	22
2	
1 1 2 2 0 -1 0	
2	

Задача С. И-Скакун

Имя входного файла: `dhopper.in`
Имя выходного файла: `dhopper.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Первое. Главное — управление.

правило транспортной эргономики

И-Скакун — устройство, позволяющее не-магам путешествовать по Измерениям. Координаты измерения, в которое необходимо попасть (а также координаты измерения, в которое потом надо вернуться) задаются с помощью устройства, которое представляет из себя строку S , написанную на окружности в $|S|$ ячейках. Конструкция И-Скакунa позволяет переставлять ячейки так, что по окончании перестановки все ячейки останутся на расстоянии не более k от своей исходной позиции (окружность зафиксирована и вращать ее нельзя). Расстояние считается по окружности в обоих направлениях.

Каждая получившаяся в результате строка задаёт некоторое измерение, в которое доставляет И-Скакун. Так что далеко не каждый И-Скакун позволяет попасть в любое из известных измерений. Вам требуется проверить, можно ли с помощью заданного И-Скакунa попасть в измерение, определяемое строкой S' .

Формат входного файла

В первой строке входного файла задано число k ($1 \leq k \leq 200$). Далее следуют две непустые строки одинаковой длины, состоящие из маленьких букв латинского алфавита — S и S' . Длина каждой строки не превосходит 200.

Формат выходного файла

Выведите YES, если попасть в заданное измерение можно, и NO в противном случае.

Пример

<code>dhopper.in</code>	<code>dhopper.out</code>
1 abac cbaa	YES
1 abac acab	NO

Задача D. Информатики

Имя входного файла: `informatics.in`
Имя выходного файла: `informatics.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Троих из 60 наших инженеров зовут
Геннадиями.

из отчёта об успехах геномной инженерии

— Ааз, а почему всё-таки изврская кухня настолько отличается от той, что принята на Пенте?
— спросил Сквив.

— Это потому, что изверги превосходят все остальные расы, — ответил демон.

— И это имеет научные основания, — продолжил он свой рассказ.

Оказывается, что эволюция на Извре происходила несколько иным путём, чем в других измерениях. Простейшие одноклеточные, с которых началось развитие изврской жизни, получили странное название «*информатики*». Их основное предназначение — записывать в свою ДНК информацию и сохранять её для последующей эволюции.

Информатики могли сливаться между собой, более того, в один момент времени могли сливаться сразу несколько информатиков. Образующийся при этом информатик имеет ДНК, которая является конкатенацией ДНК участвовавших в слиянии информатиков.

Кроме этого, информатики могли также делиться. При делении информатика образовывалось несколько информатиков с такой же ДНК, как у родителя, при этом родитель исчезал.

Таким образом, эволюция на Извре была в значительной мере направленной.

— Если ты думаешь, что я просто так тебе это рассказал, то ты ошибаешься, — заметил Ааз.
— Дело в том, что я узнал, что группа учёных ищет финансирование под эксперимент, который в случае успеха позволит устанавливать степень родства между двумя произвольными обитателями Извра. И тогда мы, как обладатели эксклюзивных прав на данную технологию, получим гигантскую прибыль от многочисленных изврских адвокатов, да и от самой полиции: дела о наследстве всегда достаточно прибыльны.

— А каким образом они научатся разбираться в степени родства? — недоверчиво спросил Сквив.

Оказалось, что изврскими учёными запланирован эксперимент по моделированию эволюции, в котором n информатиков, каждый из которых содержит некоторую уникальную информацию, были бы помещены в замкнутое пространство и начали бы развиваться по заданной стратегии. Получающимся при выполнении каждой операции новым информатикам должны последовательно присваиваться номера $n + 1, n + 2, \dots$

А для контроля Ааз решил, что последний этап — определить, содержал ли информатик x во время своей жизни информацию, которая была исходно помещена в информатика y — будет поручен программистам Корпорации МИФ.

Формат входного файла

В первой строке входного файла содержится длина стратегии развития k ($1 \leq k \leq 100$) и исходное количество информатиков n ($1 \leq n \leq 100$). Далее следуют k строк, описывающих стратегию. Операция слияния записывается как $1\ l_i\ a_{(i,1)}\ a_{(i,2)}\ \dots\ a_{(i,l_i)}$, где l_i — количество участвующих в слиянии информатиков ($2 \leq l_i \leq 100$), а $a_{(i,1)}, a_{(i,2)}, \dots$ — их номера. Описание операции деления имеет вид $2\ b_i\ c_i$, где b_i — номер информатика, который разделится, а c_i — количество его детей ($2 \leq c_i \leq 100$). В следующей строке записано число запросов q ($1 \leq q \leq 1000$). Каждый запрос имеет вид $x_i\ y_i$, где x_i — номер информатика, которого нужно проверить на наличие информации, помещенной в начальный момент времени в информатика y_i ($1 \leq y_i \leq n$). Гарантируется, что все номера информатиков, заданные во входном файле, корректны.

Формат выходного файла

Выведите для каждого запроса на отдельной строке слово YES, если информация из y_i содержалась в информатике x_i , и NO в противном случае.

Пример

informatics.in	informatics.out
3 5	YES
1 2 2 3	NO
2 4 3	YES
2 6 2	YES
5	NO
11 3	
8 2	
1 1	
6 2	
2 3	

Задача E. Простая задача

Имя входного файла: `prime.in`
Имя выходного файла: `prime.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Это проще простого.

нестандартное определение единицы

— Идиоты, — сказал Дон Брюс.

— А что случилось? — поинтересовался Скив у Дона Брюса.

— Мои люди опять завалили дело. Они опять всё переусложнили. Вот я и хотел посоветоваться с Вами о том, как лучше всего это исправить. Разумеется, я понимаю, что это выходит за рамки Вашего договора с Синдикатом, и готов пересмотреть в лучшую для Вас сторону условия нашего договора. Мне просто нужен Ваш совет, — сказал Дон Брюс, обращаясь к Скиву, но почему-то смотря в сторону Ааза.

— Совет... Приучайте ваших людей к простому. Начните с того, чтобы в группах, отправляющихся на дело, всегда было простое количество членов Синдиката, — выдал идею Ааз.

— Обычно на дело отправляется «команда» из N человек, которая разбивается на две группы. Если бы можно было для каждого N сказать, можно ли так разбить команду, и если можно, то сразу назвать количественный состав групп, Ваша идея бы помогла.

— Я думаю, что мы сможем найти способ решить эту задачу, — сказал Ааз.

После того, как Дон Брюс вышел, Скив спросил у Ааза:

— А как ты планируешь её решить?

— Очень просто. В некоторых измерениях есть такие люди — программисты. Они занимаются тем, что решают разные задачи. Я думаю, что Корпорации МИФ стоит набрать штат программистов.

Вы оказались в группе программистов, работающей в Корпорации МИФ, и вашим первым заданием стало написание соответствующей программы.

Формат входного файла

В первой строке задано количество «команд», которые необходимо отправить (T , $1 \leq T \leq 100$). Далее следуют T строк, каждая из которых содержит по одному значению N_i ($1 \leq N_i \leq 2^{32} - 1$).

Формат выходного файла

Для каждого из N_i во входном файле выведите два положительных целых числа — искомое разбиение, или два нуля, если такого разбиения не существует.

Пример

<code>prime.in</code>	<code>prime.out</code>
3	3 2
5	3 5
8	0 0
11	

Задача F. Ферзь на торе

Имя входного файла: `queentor.in`
Имя выходного файла: `queentor.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Я готов поклясться на торе. . .

из речи тополога на суде

— Совсем мои дела плохи, — пожаловался Аазу знакомый девол, держатель игрового притона с Базара-на-Деве.

— Клиенты не заходят? — поинтересовался изверг. — Или что хуже?

Оказалось, что хуже. Посетители Базара научились играть во все мудрёные игры, им предлагаемые, и средний выигрыш казино стал приближаться к нулю.

— Если бы ты придумал какую-то хитрую игру. . . Или вспомнил — у вас же на Извре наверняка есть что-то подобное. . . скажем, за половину всех прибылей с этой игры? — с надеждой спросил девол.

Затребовав приличный задаток, Ааз рассказал ему про изврские шахматы. Игра ведётся на тороидальной шахматной доске $m \times n$, которая получается из прямоугольной шахматной доски $m \times n$ склеиванием нижней и верхней стороны, а также левой и правой стороны. При этом перед каждой партией из доски случайным образом вырезаются некоторые клетки.

— Отлично. Пойдёт. Значит, через неделю у меня должна быть схема, которая позволяет в это обыгрывать простаков — и задаток твой.

Так что программисты корпорации МИФ получили задание разработать программу, играющую в изврские шахматы. В рамках этого проекта Вам поставлена задача определить, может ли ферзь, помещённый в некоторую клетку доски, одним ходом переместиться в другую заданную клетку.

Формат входного файла

В первой строке входного файла задан размер доски — два числа m и n ($1 \leq m, n \leq 10^9$), а также количество вырезанных клеток k ($0 \leq k \leq 20\,000$). Далее следуют k строк, в каждой из которых записаны координаты одной из вырезанных клеток x_i и y_i ($0 \leq x_i < m$, $0 \leq y_i < n$). Следующая строка файла содержит число запросов, которые необходимо выполнить — l , $0 \leq l \leq 20\,000$. Далее следуют l строк, в каждой из которых записаны координаты двух клеток — начальной и конечной клеток движения ферзя ($x_{(j,1)}$ $y_{(j,1)}$ $x_{(j,2)}$ $y_{(j,2)}$, $0 \leq x_{(j,1)}, x_{(j,2)} < m$, $0 \leq y_{(j,1)}, y_{(j,2)} < n$). Координаты начальной и конечной клеток различны. Гарантируется, что начальная и конечные клетки в любом запросе не являются вырезанными.

Формат выходного файла

Для каждого из запросов необходимо вывести YES, если ферзь может за один ход прийти от заданной начальной клетки до заданной конечной, и NO в противном случае.

Пример

queentor.in	queentor.out
8 8 2	YES
4 4	YES
6 2	NO
4	NO
1 7 3 1	
2 2 6 6	
3 3 6 5	
5 3 7 1	
3 5 0	YES
1	
0 0 2 3	

Задача G. Заклинание «Локатор»

Имя входного файла: `radarset.in`
Имя выходного файла: `radarset.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

... И поэтому люди прозвали нас
вампирами.

из легенд вампиров

— И зачем мы припёрлись сюда людной ночью? — спросил Ааз.

Скив не знал, что ответить. Он никак не мог привыкнуть, что на Лимбе ночь — это то же самое, что на Пенте день, и что вампиры, жители этого измерения, ведут ночной образ жизни.

— Ааз, тебе не кажется, что вместо того, чтобы читать Скиву морали, нам стоит подумать, как всё же пробраться в это здание? — перевела разговор в более практическое русло Тананда.

— Ну, раз Скив нас сюда ночью вытащил, то пусть он использует заклинание «Магический Локатор», которому я его научил.

Заклинание «Магический Локатор» работает следующим образом: маг находится в точке $(0, 0)$, и в заданный момент времени может обнаружить летящие объекты внутри заданного угла величины α . При этом биссектриса угла повернута против часовой стрелки от оси OY на заданный угол ω .

Любой пролетающий мимо вампир упрощённо представляет собой замкнутый прямоугольник со сторонами, параллельными осям координат.

— Я не знаю, поможет ли оно... — Скив реально оценивал свои силы.

Вам надо помочь Скиву, Аазу и Тананде и определить количество вампиров, которое при заданной конфигурации окажется в поле зрения Скива. Вампир считается находящимся в поле зрения мага, если хотя бы одна его точка находится внутри замкнутого угла видимости «магического локатора».

Формат входного файла

В первой строке входного файла заданы вещественные числа α и ω ($0 \leq \alpha < 2\pi$, $0 \leq \omega < 2\pi$). Во второй строке задано общее количество вампиров N ($1 \leq N \leq 10\,000$). Далее следуют N строк — вещественные координаты вершин вампиров, в формате $x_1 y_1 x_2 y_2$ ($-100 \leq x_1 < x_2 \leq 100$, $-100 \leq y_1 < y_2 \leq 100$).

Формат выходного файла

Выведите количество вампиров, которых увидит Скив, используя заклинание.

Пример

<code>radarset.in</code>	<code>radarset.out</code>
0.5 0.5	2
3	
1 2 3 4	
-2 2 -1 3	
-3 -4 2 5	

Задача Н. Рассадка

Имя входного файла: `seating.in`
Имя выходного файла: `seating.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Будете болтать — расскажу.

сиамский учитель — сиамским близнецам

Популярность Великого Скива выросла до такой степени, что появилось много желающих отдать своих детей к нему в ученики.

Скив сначала отказывался, объясняя, что не обладает достаточным педагогическим опытом для того, чтобы учить других магии, ибо не так давно он сам был учеником.

Но потом состоялся разговор с Аазом.

— Если ты сам не можешь понять того, что все эти ученики потом станут работать королевскими магами в самых разных измерениях и, тем самым, создадут Корпорации МИФ небывалую сеть, то ты хотя бы мог спросить у меня совета, — демон настойчиво советовал всё же взяться за обучение, а ещё лучше — создать в измерениях сеть «Магических Школ Великого Скива», наняв туда опытных магов-педагогов.

И вот настал пробный выпуск. На экзамен, который принимал лично Ааз, прибыли ученики школ из K различных измерений, при этом для i -го измерения известно количество учеников c_i . Столы для учеников расставлены в R вертикальных рядов по M столов в каждом. Для уменьшения вероятности жульничества Ааз хочет рассадить учеников таким образом, чтобы значение функции качества

$$\sum_{i=1}^K \min_{1 \leq a_i < b_i \leq c_i} |y_{(i,a_i)} - y_{(i,b_i)}|$$

было максимально возможным. Здесь $y_{(i,t)}$ — это номер стола t -го ученика i -го измерения в соответствующем вертикальном ряду.

Формат входного файла

В первой строке входного файла заданы числа K , R и M ($1 \leq K \leq 18$, $1 \leq R, M \leq 6$). Во второй строке заданы K чисел $c_i > 1$ (сумма всех c_i не превосходит $R \cdot M$).

Формат выходного файла

В первой строке выходного файла выведите максимально возможное значение функции качества. Далее выведите M строк, каждая из которых содержит R чисел — номер измерения, ученик из которого сидит за соответствующим столом, или 0, если стол свободен.

Пример

<code>seating.in</code>	<code>seating.out</code>
3 3 4	3
3 3 3	1 2 3
	2 3 0
	3 0 1
	0 1 2

Задача I. Суффиксный пулемёт-2

Имя входного файла: `suffix.in`
Имя выходного файла: `suffix.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Болтай — находка для Шалтая.

вся королевская рать

... Полковник Мега Байт, изобретатель легендарного информационного оружия армии Поссилтума — суффиксного пулемёта — радостно принял бывшего курсанта Гвидо. Ещё бы: как мы уже знаем, идея суффиксного пулемёта была подана именно Гвидо и его кузеном Нунцио.

— Рад Вас видеть, и сожалею, что Вы не пошли по военной стезе, — обратился к Гвидо полковник.

— Сейчас я работаю на достаточно могущественную корпорацию, и армейский опыт мне очень сильно помогает, так что я не жалуясь, — Гвидо решил, что немного лести не помешает.

— Да, это замечательно. Я думаю, что Вы сможете посмотреть свежим взглядом на нынешнее устройство суффиксного пулемёта. За это время он сильно изменился... Может, Вам придёт в голову какая-то идея, как его улучшить.

... Уже к вечеру Гвидо рассказывал Тананде и Коррешу устройство секретного агрегата. Суффиксный пулемёт стреляет строчками. Его боеприпасы имеют вид некоторой заданной строки S . Для выстрела суффиксный пулемёт заряжается некоторым суффиксом этой строки, длины не менее K (соответственно, из строки этот суффикс удаляется). При выстреле находившаяся в пулемёте строка вылетает из него и приписывается к *аккумулятору* в конец. Аккумулятор изначально является пустой строкой.

— Ну да, и утверждается, что при вскрытии сейфа с помощью этого оружия в аккумуляторе как раз и получится строка S' .

— Думаю, что ваш пулемёт — пустая трата времени. Кажется мне, что эта строка никогда не сможет оказаться в аккумуляторе, — скептически заметил Корреш.

Для оценки шансов на вскрытие сейфа с приказами командования вам требуется определить, можно ли так стрелять из суффиксного пулемёта, чтобы в результате в аккумуляторе оказался нужный лозунг — строка S' .

Формат входного файла

В первой строке входного файла записано число K ($1 \leq K \leq 4000$). Далее даны две непустых строки S и S' , каждая — длиной не более 4000 маленьких латинских букв.

Формат выходного файла

Если искомой последовательности не существует, то требуется вывести единственное число 0. Иначе необходимо вывести пример последовательности выстрелов, приводящей к требуемому результату. В первой строке выходного файла выводится количество выстрелов, а во второй — длины суффиксов, которые заряжаются в пулемёт на каждом выстреле, в порядке их выполнения.

Пример

<code>suffix.in</code>	<code>suffix.out</code>
1 ab bc	0
1 abacaba baaac	3 2 1 2

Задача J. Дерево

Имя входного файла: `tree.in`
Имя выходного файла: `tree.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Он пытался доказать, что граф является деревом.

из следствия по делу о дуэли

Троль оказался прав — суффиксный пулемёт делу не помог, и вообще, его значимость для армии Поссилтума оказалась сильно преувеличена.

— Гвидо, ты не мог бы описать сейф поподробнее? — вдруг спросила Тананда.

— Ну, сейф как сейф. . . ключевая строка там, все дела. . . сверху ещё как бы дерево нарисовано. . .

— Так, кажется, я знаю, что это за сейф. В Гильдии Убийц мне часто приходилось иметь дело с подобными замками. Механизм представляет собой полное двоичное дерево глубины k и позволяет за одну операцию брать любую вершину, расстояние от которой до листьев не менее 2, и менять порядок поддеревьев ее сыновей следующим образом:

```
      *           *
     / \       / \
    *  *   -> *  *
   /|  |\   /|  |\
  A B  C D  A C  B D
```

Здесь A, B, C, D — некоторые поддеревья.

При этом изначально в листьях дерева слева направо написана некоторая строка S .

— А это действительно так? — спросил Нунцио, который уже устал от всех этих строк и суффиксных пулемётов и хотел побыстрее подменить приказ, закончив, наконец, с этим делом.

— На девять десятых, — сказал Корреш. Но для страховки пусть наши программисты проверят. Заодно и скажут, как его открывать.

Вам требуется определить, можно ли получить из строки S заданную строку T , и если да, то как.

Формат входного файла

В первой строке входного файла записано число k ($2 \leq k \leq 6$). Во второй строке входного файла записана строка S длины 2^k , состоящая из больших и маленьких букв латинского алфавита, цифр, точек и запятых. В третьей строке записана строка T длины 2^k , состоящая из тех же символов, но в другом порядке. Строки S и T различны.

Формат выходного файла

Выведите в выходной файл какой-либо способ получения строки T из строки S не более чем за 10 000 операций. В первой строке выведите число операций K , а далее выведите $K + 1$ строку — последовательные преобразования строки S в строку T . Если не существует способа получить строку T не более, чем за 10 000 операций, выведите единственное число -1 .

Пример

tree.in	tree.out
3 ABCDABCD ABABCCDD	2 ABCDABCD ABABCCDD ABABCCDD
2 ABCD DCBA	-1