Задача A. Короткопалый бюльбюль

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

Короткопалый бюльбюль Леонид из Якутии попал в довольно затруднительное положение. В его уютном жилище поселился нежданный гость — детерминированный конечный автомат (ДКА). Как известно, ДКА допускает не каждое слово, и теперь Леониду нужно тщательно следить за своей речью. В связи с этим у него возникла следующая задача: определить, допускает ли данный ДКА заданное слово.

Формат входных данных

В первой строке входного файла находится слово, состоящее из не более чем 100000 строчных латинских букв. Во второй строке содержатся целые числа n, m и k —количества состояний, переходов и терминальных состояний в автомате соответственно. ($1 \leqslant n, m \leqslant 100000$, $1 \leqslant k \leqslant n$). В следующей строке содержатся k целых чисел — номера терминальных состояний (состояния пронумерованы от 1 до n). В следующих m строках описываются переходы в формате «a b c», где a — номер исходного состояния перехода, b — номер состояния, в которое осуществляется переход, и c — символ (строчная латинская буква), по которому осуществляется переход. Стартовое состояние автомата всегда имеет номер 1. Гарантируется, что из любого со- стояния существует не более одного перехода по каждому символу.

Формат выходных данных

Требуется выдать строку «Accepts», если автомат принимает заданное слово, и «Rejects» в противном случае.

Примеры

bulbul.in	bulbul.out
abacaba 2 3 1 2 1 2 a 2 1 b 2 1 c	Accepts

Задача B. Малый черноголовый дубонос

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

Малому черноголовому дубоносу Евлампии из Караганды остаётся только завидовать бюльбюлю Леониду. На протяжении трёх лет её голубой мечтой было вступление в Социалистическое сопротивление Казахстана, однако, специально для этого взяв отгул на работе и приехав в Алма-Ату, она узнала, что для вступления в эту организацию нужно пройти особое собеседование. И всё было бы хорошо, если бы её интервьюером не оказался недетерминированный конечный автомат (НКА). Таким образом, у Евлампии возникла задача, похожая на ту, с которой столкнулся Леонид: определить, допускает ли данный НКА заданное слово.

Формат входных данных

В первой строке входного файла находится слово, состоящее из не более чем 10 000 строчных латинских букв. Во второй строке содержатся целые числа n, m и k — количества состояний, переходов и терминальных состояний в автомате соответственно ($1 \leqslant n \leqslant 100$, $1 \leqslant m \leqslant 1\,000$, $1 \leqslant k \leqslant n$). В следующей строке содержатся k чисел — номера терминальных состояний (состояния пронумерованы от 1 до n). В следующих m строках описываются переходы в формате «a b c», где a — номер исходного состояния перехода, b — номер состояния, в которое осуществляется переход, и c — символ (строчная латинская буква), по которому осуществляется переход. Стартовое состояние автомата всегда имеет номер 1.

Формат выходных данных

Требуется выдать строку «Accepts», если автомат принимает заданное слово, и «Rejects» в противном случае.

Примеры

hawfinch.in	hawfinch.out
abacaba 4 6 1 2 1 2 a 2 1 c 2 3 b 3 2 a 2 4 b 1 4 a	Accepts

Задача C. Степная тиркуша

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

Степная тиркушка Иннокентий из Лондона не привык жаловаться на жизнь, но сейчас ему действительно не позавидуешь. Дело в том, что в результате невероятного стечения обстоятельств он оказался в Красноярском крае, на берегу реки Сэккэль–Мачиль–Кыкаля. Иннокентий не раз попадал в тяжёлые жизненные ситуации, поэтому давно взял за прави- ло всегда брать с собой в дорогу устройство спутниковой связи собственного изобретения. Пожалуй, единственный недостаток этого устройства — сложная система авторизации. В её основе лежит детерминированный конечный автомат, и для того, чтобы можно было послать сигнал о помощи, нужно назвать все слова, которые допускает этот ДКА, причём только их.

Таким образом, Иннокентию требуется по данному ДКА определить количество допускаемых им слов. Следует заметить, что наш герой не привык к суровому климату Красноярского края, поэтому сейчас его мало волнуют большие числа. Его вполне удовлетворит, если ответ будет найден по модулю числа $10^9 + 7$.

Формат входных данных

В первой строке содержатся числа n, m и k — количества состояний, переходов и тер- минальных состояний в автомате соответственно ($1 \leqslant n, m \leqslant 100\,000$, $1 \leqslant k \leqslant n$). В следующей строке содержатся k чисел — номера терминальных состояний (состояния пронумерованы от 1 до n).

В следующих m строках описываются переходы в формате «a b c», где a — номер исходного состояния перехода, b — номер состояния, в которое осуществляется переход и c — символ (строчная латинская буква), по которому осуществляется переход. Стартовое состояние автомата всегда имеет номер 1. Гарантируется, что из любого состояния существует не более одного перехода по каждому символу.

Формат выходных данных

Выведите количество слов, допускаемых автоматом, по модулю $10^9 + 7$. Если таких слов существует бесконечно много, требуется вывести «`-1`».

Примеры

pratincole.in	pratincole.out
1 1 1 1 1 1 a	-1
3 5 1 3 1 2 a 1 2 b 2 3 a 2 3 b 2 3 c	6

Замечание

Пустая строка является корректным словом.

Задача D. Обыкновенная пустельга

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

Мы не будем обсуждать то, как связана обыкновенная пустельга Зинаида с советскими атомными подводными лодками. Важно сейчас лишь то, что из-за тёмного прошлого она не может пройти упрощённую процедуру получения визы в Бангладеш. Министерство иностранных дел Бангладеша довольно оригинально, поэтому Зинаиде предстоит обратить детерминированный конечный автомат. А именно, нужно для данного ДКА построить другой *детерминированный* конечный автомат, допускающий те и только те слова, которые, будучи перевёрнутыми, допускаются данным ДКА. Дополнительным огорчением для Зинаиды оказался тот факт, что автомат, который ей дали в посольстве, записан в формате, отличном от того, с которым столкнулись её друзья Леонид, Евлампия и Иннокентий.

Формат входных данных

В первой строке содержится непустая строка из отсортированных по возрастанию строч- ных латинских букв — алфавит языка. Длина строки (L) — целое число из отрезка [1, 4]. Во второй строке содержится единственное целое число N из отрезка [1, 10] — число состояний автомата. В третьей строке содержатся следующие целые числа: номер начального состояния S ($1 \leqslant S \leqslant N$), число конечных состояний T ($1 \leqslant T \leqslant N$) и T чисел из отрезка [1, N] — номера конечных состояний. В каждой из следующих N строк содержится L целых чисел: j-е число i-й строки задаёт номер состояния, в которое совершается переход из i-го состояния по j-му символу алфавита.

Формат выходных данных

Выведите обращённый автомат в формате, аналогичном формату входных данных, без указания алфавита.

Примеры

kestrel.in	kestrel.out
ab 5 1 1 4 2 2 3 3 4 5 4 4 5 5	8 1 4 7 8 5 6 2 1 3 4 7 8 5 6 3 4 2 1 7 8 5 6
hi 2 1 1 2 2 1 1 2	2 1 1 2 2 1 1 2

Задача E. Минимизация ДКА

Имя входного файла:	`stdin`
Имя выходного файла:	`stdout`
Ограничение по времени:	8 секунд
Ограничение по памяти:	256 мегабайт

Дан детерминированный конечный автомат A. Постройте детерминированный конечный автомат, принимающий тот же язык, что и A, и имеющий наименьшее возможное число состояний.

Формат входных данных

Первая строка входного файла содержит алфавит Σ, который является непустым подмножеством латинского алфавита (все буквы строчные).

Следующая строка содержит число $|Q|$ — количество состояний автомата ($1 \leqslant |Q| \leqslant 1\,000$).

Состояния нумеруются числами от 1 до $|Q|$.

Следующая строка содержит число q_0 ($1 \leqslant q_0 \leqslant |Q|$) — номер начального состояния, затем число $|T|$ — количество терминальных состояний, затем $|T|$ чисел от 1 до $|Q|$ — номера терминальных состояний.

Следующие $|Q|$ строк содержат по $|\delta|$ чисел — описание функции переходов δ. (Для каждого состояния в отдельной строке приводятся номера состояний, в которые из него ведут переходы по всем символам алфавита).

Формат выходных данных

Выведите описание искомого детерминированного конечного автомата в формате, описанном выше, но без первой строки (строки с алфавитом).

Примеры

stdin	stdout
ab 5 1 2 2 3 2 3 1 4 4 1 3 2 5 5	2 1 1 2 2 2 1 1

Задача F. Непересекающиеся регулярные выражения

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Павел разрабатывает новую социальную сеть ВКосмосе для обитателей спутников Марса Фобоса и Деймоса. Недавно он решил добавить на страницы информацию о том, на каком спутнике живет владелец страницы. Конечно, можно было бы спросить соответствующую информацию о пользователях у них самих, но он решил автоматизировать процесс. Для этого он изучил, как устроены имена обитателей спутников.

Имя каждого пользователя ВКосмосе представляет собой непустую строку, состоящую из строчных букв латинского алфавита. У пользователей с Фобоса имена подходят под регулярное выражение P, а у пользователей с Деймоса имена подходят под регулярное выражение D.

Однако Павел задумался над таким вопросом: а вдруг у какого-нибудь пользователя имя подходит под оба регулярных выражения. Два таких выражения будем называть *непересекающимися*, если никакая непустая строка s не подходит одновременно под оба выражения.

Помогите Павлу определить, являются ли заданные регулярные выражения непересекающимися. Если они не являются непересекающимися, требуется найти кратчайшую непустую строку s, которая подходит под оба выражения.

Замечание

- Одна буква c представляет собой корректное регулярное выражение. Под него подходит единственная строка, состоящая из одной буквы c.
- Операция выбора: если P и Q представляют собой регулярные выражения, то $(P|Q)$ — регулярное выражение, под которое подходят все строки α, которые подходят под P или под Q.
- Конкатенация: если P и Q представляют собой регулярные выражения, то (PQ) представляет собой регулярное выражение, под которое подходят строки α, которые можно представить в виде $\alpha = \beta\gamma$, где β подходит под P, а γ подходит под Q.
- Звездочка Клини: если P представляет собой регулярное выражение, то $(P*)$ представляет собой регулярное выражение, под которое подходят строки α, которые можно представить в виде конкатенации нуля или более строк $\alpha_1\alpha_2\ldots\alpha_k$, где каждая из α_i подходит под P. В частности, пустая строка всегда подходит под звездочку Клини любого выражения.

  Можно опускать скобки, в этом случае звездочка Клини имеет максимальный приоритет, затем конктенация и затем выбор. Например, "`abc*|de`" означает "`(ab(c*))|(de)`".

Формат входных данных

Вход содержит две строки. Первая строка содержит регулярное выражение P. Вторая строка содержит регулярное выражение D. Длина каждого регулярного выражения от 1 до 100 символов.

Формат выходных данных

Если выражения являются непересекающимися, выведите "`Correct`". В противном случае выведите "`Wrong`" на первой строка, а на второй строке выведите кратчайшую строку, которая подходит под оба выражения. Если таких строк несколько, выведите любую.

Примеры

стандартный ввод	стандартный вывод
`a(ab)*b` `a(a|b)*ab`	`Correct`
`a(ab)*a` `a(a|b)*ba`	`Wrong` `aaba`