

Задача 3-1.

Вам даны два детерминированных конечных автомата A и B . Необходимо определить, эквивалентны ли они.

Во входном файле сначала идет описание автомата A , а потом в том же формате описание автомата B .

Каждое описание начинается с трех целых чисел n, k и l . n — количество состояний автомата. k — количество терминальных состояний. l — количество букв в используемом алфавите. Всегда будут использоваться первые l маленьких латинских букв.

Справедливы ограничения $1 \leq n \leq 1\,000$, $0 \leq k \leq n$, $1 \leq l \leq 26$.

В следующей строке k различных целых чисел от 0 до $n - 1$ — номера терминальных состояний автомата. Все состояния занумерованы от 0 до $n - 1$.

Начальным состоянием автомата считается нулевое состояние. При этом, оно может быть терминальным.

В следующих nl строках перечислены все переходы автомата. Переход записывается в виде $a\ b\ c$, где a — начальное состояние перехода, b — символ для перехода, c — конечное состояние перехода. Переходы могут быть перечислены в произвольном порядке.

В выходной файл необходимо вывести строку **EQUIVALENT**, если автоматы эквивалентны и строку **NOT EQUIVALENT**, если они не эквивалентны.

Автоматы используют общий алфавит — то есть число l будет одно и то же у обоих автоматов.

Пример входа	Пример выхода
4 1 2 2 0 a 1 0 b 0 1 a 1 1 b 2 2 a 3 2 b 3 3 a 3 3 b 3 2 1 2 1 0 a 1 0 b 1 1 a 1 1 b 1	NOT EQUIVALENT
4 3 1 1 2 3 0 a 1 1 a 2 2 a 3 3 a 3 2 1 1 1 0 a 1 1 a 1	EQUIVALENT
4 1 2 2 0 a 1 0 b 0 1 a 1 1 b 2 2 a 3 2 b 3 3 a 3 3 b 3 3 1 2 2 0 a 1 0 b 0 1 a 1 1 b 2 2 a 1 2 b 0	NOT EQUIVALENT

Задача 3-2.

Дан детерминированный конечный автомат A . Необходимо найти минимальный ДКА, который ему эквивалентен.

Во входном файле — описание автомата A . См. описание формата описания автомата в первой задаче. Ограничения на параметры автомата — те же, что и в первой задаче.

В выходной файл необходимо вывести количество состояний в минимальном ДКА, эквивалентном A .

Пример входа	Пример выхода
4 1 2 2 0 a 1 0 b 0 1 a 1 1 b 2 2 a 3 2 b 3 3 a 3 3 b 3	4
4 1 2 3 0 a 1 0 b 2 1 a 1 1 b 3 2 a 2 2 b 3 3 a 3 3 b 3	3
2 1 1 1 0 a 0 1 a 1	1

Задача 3-3.

Вам предлагается реализовать операцию преобразования BWT (Burrows–Wheeler Transform, см. en.wikipedia.org/wiki/Burrows-Wheeler_transform)

Дана непустая строка α , состоящая из строчных латинских букв. Длина n строки не превосходит 100 000. Преобразование осуществляется следующим образом:

- рассматриваются все циклические сдвиги α (всего n строк),
- сдвиги сортируются в лексикографическом порядке и записываются в виде символьной матрицы M размера $n \times n$,
- результатом объявляется строка, получающаяся чтением (сверху вниз) последнего столбца матрицы M .

Например, если $\alpha = \text{ababc}$, то получится следующая матрица:

$$M = \begin{pmatrix} a & b & a & b & c \\ a & b & c & a & b \\ b & a & b & c & a \\ b & c & a & b & a \\ c & a & b & a & b \end{pmatrix}.$$

Итак, ответом будет строка **cbaab**.

В выходной файл нужно вывести единственную строку — результат преобразования строки α .

Пример входа	Пример выхода
ababc	cbaab
a	a
aaaaa	aaaaa
abcde	eabcd

Задача 3-4.

Вам предлагается реализовать алгоритм, схожий с тем, что применяется в методах сжатия LZ ([http://en.wikipedia.org/wiki/LZ77_\(algorithm\)](http://en.wikipedia.org/wiki/LZ77_(algorithm))).

Вам дана строка α , состоящая из строчных латинских букв. Необходимо для каждой позиции i в строке α найти наибольшую по длине подстроку β , начинающуюся в позиции i в α , которая также ранее встречается в строке α . Иными словами нужно найти наибольшую длину $l_i \geq 0$, для которой найдется позиция $i' < i$, такая что $\alpha[i'..i'+l_i-1] = \alpha[i..i+l_i-1]$.

Во входном файле задана единственная строка α . В выходной файл нужно вывести n чисел l_i (где n — длина α), по одному в строке. $1 \leq n \leq 100\,000$.

Пример входа	Пример выхода
ababaab	0 0 3 2 1 2 1
aaaaa	0 4 3 2 1