

Задача А. 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 Сэkkэль–Мачиль–Кыкала. Иннокентий не раз попадал в тяжёлые жизненные ситуации, поэтому давно взял за правило 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 терминальных состояний в автомате соответственно ($1 \leq n, m \leq 100\,000$, $1 \leq k \leq n$). 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. 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 В. 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 ($1 \leq |Q| \leq 1000$).

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

Следующая строка содержит число q_0 ($1 \leq q_0 \leq |Q|$) — 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

<code>stdin</code>	<code>stdout</code>
ab	2
5	1 1 2
1 2 2 3	2 2
2 3	1 1
1 4	
4 1	
3 2	
5 5	

Задача С. Дерево разбора

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

Арифметические выражения, использующие сложение, вычитание, умножение, деление и возведение в степень определяются следующей грамматикой:

Сложение, вычитание, умножение и деление левоассоциативны, а возведение в степень правоассоциативно.

Для арифметического выражения определено его *дерево разбора*. Это двоичное дерево, в котором внутренние узлы соответствуют бинарным операциям, а листья соответствуют переменным. Дерево строится рекурсивно.

- Дерево для переменной — это дерево из одной вершины, в которой записана эта переменная.
- Дерево для элемента, являющегося выражением в скобках — это дерево для самого выражения.
- Дерево для множителя, являющегося элементом — это дерево для этого элемента. Дерево для множителя вида «элемент e , возведённый в f » — это дерево, в котором в корне записана операция '^', левое поддерево корня — дерево для элемента e , правое поддерево корня — дерево для множителя f .
- Деревья для множителя и слагаемого определяется аналогично, с тем лишь различием, что соответствующие операции лево-ассоциативные.

Вам дано арифметическое выражение, выведите его дерево разбора.

Формат входных данных

Во входном файле содержится корректное арифметическое выражение, состоящее не более, чем из 400 символов

Формат выходных данных

Во входной файл выведите дерево разбора.

Дерево разбора для переменной должно быть размера 1×1 и содержать эту переменную.

Дерево, в корне которого записана операция, с поддеревьями T_1 и T_2 , которые имеют размеры $h_1 \times w_1$ и $h_2 \times w_2$ соответственно, должно быть размера $(\max\{h_1, h_2\} + 2) \times (w_1 + w_2 + 5)$.

Подробнее о формате вывода можно узнать, изучив пример выходного файла. Следует использовать следующие вспомогательные символы: минус '-' (код ASCII 45), точка '.' (код ASCII 46), вертикальная черта '|' (код ASCII 124), квадратные скобки '[' и ']' (коды ASCII 91 and 93).

Примеры

стандартный ввод	стандартный вывод
$(a+b+c)*(d-a)$	<pre> .----[*]----. .----[+]--. .-[]-. .-[+]--. c d a a b</pre>

Задача D. 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

Задача Е. КС-грамматика

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

Дана КС-грамматика, задающая язык L . Дано слово w . Определить, принадлежит ли слово w языку L .

Формат входных данных

В первой строке задано число продукций n и стартовый символ. Следующие n строк содержат продукции вида $A_i \rightarrow N_i$, где A_i — нетерминалы, а N_i — строка из терминалов и нетерминалов. В следующей строке задано слово w ($1 \leq n \leq 50$, $0 \leq |N_i| \leq 5$, $1 \leq |w| \leq 100$).

Каждый нетерминальный символ представлен в виде большой буквы латинского алфавита, а терминальный — маленькой. Все слова состоят только из маленьких букв латинского алфавита. Строка N_i может быть пустой.

Формат выходных данных

Для каждого w_i выведите «yes», если слово принадлежит языку, или «no» в противном случае.

Примеры

стандартный ввод	стандартный вывод
5 S S -> AB S -> C A -> a B -> b T -> c ab	yes

Задача F. Числа Каталана

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

Эндрю любит числа Каталана, а ещё больше Эндрю любит шутки.

Являясь достаточно опытным автором задач, Эндрю готовит достаточно много контестов для различных лагерей и сборов. В каждом контесте он даёт задачу, в которой входные данные и выходные данные состоят из одного числа, а ответами для 0, 1, 2, 3, 4 и 5 являются 1, 1, 2, 5, 14 и 42 соответственно. Однако ответы на большие тесты не являются соответствующими числами Каталана.

Эндрю уже подготовил так много контестов, что идеи для хороших задач с таким свойством начали заканчиваться, поэтому он решил автоматизировать процесс создания таких задач. В качестве хорошего источника таких задач Эндрю рассматривает проблемы подсчёта строк специальной длины в некотором детерминированном конечном автомате. Эндрю выбрал число k — желаемый ответ на задачу для теста 6 и теперь хочет найти детерминированный конечный ответ, который приводит к ответам 1, 1, 2, 5, 14, 42, k для входных данных 0, 1, 2, 3, 4, 5, 6.

По заданному k найдите и выведите детерминированный конечный автомат с таким свойством и с алфавитом не превосходящим 20, среди таких автоматов выберите тот, в котором число состояний минимально.

Формат входных данных

Входной файл состоит из нескольких тестов, разделённых переводами строк, каждый тест представляет из себя одно число k ($120 \leq k \leq 140$).

Входной файл завершается строкой с $k = 0$.

Формат выходных данных

Для каждого теста выведите автомат, являющийся ответом в следующем виде:

В первой строке выведите n — число состояний и s — размер алфавита ($1 \leq s \leq 20$).

Обратите внимание, что вам требуется минимизировать n и не требуется минимизировать s .

Пронумеруйте буквы от 1 до s и состояния от 1 до n , где состояние 1 является начальным состоянием автомата.

Во второй строке выведите n нулей или единиц, определяющих является ли соответствующее состояние терминальным.

В i -ой из последующих n строк выведите s чисел: куда ведёт переход из i -го состояния по соответствующему символу, если из i -го состояния нет перехода по соответствующему символу выведите 0.

Примеры

стандартный ввод	стандартный вывод
131	3 4
0	1 0 0
	1 2 0 0
	1 2 2 3
	2 3 3 0

Замечание

Если обозначить алфавит в примере как $\Sigma = \{a, b, c, d\}$, то пятью словами длины 3, которые принимает автомат, являются “aaa”, “aba”, “baa”, “bba” и “bca”.