

Задача А. 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 входного файла

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

Формат выходного файла

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 В. Циклические сдвиги

Имя входного файла: `shifts.in`
Имя выходного файла: `shifts.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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 входного файла

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 выходного файла

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

Примеры

shifts.in	shifts.out
abacabac 4	cabacaba
abacabac 5	IMPOSSIBLE

Задача С. Количество подстрок (3 балла)

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

Вычислите количество различных подстрок строки s .

Формат входного файла

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

Формат выходного файла

Выведите одно число — ответ на задачу.

Пример

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

Задача D. Циклические суффиксы

Имя входного файла: `cyclic.in`
Имя выходного файла: `cyclic.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 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 $S = \text{abab}$, первые четыре суффикса циклической строки \tilde{S} — это:

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

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 $S = \text{cabcab}$:

$$\begin{aligned}(1) \quad \tilde{S}_2 &= \text{abcabcbca} \dots \\ (2) \quad \tilde{S}_5 &= \text{abcabcbca} \dots \\ (3) \quad \tilde{S}_3 &= \text{bcabcbcab} \dots \\ (4) \quad \tilde{S}_6 &= \text{bcabcbcab} \dots \\ (5) \quad \tilde{S}_1 &= \text{cabcbcabcb} \dots \\ (6) \quad \tilde{S}_4 &= \text{cabcbcabcb} \dots\end{aligned}$$

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 входного файла

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

Формат выходного файла

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>cabcb</code>	5

Задача Е. Рефрен

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

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 входного файла

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 выходного файла

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>
9 3	9
1 2 1 2 1 3 1 2 1	9
	1 2 1 2 1 3 1 2 1

Задача F. Бордеры

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

Строка t является бордером строки s , если она одновременно является ее собственным суффиксом и собственным префиксом. Например, у строки *abbabba* три бордера: *abba*, *a* и пустая строка.

Дана строка s , посчитайте суммарное число бордеров у всех ее подстрок.

Формат входного файла

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

Формат выходного файла

Выведите одно число — суммарное число бордеров у всех подстрок строки s .

Пример

<code>sumborder.in</code>	<code>sumborder.out</code>
ababb	20

Задача G. Периодические префиксы

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

Дана строка s , найдите для каждого ее префикса максимальное k_i , такое, что префикс длины i является конкатенацией k_i одинаковых строк.

Формат входного файла

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

Формат выходного файла

Выведите $|s|$ чисел: для всех длин от 1 до $|s|$ выведите максимальное k_i , такое, что префикс длины i является конкатенацией k_i одинаковых строк.

Пример

<code>periodic.in</code>	<code>periodic.out</code>
aabaabb	1 2 1 1 1 2 1

Задача Н. Контрольное списывание

Имя входного файла: `kthsubstr.in`
Имя выходного файла: `kthsubstr.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Сегодня на уроке преподаватель Массивов Автомат Укконевич рассказывал своим ученикам про строки, суффиксные структуры и всё такое. Например, он рассказал им, как сравнить две строки A и B *лексикографически*. Если одна из них является префиксом другой, то более короткая будет лексикографически меньше, иначе необходимо сравнить символы стоящие на первой позиции, в которой они отличаются. Строка с меньшим по номеру в алфавите символом на данной позиции и будет лексикографически меньше.

Чтобы проверить понимание учениками нового материала, Автомат Укконевич дал им следующее задание: найти k -ю лексикографически непустую уникальную подстроку строки S .

Так как учитель знает, что Михаил В. и Роман Б. очень любят списывать у известного в узких кругах Максима И., каждый школьник получил своё число k и вынужден был обратиться к вам за помощью.

Формат входного файла

В первой строке входного файла находится строка S ($|S| \leq 10^5$). Вторая строка содержит число k ($1 \leq k \leq 10^{18}$) — порядковый номер запрашиваемой подстроки.

Формат выходного файла

Если ответ существует, выведите искомую подстроку строки S . В противном случае выведите её лексикографически максимальную подстроку.

Пример

<code>kthsubstr.in</code>	<code>kthsubstr.out</code>
abacaba 10	acab

Задача I. Бинарные строки

Имя входного файла: `binary.in`
Имя выходного файла: `binary.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Строка называется *бинарной*, если она состоит только из символов '0' и '1'.

Рассмотрим бинарную строку w длины n . *Суффиксным массивом* строки w называется массив $a[1..n]$ такой, что строка $w[a[i]..n]$ является i -ым в лексикографическом порядке суффиксом строки w . Например, в результате сортировки суффиксов строки $w = "001011"$ они будут расположены следующим образом: "001011", "01011", "011", "1", "1011", "11". Следовательно, суффиксный массив для строки w выглядит так: (1, 2, 4, 6, 3, 5).

Вам дан суффиксный массив a неизвестной строки w . Требуется восстановить строку w .

Формат входного файла

Первая строка входного файла содержит n — длину строки w ($1 \leq n \leq 300\,000$). Вторая строка содержит n различных целых чисел в диапазоне от 1 до n — суффиксный массив строки w .

Формат выходного файла

Выведите единственную строку — искомую бинарную строку w , суффиксный массив которой совпадает с массивом, заданным во входных данных. Если таких строк несколько, выведите любую из них. В случае, если таких строк не существует, выведите "**Error**".

Пример

binary.in	binary.out
6 1 2 4 6 3 5	001011