

Задача А. а.*b

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В задачах 1-4 решением является текстовый файл, содержащий описание детерминированного конечного автомата в установленном формате.

В первой задаче вам необходимо реализовать автомат, принимающий все слова, начинающиеся с буквы а и заканчивающиеся буквой b (то есть удовлетворяющие регулярному выражению а.*b).

Формат описания автомата

Автомат обрабатывает текстовые строки, состоящие из произвольных ASCII-символов (без пробелов и специальных символов). Автомат имеет N состояний, пронумерованных от 1 до N , состояние номер 1 является начальным, у автомата есть одно или несколько терминальных состояний.

Первая строка описания автомата содержит число N . Вторая строка описания автомата содержит номера терминальных состояний автомата (числа от 1 до N), разделенных пробелами.

Все следующие строки описания автомата задают переходы. Описание перехода состоит из 3 частей, записанных через пробел. Первая часть — номер начального состояния автомата. Вторая часть: последовательность из символов входных данных, по которым осуществляется переход. Третья часть: номер конечного состояния. Например, запись:

2 abc 3

означает, что если автомат находится в состоянии 2 и получает на вход один из символов “a”, “b”, “c”, то он переходит в состояние 3. Для каждой пары (состояние, символ) может быть задано не более одного перехода из данного состояния по данному символу.

Вместо второй части может присутствовать символ “*”, обозначающий переход по любому символу, для которого переход не задан явно.

Автомат принимает входное слово, если он завершает работу в терминальном состоянии. Автомат не принимает слово, если он останавливается в нетерминальном состоянии, или если для текущего состояния автомата и входного символа нет перехода (в том числе заданного при помощи “*”) в описании автомата.

Пример автомата, принимающего все слова, содержащие четное число гласных букв:

```
2
1
1 aeiou 2
1 * 1
2 aeiou 1
2 * 2
```

Задача В. Делимость на 3

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Автомат получает на вход целое неотрицательное число (последовательность десятичных цифр). Реализуйте автомат, который принимает входное слово, если число делится на 3. Например, автомат должен принимать слова “0”, “39”, “123” и не должен принимать слова “7”, “14”, “179” и т.д.

Формат ввода автомата

Автомат обрабатывает текстовые строки, состоящие из произвольных ASCII-символов (без пробелов и специальных символов). Автомат имеет N состояний, пронумерованных от 1 до N , состояние номер 1 является начальным, у автомата есть одно или несколько терминальных состояний.

Первая строка описания автомата содержит число N . Вторая строка описания автомата содержит номера терминальных состояний автомата (числа от 1 до N), разделенных пробелами.

Все следующие строки описания автомата задают переходы. Описание перехода состоит из 3 частей, записанных через пробел. Первая часть — номер начального состояния автомата. Вторая часть: последовательность из символов входных данных, по которым осуществляется переход. Третья часть: номер конечного состояния. Например, запись `2 abc 3` означает, что если автомат находится в состоянии 2 и получает на вход один из символов «a», «b», «c», то он переходит в состояние 3. Для каждой пары (состояние, символ) может быть задано не более одного перехода из данного состояния по данному символу. Вместо второй части может присутствовать символ «*», обозначающий переход по любому символу, для которого переход не задан явно.

Автомат принимает входное слово, если он завершает работу в терминальном состоянии. Автомат не принимает слово, если он останавливается в нетерминальном состоянии, или если для текущего состояния автомата и входного символа нет перехода (в том числе заданного при помощи «*») в описании автомата. Пример автомата, принимающего все слова, содержащие четное число гласных букв:

```
2
1
1 aeiou 2
1 * 1
2 aeiou 1
2 * 2
```

Задача С. Каких букв больше?

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Реализуйте автомат, распознающий слова, в которых число букв “а” больше, чем число букв “b”.
Дополнительное условие: все входные слова содержат не более 10 букв. Слово может содержать любые латинские буквы.

Формат ввода автомата

Автомат обрабатывает текстовые строки, состоящие из произвольных ASCII-символов (без пробелов и специальных символов). Автомат имеет N состояний, пронумерованных от 1 до N , состояние номер 1 является начальным, у автомата есть одно или несколько терминальных состояний.

Первая строка описания автомата содержит число N . Вторая строка описания автомата содержит номера терминальных состояний автомата (числа от 1 до N), разделенных пробелами.

Все следующие строки описания автомата задают переходы. Описание перехода состоит из 3 частей, записанных через пробел. Первая часть — номер начального состояния автомата. Вторая часть: последовательность из символов входных данных, по которым осуществляется переход. Третья часть: номер конечного состояния. Например, запись `2 abc 3` означает, что если автомат находится в состоянии 2 и получает на вход один из символов «a», «b», «c», то он переходит в состояние 3. Для каждой пары (состояние, символ) может быть задано не более одного перехода из данного состояния по данному символу. Вместо второй части может присутствовать символ «*», обозначающий переход по любому символу, для которого переход не задан явно.

Автомат принимает входное слово, если он завершает работу в терминальном состоянии. Автомат не принимает слово, если он останавливается в нетерминальном состоянии, или если для текущего состояния автомата и входного символа нет перехода (в том числе заданного при помощи «*») в описании автомата. Пример автомата, принимающего все слова, содержащие четное число гласных букв:

```
2
1
1 aeiou 2
1 * 1
2 aeiou 1
2 * 2
```

Задача D. Сложное выражение

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Реализуйте автомат, распознающий регулярное выражение:

$(ab\{1,2\}|c)+(b*c|b+a*|c+a+)$

Задача Е. 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 бюль-бюлю 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 Алматы, 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 ($1 \leq n \leq 100$, $1 \leq m \leq 1\,000$, $1 \leq k \leq n$). 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

<code>hawfinch.in</code>	<code>hawfinch.out</code>
<code>abacaba</code>	<code>Accepts</code>
<code>4 6 1</code>	
<code>2</code>	
<code>1 2 a</code>	
<code>2 1 c</code>	
<code>2 3 b</code>	
<code>3 2 a</code>	
<code>2 4 b</code>	
<code>1 4 a</code>	

Задача F. Количество слов

Имя входного файла: numwords.in
Имя выходного файла: numwords.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Дан детерминированный конечный автомат. Определите, сколько существует различных слов длины K , принимаемых данным автоматом.

Формат входных данных

Первая строка входных данных содержит два целых числа N и M ($1 \leq N \leq 30$) — количество состояний в автомате и количество переходов. В следующих M строках записаны переходы данного автомата. Каждый переход задается тройкой S_i, C_i, T_i , где S_i — номер исходного состояния перехода ($1 \leq S_i \leq N$), C_i — символ, по которому осуществляется переход (строчная буква латинского алфавита), T_i — конечное состояние перехода ($1 \leq T_i \leq N$).

Далее записано число T — количество терминальных состояний автомата ($0 \leq T \leq N$). В следующей строке записано T различных чисел — номера терминальных состояний. Последняя строка входных данных содержит число K ($0 \leq K \leq 1000$) — длина входного слова.

Начальное состояние автомата имеет номер 1. Если в процессе работы автомата появится невозможный переход (то есть возникает комбинация состояния и символа, не описанная в списке возможных переходов), то такое входное слово считается не распознанным автоматом.

Формат выходных данных

Выведите остаток от деления числа всевозможных входных слов длины K , распознаваемых данным автоматом, на $10^9 + 7$.

Примеры

numwords.in	numwords.out
5 8 1 a 2 1 b 3 2 a 4 2 b 3 4 b 3 3 a 2 3 b 5 5 a 2 4 2 3 4 5 4	10

Задача G. 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 $\alpha_1\alpha_2\dots\alpha_k$, 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 конкатенация 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 строке, 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