

Задача А. Сравнения подстрок

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

Дана строка. Нужно уметь отвечать на запросы вида: равны ли подстроки $[a..b]$ и $[c..d]$.

Формат входных данных

Сперва строка S (не более 10^5 строчных латинских букв). Далее число M — количество запросов.

В следующих M строках запросы a, b, c, d . $0 \leq M \leq 10^5, 1 \leq a \leq b \leq |S|, 1 \leq c \leq d \leq |S|$

Формат выходных данных

M строк. Выведите **Yes**, если подстроки совпадают, и **No** иначе.

Пример

стандартный ввод	стандартный вывод
trololo	Yes
3	Yes
1 7 1 7	No
3 5 5 7	
1 1 1 5	

Задача В. Префикс-функция

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

Постройте префикс-функцию для заданной строки s .

Формат входных данных

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

Формат выходных данных

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

Пример

стандартный ввод	стандартный вывод
aaaAAA	0 1 2 0 0 0

Задача С. Z-функция

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

Постройте Z-функцию для заданной строки s .

Формат входных данных

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

Формат выходных данных

Выведите значения Z-функции строки s для индексов $2, 3, \dots, |s|$.

Примеры

стандартный ввод	стандартный вывод
aaaAAA	2 1 0 0 0
abacaba	0 1 0 3 0 1

Задача D. Быстрый поиск подстроки в строке

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
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 входных данных

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

Формат выходных данных

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.

Пример

стандартный ввод	стандартный вывод
aba	2
abaCaba	1 5

Задача Е. Поиск периода

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

Дана строка s . Требуется найти минимальную по длине строку t , такую что s представима в виде конкатенации одной или нескольких строк t .

Формат входных данных

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

Формат выходных данных

Выведите длину искомой строки t .

Примеры

стандартный ввод	стандартный вывод
abcbabcabc	3
abacaba	7

Задача F. Подстроки-3

Имя входного файла: `substr3.in`
Имя выходного файла: `substr3.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны K строк из маленьких латинских букв. Требуется найти их наибольшую общую подстроку.

Формат входных данных

В первой строке число K ($1 \leq K \leq 10$).

В следующих K строках — собственно K строк (длины строк от 1 до 10 000).

Формат выходных данных

Наибольшая общая подстрока.

Пример

<code>substr3.in</code>	<code>substr3.out</code>
3 abacaba mycabarchive acabistrue	<code>cab</code>

Задача G. Множественный поиск

Имя входного файла: `search4.in`
Имя выходного файла: `search4.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

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

Формат входных данных

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 строчных латинских букв.

Формат выходных данных

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.

Пример

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

Задача Н. Множественный поиск 2

Имя входного файла: `search5.in`
Имя выходного файла: `search5.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

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

Формат входных данных

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 строчных латинских букв.

Формат выходных данных

Для каждой строки из S выведите одно число: сколько раз она встречается в t . Строки нумеруются в порядке появления во входном файле.

Пример

search5.in	search5.out
3	1
abc	0
abcdr	1
abcde	
xabcdef	

Задача I. Множественный поиск 3

Имя входного файла: `search6.in`
Имя выходного файла: `search6.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Дано множество строк S и строка t . Требуется для каждой строки $p \in S$ найти самое левое и самое правое вхождение в t как подстроки.

Формат входных данных

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 строчных латинских букв.

Формат выходных данных

Для каждой строки из S выведите два числа: индексы самой левой и самой правой позиции, в которых она встречается в t . Если строка не встречается в t ни разу, выведите $-1 -1$. Строки нумеруются в порядке появления во входном файле. Позиции нумеруются с 0.

Пример

<code>search6.in</code>	<code>search6.out</code>
3	0 4
ab	1 1
bcd	-1 -1
abde	
abcdab	

Задача J. Суффиксный массив

Имя входного файла: `array.in`
Имя выходного файла: `array.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Постройте суффиксный массив для заданной строки s , для каждой двух соседних суффиксов найдите длину максимального общего префикса.

Формат входных данных

Первая строка входного файла содержит строку s ($1 \leq |s| \leq 400\,000$). Строка состоит из строчных латинских букв.

Формат выходных данных

В первой строке выведите $|s|$ различных чисел — номера первых символов суффиксов строки s так, чтобы соответствующие суффиксы были упорядочены в лексикографически возрастающем порядке. Во второй строке выведите $|s| - 1$ чисел — длины наибольших общих префиксов.

Пример

<code>array.in</code>	<code>array.out</code>
ababb	1 3 5 2 4 2 0 1 1

Задача К. Количество подстрок

Имя входного файла: `count.in`
Имя выходного файла: `count.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Вычислите количество различных подстрок строки s .

Формат входных данных

Единственная строка входного файла содержит строку s ($1 \leq |s| \leq 400\,000$). Строка состоит из строчных латинских букв.

Формат выходных данных

Выведите одно число — ответ на задачу.

Пример

<code>count.in</code>	<code>count.out</code>
ababb	11

Задача L. Циклические сдвиги

Имя входного файла: `shifts.in`
Имя выходного файла: `shifts.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

k -м *циклическим сдвигом* строки S называется строка, полученная перестановкой k первых символов строки S в конец строки.

Рассмотрим все различные циклические сдвиги строки S и отсортируем их по возрастанию. Требуется вычислить i -ю строчку этого массива.

Например, для строки `abacabac` существует четыре различных циклических сдвига: нулевой (`abacabac`), первый (`bacabaca`), второй (`acabacab`) и третий (`cabacaba`). После сортировки по возрастанию получится такой массив: `abacabac`, `acabacab`, `bacabaca`, `cabacaba`.

Формат входных данных

В первой строке входного файла записана строка S , длиной не более 100 000 символов с ASCII-кодами от 32 до 126. Во второй строке содержится единственное целое число k ($1 \leq k \leq 100\,000$).

Формат выходных данных

В выходной файл выведите k -й по возрастанию циклический сдвиг строки S , или слово IMPOSSIBLE, если такого сдвига не существует.

Примеры

<code>shifts.in</code>	<code>shifts.out</code>
<code>abacabac</code> 4	<code>cabacaba</code>
<code>abacabac</code> 5	IMPOSSIBLE

Задача М. Циклические суффиксы

Имя входного файла: `cyclic.in`
Имя выходного файла: `cyclic.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Рассмотрим строку $S = s_1 s_2 s_3 \dots s_{n-1} s_n$ над алфавитом Σ . Циклическим расширением порядка m строки S назовем строку $s_1 s_2 s_3 \dots s_{n-1} s_n s_1 s_2 \dots$ из m символов; это значит, что мы приписываем строку S саму к себе, пока не получим требуемую длину, и берем префикс длины m .

Циклической строкой \tilde{S} назовем бесконечное циклическое расширение строки S .

Рассмотрим суффиксы циклической строки \tilde{S} . Очевидно, существует не более $|S|$ различных суффиксов: $(n+1)$ -ый суффикс совпадает с первым, $(n+2)$ -ой совпадает со вторым, и так далее. Более того, различных суффиксов может быть даже меньше. Например, если $S = abab$, первые четыре суффикса циклической строки \tilde{S} — это:

$$\begin{aligned}\tilde{S}_1 &= ababababab\dots \\ \tilde{S}_2 &= bababababa\dots \\ \tilde{S}_3 &= ababababab\dots \\ \tilde{S}_4 &= bababababa\dots\end{aligned}$$

Здесь существует всего два различных суффикса, в то время как $|S| = 4$.

Отсортируем первые $|S|$ суффиксов \tilde{S} лексикографически. Если два суффикса совпадают, первым поставим суффикс с меньшим индексом. Теперь нас интересует следующий вопрос: на каком месте в этом списке стоит сама строка \tilde{S} ?

Например, рассмотрим строку $S = cabcab$:

$$\begin{aligned}(1) \quad \tilde{S}_2 &= abcabcabca\dots \\ (2) \quad \tilde{S}_5 &= abcabcabca\dots \\ (3) \quad \tilde{S}_3 &= bcabcabcab\dots \\ (4) \quad \tilde{S}_6 &= bcabcabcab\dots \\ (5) \quad \tilde{S}_1 &= cabcabcabc\dots \\ (6) \quad \tilde{S}_4 &= cabcabcabc\dots\end{aligned}$$

Здесь циклическая строка $\tilde{S} = \tilde{S}_1$ находится на пятом месте.

Вам дана строка S . Ваша задача — найти позицию циклической строки \tilde{S} в описанном порядке.

Формат входных данных

Во входном файле записана единственная строка S ($1 \leq |S| \leq 1\,000\,000$), состоящая из прописных латинских букв.

Формат выходных данных

В выходной файл выведите единственное число — номер строки \tilde{S} в описанном порядке среди первых $|S|$ суффиксов.

Примеры

<code>cyclic.in</code>	<code>cyclic.out</code>
<code>abracadabra</code>	3
<code>cabcab</code>	5

Задача N. Наибольшая общая подстрока

Имя входного файла: `common.in`
Имя выходного файла: `common.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Найдите наибольшую общую подстроку строк s и t .

Формат входных данных

Первая строка входного файла содержит строку s , вторая — t ($1 \leq |s|, |t| \leq 100,000$). Строки состоят из строчных латинских букв.

Формат выходных данных

Выведите одну строку — наибольшую общую подстроку строк s и t . В случае, если ответ не единственный, выведите минимальный лексикографически.

Пример

<code>common.in</code>	<code>common.out</code>
ababb abacabba	aba

Задача О. Рефрен

Имя входного файла: `refrain.in`
Имя выходного файла: `refrain.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Рассмотрим последовательность n целых чисел от 1 до m . Подпоследовательность подряд идущих чисел называется рефреном, если произведение ее длины на количество вхождений в последовательность максимально.

По заданной последовательности требуется найти ее рефрен.

Формат входных данных

Первая строка входного файла содержит два целых числа: n и m ($1 \leq n \leq 150\,000$, $1 \leq m \leq 10$).

Вторая строка содержит n целых чисел от 1 до m .

Формат выходных данных

Первая строка выходного файла должна содержать произведение длины рефрена на количество ее вхождений. Вторая строка должна содержать длину рефрена. Третья строка должна содержать последовательность которая является рефреном.

Пример

<code>refrain.in</code>	<code>refrain.out</code>
8 3	9
1 2 1 2 1 1 2 1	3
	1 2 1