

Содержание

Для любителей литературы	2
Задача 13А. Циклические суффиксы [0.5 sec, 256 mb]	2
Задача 13В. Десятичная дробь [0.5 sec, 256 mb]	3
Для искателей острых ощущений	5
Задача 13С. Архиватор [1 sec, 256 mb]	5

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

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.

Обратите внимание на GNU C++ компиляторы с суффиксом `inc`, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

Для любителей литературы

Задача 13А. Циклические суффиксы [0.5 сек, 256 mb]

Рассмотрим строку $S = s_1s_2s_3 \dots s_{n-1}s_n$ над алфавитом Σ . Циклическим расширением порядка m строки S назовем строку $s_1s_2s_3 \dots s_{n-1}s_ns_1s_2 \dots$ 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 $S = cabcab$:

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

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

stdin	stdout
abracadabra	3
cabcab	5

Замечание

Заметьте, мы пока знаем только Z-функцию, π -функцию, хеши. Этого хватит.

Задача 13В. Десятичная дробь [0.5 sec, 256 mb]

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, $0.999999\dots$ и $1.000000\dots$ — 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 $0.999\dots$, 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.

Замечание

Заметьте, мы пока знаем только Z -функцию, π -функцию, хеши. Этого хватит.

Для искателей острых ощущений

Задача 13С. Архиватор [1 сек, 256 mb]

Вася решил покорить рынок лучших архиваторов мира. Совсем недавно он придумал очень нетривиальную идею для сжатия текста из маленьких латинских букв. А именно, он решил, что можно хранить текст как последовательность команд. Команды бывают двух типов:

- «с»: дописать к текущей строке символ s .
- «i k»: дописать к текущей строке k символов один за другим. При этом первый дописываемый символ совпадает с символом i текущей строки, второй с символом $i + 1$ и так далее, k -ый добавляемый символ совпадает с символом $i + k - 1$. Гарантируется, что i не превосходит текущей длины строки.

Например последовательность команд «a, b, 1 3» кодирует строку «ababa», а последовательность команд «a, 1 3, b, 3 3» кодирует строку «aaaabaab».

На хранение команды первого типа Васе требуется 1 байт, а второго типа 5 байт. К сожалению, пока Вася умеет только по командам восстановить исходную строку, а наоборот не умеет. Вам предлагается помочь бедному Васе в покорении архиваторного рынка. Найдите последовательность команд, которая архивирует заданную строку указанным способом, при этом потратив как можно меньше байт на ее хранение.

Формат входных данных

Во входном файле вам задана строка s из строчных латинских букв длиной не более 4 000 символов.

Формат выходных данных

В первой строке выходного файла вы должны вывести количество байт, которое потребуется для хранения последовательности команд и количество команд в последовательности. На следующих строках выведите саму последовательность, по одной команде на строке. Если команда первого типа, то выведите просто букву, иначе выведите два числа: позиция символа (символы нумеруются начиная с единицы) в строке s , начиная с которого надо начать копирование, и количество символов, которое надо скопировать.

Примеры

stdin	stdout
abcdqwertyqwertyu	16 12
	a
	b
	c
	d
	q
	w
	e
	r
	t
	y
	5 6
	u

Подсказка по решению

LCP! (но при желании можно использовать и z-функция/хеши).