

Содержание

Must have	2
Задача 20А. Различные подстроки [0.1, 256]	2
Задача 20В. Сравнения подстрок [0.1, 256]	3
Задача 20С. Дана строка [0.1, 256]	4
Задачи здорового человека	5
Задача 20D. Ретрострока [0.1, 256]	5
Задача 20Е. Свобода выбора [0.5, 256]	6
Задача 20F. Подстроки-3 [0.1, 256]	7
Для искателей острых ощущений	8
Задача 20G. Десятичная дробь [0.1, 256]	8
Задача 20H. Две строки [0.1, 256]	10
Задача 20I. Wordperiods [0.5, 256]	11

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

Обратите внимание на GNU C++ компиляторы с суффиксом inc.

Подни можно пользоваться **дополнительной библиотекой** (optimization.h).

То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.

И быструю аллокацию памяти (ускоряет vector-set-map-весь-STL): **пример**.

Для тех, кто хочет разобраться, как всё это работает.

Короткая версия быстрого ввода-вывода (**тык**) и короткая версия аллокатора (**тык**).

Must have

Задача 20А. Различные подстроки [0.1, 256]

Подстрокой строки $s = s_1s_2 \dots s_n$ называется непрерывная подпоследовательность символов этой строки $s_i s_{i+1} s_{i+2} \dots s_{j-1} s_j$.

Дана строка. Сколько различных подстрок, не считая пустой, она содержит?

Формат входных данных

В первой строке входного файла задана строка длины от 1 до 100 символов, включительно. Строка состоит из строчных букв латинского алфавита.

Формат выходных данных

В первой строке выходного файла выведите одно число — количество различных подстрок данной строки, не считая пустой.

Примеры

stdin	stdout
aab	5
dabux	15

Подсказка по решению

Пожалуйста, не перемудрите.

Нужно придумать самое *простое* решение.

Посмотрите, какое ограничение на длины строк? *Нужно простое решение.*

Задача 20В. Сравнения подстрок [0.1, 256]

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

stdin	stdout
trololo	Yes
3	Yes
1 7 1 7	No
3 5 5 7	
1 1 1 5	

Замечание

13-й — антихеш тест.

Мощь хешей в действии.

Задача 20С. Дана строка [0.1, 256]

Даже больше — дано две строки, α и β . Вам требуется узнать, где в строке α можно найти строку β как подстроку и выписать все такие позиции.

Формат входных данных

В первой строке входного файла содержится строка α , во второй — строка β . Строки состоят только из строчных латинских букв ($a-z$), их длины не превосходят 100 000.

Формат выходных данных

В первой строке выходного файла выведите одно число — количество вхождений строки β в строку α . Во второй строке для каждого вхождения выведите номер символа в строке α , где начинается очередная строка β . Вхождения нужно выводить в возрастающем порядке.

Пример

stdin	stdout
abacaba	2
aba	1 5

Подсказка по решению

Собственно он.

Задачи здорового человека

Задача 20D. Ретрострока [0.1, 256]

Строкой S называется последовательность символов S_1, \dots, S_n , где $|S| = n$ — это *длина* строки S .

Для любого k ($1 \leq k \leq |S|$) k -м *префиксом* строки S называется строка S_1, \dots, S_k длины k . Если $k < |S|$, то префикс называется *собственным*.

Аналогично для любого k ($1 \leq k \leq |S|$) k -м *суффиксом* строки S называется строка $S_{|S|-k+1}, \dots, S_{|S|}$ длины k . Если $k < |S|$, то суффикс также называется *собственным*.

Назовём *числом повторяемости* строки S количество её различных собственных суффиксов, каждый из которых совпадает с префиксом той же длины, что и этот суффикс.

Назовём строку *ретрострокой*, если её число повторяемости строго больше чисел повторяемости всех её собственных префиксов.

Дана строка S . Нужно найти её префикс максимальной длины (не обязательно собственный), являющийся ретрострокой.

Формат входных данных

В первой строке входного файла записана строка S , $1 \leq |S| \leq 1\,000\,000$. Строка содержит лишь символы с ASCII-кодом от 33 до 126.

Формат выходных данных

В первой строке выходного файла должен быть выведен префикс S максимальной длины, являющийся ретрострокой.

Пример

stdin	stdout
z	z
aabaabaaabaabaabaaba	aabaabaaabaaba

Подсказка по решению

Мастера π -функции, эту задачу решают в пару строк (кроме собственно π -функции).

Задача 20Е. Свобода выбора [0.5, 256]

Даны две строки, состоящих из заглавных латинских букв. Нужно найти их наибольшую общую подстроку. Полное условие можно посмотреть на [тимусе](#).

Формат входных данных

На первой строке число n ($1 \leq n \leq 10^5$).

На второй и третьей строках находятся по n заглавных английских букв.

Формат выходных данных

Максимальную по длине общую подстроку. Если оптимальных ответов несколько, выведите любой.

Примеры

stdin	stdout
28 VOTEFORTHEGREATALBANIAFORYOU CHOOSETHEGREATALBANIANFUTURE	THEGREATALBANIA

Замечание

19-й — антихеш тест.

Задача на хеши. Разбиралась на практике.

Задача 20F. Подстроки-3 [0.1, 256]

Даны K строк из маленьких латинских букв. Найдите 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.

Примеры

stdin	stdout
3 abacaba mucabarchive acabistrue	cab

Подсказка по решению

Нужно написать хороший код, чтобы пройти TL.
Возможно, понадобится своя хеш-таблица.

Для искателей острых ощущений

Задача 20G. Десятичная дробь [0.1, 256]

В этой задаче требуется найти оптимальный период для бесконечной десятичной дроби.

Рассмотрим бесконечную десятичную дробь $x_0.x_1x_2x_3\dots$, которая является записью некоторого вещественного числа x от 0 до 1 включительно: $x = x_0 + x_1 \cdot 10^{-1} + x_2 \cdot 10^{-2} + x_3 \cdot 10^{-3} + \dots$. Здесь x_i — это десятичные цифры от 0 до 9. В этой задаче нет никаких ограничений на дробь, кроме приведённых выше. В частности, это означает, что, например, 0.999999... и 1.000000... — корректные бесконечные десятичные дроби, являющиеся записью одного и того же вещественного числа 1.

Периодическая десятичная дробь — это способ записи бесконечной десятичной дроби в виде $y_0.y_1y_2y_3\dots y_r(y_{r+1}y_{r+2}\dots y_s)$, где $r \geq 0$ и $s > r$. Эту запись можно *раскрыть* в бесконечную десятичную дробь $y_0.y_1y_2y_3\dots y_r y_{r+1}y_{r+2}\dots y_s y_{r+1}y_{r+2}\dots y_s y_{r+1}y_{r+2}\dots y_s \dots$, то есть бесконечную дробь, начинающуюся с $y_0.y_1y_2y_3\dots y_r$ и затем повторяющую последовательность цифр $y_{r+1}y_{r+2}\dots y_s$ в бесконечном цикле. Будем говорить, что r — это длина *предпериода*, а $s - r$ — это длина *периода*. Не всякую бесконечную десятичную дробь можно записать как периодическую. На самом деле такое представление существует тогда и только тогда, когда вещественное число x является рациональным.

Нам заданы несколько первых цифр бесконечной десятичной дроби, оставшиеся цифры просто отброшены (никакого округления не происходит). Теперь мы хотим записать какую-нибудь периодическую десятичную дробь, раскрыв которую, мы получим дробь, начинающуюся с заданной конечной части. Среди таких бесконечных десятичных дробей найдите ту, у которой сумма длин предпериода и периода минимально возможная.

Формат входных данных

Первая строка ввода содержит начало бесконечной десятичной дроби в формате $x_0.x_1x_2x_3\dots x_n$ ($1 \leq n \leq 1\,000\,000$). Здесь x_i — десятичные цифры от 0 до 9, а вещественное число x , записью которого является дробь, лежит между 0 и 1 включительно.

Формат выходных данных

Выведите одну строку, содержащую периодическую десятичную дробь в формате $y_0.y_1y_2y_3\dots y_r(y_{r+1}y_{r+2}\dots y_s)$, где $r \geq 0$ и $s > r$. Здесь y_i — десятичные цифры от 0 до 9. Раскрыв период, мы должны получить бесконечную цепную дробь, начинающуюся с $x_0.x_1x_2x_3\dots x_n$ (это начало задано во вводе), а сумма длин предпериода и периода должна быть минимально возможной. Если возможных ответов несколько, выведите один любой из них. Гарантируется, что хотя бы один ответ существует.

Примеры

stdin	stdout
0.9999999	0.(9)
0.63573573	0.6(357)
0.123456789	0.12345(6789)

Пояснения к примерам

В первом примере периодическая десятичная дробь 0.(9) раскрывается в бесконечную десятичную дробь 0.999..., которая начинается с 0.9999999. Здесь длина предпериода равна 0, а длина периода равна 1. Другие ответы, например, 0.9(99) или даже 0.99999998(7), также

раскрываются в дробь, начинающуюся с 0.9999999, но они не оптимальны. Заметим, что, хотя $0.9999999\dots = 1$ как вещественное число, ответ $1.(0)$ **не** является корректным, так как он раскрывается в дробь, которая не начинается на 0.9999999.

Во втором примере ответ $0.6(357)$ раскрывается в $0.6357357357357\dots$. Здесь длина предпериода равна 1, а длина периода равна 3. Первые несколько цифр соответствуют заданному началу.

В третьем примере возможные ответы таковы: $0.(123456789)$, $0.1(23456789)$, \dots , $0.12345678(9)$. Помните, что длина предпериода должна быть неотрицательна, а длина периода — положительна.

Задача 20Н. Две строки [0.1, 256]

Вам заданы две строки длиной не более 50 000 символов. Назовем строку хорошей, если она удовлетворяет условию, что если дописать ее в конец самой себе достаточно много раз, то в полученной строке будут содержаться в качестве подстрок обе заданные строки.

Например, для строк `ababa` и `bab` строка `ab` является хорошей — действительно, дописав ее в конец себе два раза, мы получим строку `ababab`, которая содержит обе заданные строки в качестве подстрок.

Для двух заданных строк найдите самую короткую хорошую строку.

Формат входных данных

Входной файл содержит две заданные строки. Строки состоят из символов с ASCII-кодами от 33 до 127. Длина каждой из них не превышает 50 000.

Формат выходных данных

Выведите в выходной файл ответ на задачу. Если существует несколько различных оптимальных хороших строк, то выведите любую.

Пример

stdin	stdout
ababa bab	ab

Задача 201. Wordperiods [0.5, 256]

Назовём периодом слова s такое слово t , длина которого не превосходит длины слова s и для которого существует такое натуральное число k , что слово s является префиксом слова t^k (то есть слова, полученного конкатенацией k копий слова t). Например, периодами слова **xuzxuzx** являются слова **xuz**, **xuzxuz**, **xuzxuzx**.

Пусть имеется некоторое слово w длины l . Рассмотрим l слов длины $l-1$, i — из которых получено из слова w вычёркиванием его i — буквы. Для каждого из этих слов найдём период наименьшей длины. Выведите наименьшее из получившихся l чисел.

Формат входных данных

В первой строке входного файла задано целое число d ($1 \leq d \leq 10$) — количество тестовых примеров. В последующих d строках заданы тестовые примеры, по одному на строку. В начале i — тестового примера идёт число n_i ($2 \leq n_i \leq 200\,000$) — длина l слова w . Далее через пробел следует слово w , состоящее из l строчных латинских букв.

Формат выходных данных

Для каждого тестового примера выведите в отдельной строке одно число — минимальную длину периода слова, полученного выбрасыванием из исходного слова одной буквы.

Пример

stdin	stdout
1	2
8 ababcaba	

Пояснение к примеру

Для слова w из тестового примера имеем следующие слова, наименьшие периоды и их длины:

babcaba — **babca** длина 5
aabcaba — **aabcab** длина 6
abbcaba — **abbcab** длина 6
abacaba — **abac** длина 4
abababa — **ab** длина 2
ababcba — **ababc** длина 6
ababcaa — **ababca** длина 6
ababcab — **ababc** длина 5

Соответственно, наименьшая из длин равна 2, что и является ответом на тестовый пример.