

Задача А. Наивный поиск подстроки в строке (!) (1 балл)

Имя входного файла: `search1.in`
Имя выходного файла: `search1.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны строки p и t . Требуется найти все вхождения строки p в строку t в качестве подстроки.

Формат входного файла

Первая строка входного файла содержит p , вторая — t ($1 \leq |p|, |t| \leq 10^4$). Строки состоят из букв латинского алфавита.

Формат выходного файла

В первой строке выведите количество вхождений строки p в строку t . Во второй строке выведите в возрастающем порядке номера символов строки t , с которых начинаются вхождения p . Символы нумеруются с единицы.

Примеры

<code>search1.in</code>	<code>search1.out</code>
<code>aba</code>	<code>2</code>
<code>abaCaba</code>	<code>1 5</code>

Задача В. Быстрый поиск подстроки в строке (1 балла)

Имя входного файла: `search2.in`
Имя выходного файла: `search2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны строки p и t . Требуется найти все вхождения строки p в строку t в качестве подстроки.

Формат входного файла

Первая строка входного файла содержит p , вторая — t ($1 \leq |p|, |t| \leq 10^6$). Строки состоят из букв латинского алфавита.

Формат выходного файла

В первой строке выведите количество вхождений строки p в строку t . Во второй строке выведите в возрастающем порядке номера символов строки t , с которых начинаются вхождения p . Символы нумеруются с единицы.

Примеры

<code>search2.in</code>	<code>search2.out</code>
<code>aba</code>	<code>2</code>
<code>abaCaba</code>	<code>1 5</code>

Задача С. Префикс-функция (2 балла)

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

Постройте префикс-функцию для заданной строки s .

Формат входного файла

Первая строка входного файла содержит s ($1 \leq |s| \leq 10^6$). Строка состоит из букв латинского алфавита.

Формат выходного файла

Выведите значения префикс-функции строки s для всех индексов $1, 2, \dots, |s|$.

Примеры

<code>prefix.in</code>	<code>prefix.out</code>
aaaAAA	0 1 2 0 0 0
abacaba	0 0 1 0 1 2 3

Задача D. Автомат Кнута-Морриса-Пратта (2 балла)

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Постройте автомат Кнута-Морриса-Пратта для заданной мощности алфавита n и строки s .

Формат входного файла

В первой строке находится число n ($1 \leq n \leq 26$) — мощность алфавита. Во второй строке находится строка, состоящая из строчных латинских букв ($s \leq 10^5$). Гарантируется, что в данной строке не встречается символ, номер в алфавите которого больше n .

Формат выходного файла

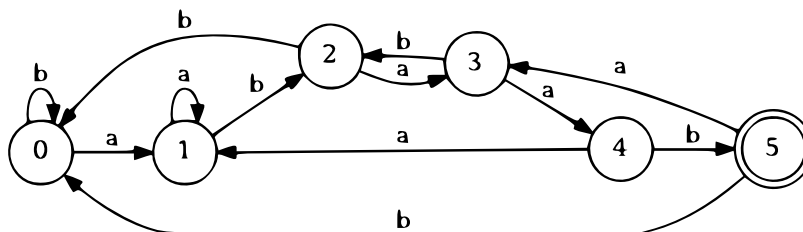
Выведите $|s| + 1$ строк, где j -я строка состоит из n чисел, разделенных пробелом — i -е число в строке обозначает, в какое состояние автомата мы перейдем по i -му символу из j -го состояния автомата. Состояния автомата нумеруются с 0. Если мы передадим в качестве слова автомату префикс данного слова длины k , то мы должны оказаться в состоянии с номером k .

Примеры

стандартный ввод	стандартный вывод
3 abacaba	1 0 0 1 2 0 3 0 0 1 2 4 5 0 0 1 6 0 7 0 0 1 2 4
2 abaab	1 0 1 2 3 0 4 2 1 5 3 0

Примечание

Пример построенного автомата для последнего примера.



Задача Е. Множественный поиск (2 балла)

Имя входного файла: `search4.in`
Имя выходного файла: `search4.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Дано множество строк S и строка t . Требуется для каждой строки $p \in S$ определить, встречается ли она в t как подстрока.

Формат входного файла

Первая строка входного файла содержит целое число n — мощность S ($1 \leq n \leq 10^6$). Следующие n строк содержат по одной строке из S . Сумма длин всех строк из S не превосходит 10^6 . Последняя строка входного файла содержит t ($1 \leq t \leq 10^6$). Все строки состоят из прописных латинских букв.

Формат выходного файла

Для каждой строки из S выведите «YES», если она встречается в t и «NO» в противном случае. Строки нумеруются в порядке появления во входном файле.

Примеры

<code>search4.in</code>	<code>search4.out</code>
3	YES
abc	NO
abcdr	YES
abcde	
xabcdef	