

Содержание

Must have	2
Задача А. Суффиксный массив [0.25, 256]	2
Задача В. LCP для суффиксного массива [0.25, 256]	3
Обязательные задачи	4
Задача С. Суффиксный массив (3 балла) [0.25, 256]	4
Задача D. Циклические сдвиги [0.25, 256]	5
Дополнительные задачи	6
Задача Е. Рефрен [0.25, 256]	6
Задача F. Набор строк [0.25, 256]	7

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь **примерами**.

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Must have

Задача А. Суффиксный массив [0.25, 256]

Данна строка, требуется построить суффиксный массив для этой строки. Суффиксный массив — лексикографически отсортированный массив всех суффиксов строки. Каждый суффикс задается целым числом — позицией начала.

Строка s лексикографически меньше строки t , если есть такое i , что $s_i < t_i$ и $s_j = t_j$ для всех $j < i$. Или, если такого i не существует и строка s короче строки t .

Здесь s_i — код i -го символа строки s .

Формат входных данных

Файл состоит из единственной строки. Эта строка — **английский литературный текст**. Длина текста не превосходит 10^5 . Коды всех символов в тексте от 32 до 127.

Формат выходных данных

Выведите N чисел — суффиксный массив данной строки.

Пример

suffarray.in	suffarray.out
99 bottles of beer.	14 3 11 19 2 1 15 4 16 17 9 13 8 12 5 18 10 7 6

Замечание

Это **очень** простая задача. Нормальное время написания — 5-10 минут.

Задача В. LCP для суффиксного массива [0.25, 256]

Дана строка длины N и отсортированный массив суффиксов этой строки (т.е. суффиксный массив), вам нужно вычислить LCP. При сортировке строка a считается меньше строки aa . LCP — наибольший общий префикс двух последовательных суффиксов в суффиксном массиве.

Формат входных данных

В первой строке число N ($1 \leq N \leq 10^5$). На второй строке файла дана N строчных латинских букв. В третьей строке N чисел от 1 до N — суффиксный массив (числом i кодируется суффикс, начинающийся с i -го символа).

Формат выходных данных

Выведите $N - 1$ число — значения LCP.

Пример

sufflcp.in	sufflcp.out
5 сасао 2 4 1 3 5	1 0 2 0

Замечание

Суффиксный массив для строки `сасао`:

асао
ао
сасао
сао
о

Подсказка по решению

24-й тест — антихеш тест.

Обязательные задачи

Задача С. Суффиксный массив (3 балла) [0.25, 256]

Постройте суффиксный массив для заданной строки s .

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

Выведите такую перестановку начальных позиций суффиксов, чтобы соответствующие суффиксы были упорядочены в лексикографически возрастающем порядке.

Примеры

array.in	array.out
ababb	1 3 5 2 4

Замечание

ababb
abb
b
babb
bb

Подсказка по решению

А вот и он!

Задача D. Циклические сдвиги [0.25, 256]

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.

Пример

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

Дополнительные задачи

Задача E. Рефрен [0.25, 256]

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

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

refrain.in	refrain.out
9 3	9
1 2 1 2 1 3 1 2 1	3
	1 2 1

Замечание

Эту задачу обязательно сдавать суффмассивом.

Даже если больше вам по душе деревья и автоматы.

Задача F. Набор строк [0.25, 256]

В Инновационном Отделе НИИ Исследований Данных Строк разработана клавиатура для внутреннего пользования, облегчающая набор строк огромной длины. Кроме обычных клавиш, соответствующих маленьким латинским буквам, на клавиатуре есть еще n функциональных клавиш F_1, \dots, F_n , соответствующих заданным строкам из словаря S_1, \dots, S_n . При нажатии такой клавиши F_i строка S_i загружается во внутреннюю память клавиатуры. В каждый момент времени в памяти может находиться не более одной строки из словаря.

Кроме того, в клавиатуру встроен графический манипулятор «Кыш», с помощью которого легким движением руки можно ввести любую подстроку находящейся в памяти строки.

Вася занимается исследованием эффективности данного нововведения. Для этого ему требуется написать программу, которая будет вычислять минимальное необходимое количество действий (нажатий и использований «Кыш») для ввода данной строки S . В момент начала ввода строки память пуста.

Например, если требуется ввести строку “abacaba”, а в словаре есть строки “baba” и “caca”, то это можно сделать за четыре действия — нажать F_1 , выбрать манипулятором подстроку “aba”, затем нажать ‘с’, и опять выбрать манипулятором подстроку “aba”. Если бы нужно было набрать с таким словарем “bacababa”, то это можно сделать за пять действий: ‘b’, F_2 , “aca”, F_1 , “baba”.

Формат входных данных

В первой строке входного файла задано число n ($1 \leq n \leq 50$). В последующих n строках заданы S_i , составленные из не более чем 500 символов. В последней строке вводится непустая строка S , длина которой не превосходит 100 000. Все символы строк — маленькие латинские буквы.

Формат выходных данных

Выведите минимальное необходимое количество действий.

Пример

typing.in	typing.out
2 baba caca abacaba	4
2 baba caca bacababa	5