Содержание

Общая информация:

Вход в контест: `http://contest.yandex.ru/contest/1122/`
Дедлайн на задачи: 2 недели, до 22-го марта 23:59.
К каждой главе есть более простые задачи (base), посложнее (advanced), и сложные (hard).
В скобках к каждой главе написано сколько любых задач из этой главы нужно сдать.
Сайт курса: `https://compscicenter.ru/courses/algorithms-2/2015-spring/`
Семинары ведёт Сергей Владимирович Копелиович,
контакты: burunduk30@gmail.com, vk.com/burunduk1

В каждом условии 2 таймлимита: для `C/C++` и для `Java, Python`.

04.Base [4/4]

Задача 4A. Различные подстроки [0.5 sec (1 sec), 256 mb]

Подстрокой строки $\S = s_1 s_2 \dots s_n$ 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

unequal.in	unequal.out
aab	5
dabyx	15

Задача 4B. Префикс-функция [0.5 sec (1 sec), 256 mb]

Дана строка s. Найдите сумму значений префикс-функции для всех позиций строки s.

Формат входных данных

Во входном файле записана единственная строка s ($1 \leqslant |s| \leqslant 150\,000$).

Формат выходных данных

В выходной файл выведите одно число — ответ на задачу.

Пример

prefix.in	prefix.out
abcaabacabc	11

Задача 4C. Дана строка [0.25 sec (1 sec), 256 mb]

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

search.in	search.out
abacaba aba	2 1 5

Задача 4D. Сравнения подстрок [0.3 sec (0.6 sec), 256 mb]

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. $0 \leqslant M \leqslant 10^5, 1 \leqslant a \leqslant b \leqslant |S|, 1 \leqslant c \leqslant d \leqslant |S|$

Формат выходных данных

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

Пример

substrcmp.in	substrcmp.out
trololo 3 1 7 1 7 3 5 5 7 1 1 1 5	Yes Yes No

04.Advanced [3/4]

Задача 4E. Основание строки [0.15 sec (0.45 sec), 256 mb]

Строка S была записана много раз подряд, после чего из получившейся строки взяли подстроку и дали вам. Ваша задача определить минимально возможную длину исходной строки S.

Формат входных данных

В первой и единственной строке входного файла записана строка, которая содержит только латинские буквы, длина строки не превышает 50 000 символов.

Формат выходных данных

В выходной файл выведите ответ на задачу.

Пример

basis.in	basis.out
zzz	1
bcabcab	3

Задача 4F. Циклические суффиксы [0.5 sec (1 sec), 256 mb]

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 $S = \texttt{abab}$, первые четыре суффикса циклической строки $\tilde{S}$ — это:

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

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 $S = \texttt{cabcab}$:

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

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 ($1 \leqslant |S| \leqslant 1\,000\,000$), 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

cyclic.in	cyclic.out
abracadabra	3
cabcab	5

Задача 4G. Кубики [0.3 sec (0.6 sec), 256 mb]

Привидение Петя любит играть со своими кубиками. Он любит выкладывать их в ряд и разглядывать свое творение. Однако недавно друзья решили подшутить над Петей и поставили в его игровой комнате зеркало. Ведь всем известно, что привидения не отражаются в зеркале! А кубики отражаются.

Теперь Петя видит перед собой N цветных кубиков, но не знает, какие из этих кубиков нестоящие, а какие — всего лишь отражение в зеркале. Помогите Пете! Выясните, сколько кубиков может быть у Пети. Петя видит отражение всех кубиков в зеркале и часть кубиков, которая находится перед ним. Часть кубиков может быть позади Пети, их он не видит.

Формат входных данных

Первая строка входного файла содержит число N ($1 \leqslant N \leqslant 100\,000$) и количество различных цветов, в которые могут быть раскрашены кубики — M ($1 \leqslant M \leqslant 100\,000$). Следующая строка содержит N целых чисел от 1 до M — цвета кубиков.

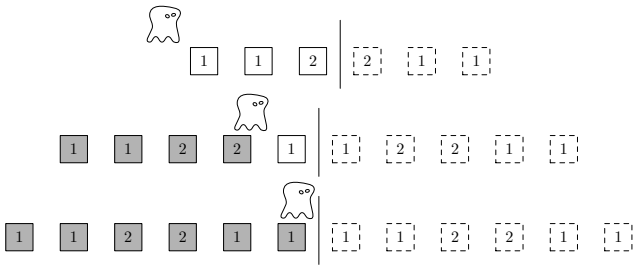
Формат выходных данных

Выведите в выходной файл все такие K, что у Пети может быть K кубиков.

Пример

cubes.in	cubes.out
6 2 1 1 2 2 1 1	3 5 6

В приведенном примере взаимные расположения Пети, кубиков и зеркала приведены на рисунке. Петя смотрит вправо, затененные на рисунке кубики находятся позади Пети и поэтому он их не видит.



Задача 4H. Преобразование строковых функций [0.5 sec (1 sec), 256 mb]

Для строки S определим Z-функцию следующим образом: $Z[i] = lcp(S, S[i..|S|])$, где $lcp(S_1, S_2)$ равно длине наибольшего общего префикса строк S_1 и S_2. Например, для $S = abacabaa$ Z-функция равна $[8, 0, 1, 0, 3, 0, 1, 1]$.

Для строки S определим ее префикс-функцию: $\pi[i] = \max\{k | 0 \leqslant k < i, S[1..k] = S[i-k+1..i]\}$. Например, для $S = abacabaa$ ее префикс-функция имеет вид: $[0, 0, 1, 0, 1, 2, 3, 1]$.

Для некоторой строки S была посчитана ее Z-функция, а строка S была утеряна. Ваша задача получить ее префикс-функцию по заданной Z-функции.

Формат входных данных

В первой строке входного файла содержится натуральное число N ($1 \leqslant N \leqslant 200\,000$), где N — длина S. Во второй строке записана Z-функция строки S.

Формат выходных данных

Выведите N чисел — искомую префикс-функцию.

Пример

trans.in	trans.out
8 8 0 1 0 3 0 1 1	0 0 1 0 1 2 3 1

04.Hard [0/4]

Задача 4I. Z в prefix [0.5 sec (1 sec), 256 mb]

Для строки S определим Z-функцию следующим образом: $Z[i] = lcp(S, S[i..|S|])$, где $lcp(S_1, S_2)$ равно длине наибольшего общего префикса строк S_1 и S_2. Например, для $S = abacabaa$ Z-функция равна $[8, 0, 1, 0, 3, 0, 1, 1]$.

Для строки S определим ее префикс-функцию: $\pi[i] = \max\{k | 0 \leqslant k < i, S[1..k] = S[i-k+1..i]\}$. Например, для $S = abacabaa$ ее префикс-функция имеет вид: $[0, 0, 1, 0, 1, 2, 3, 1]$.

Для некоторой строки S была посчитана ее префикс-функция, а строка S была утеряна. Ваша задача получить ее Z-функцию по заданной префикс-функции.

Формат входных данных

В первой строке входного файла содержится натуральное число N ($1 \leqslant N \leqslant 200\,000$), где N — длина S. Во второй строке записана префикс-функция строки S.

Формат выходных данных

Выведите N чисел — искомую Z-функцию.

Пример

invtrans.in	invtrans.out
8 0 0 1 0 1 2 3 1	8 0 1 0 3 0 1 1

Задача 4J. Word Cover [0.5 sec (1 sec), 256 mb]

Говорят, что строка α покрывается строку β, если для каждой позиции в строке β существует вхождение α, как подстроки β, содержащее эту позицию. Например, строка "aba" покрывает строку "abaabaababa", но не покрывает строку "baba". Конечно, строка покрывает саму себя. Компактностью строки β назовем длину самой короткой строки, которая покрывает β.

Вам дана строка w. Для каждого префикса $w[1..k]$ строки w найдите его компактность.

Формат входных данных

Непустая строка w, состоящая из строчных букв английского алфавита.

Длина w не превосходит 250 000.

Формат выходных данных

Для каждого k от 1 до $|w|$ выведите компактность $w[1..k]$.

Примеры

cover.in	cover.out
abaabaababa	1 2 3 4 5 3 4 5 3 10 3

Задача 4K. Wordperiods [0.8 sec (1.5 sec), 256 mb]

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 **xyzxyzx** являются слова **xyz**, **xyzxyz**, **xyzxyzx**.

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 ($1 \le d \le 10$) — количество тестовых примеров. В последующих d строках заданы тестовые примеры, по одному на строку. В начале $i-$ тестового примера идёт число n_i ($2 \le n_i \le 200\,000$) — 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

wordperiod.in	wordperiod.out
1 8 ababcaba	2

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 — **ababcb** длина 6
ababcaa — **ababca** длина 6
ababcab — **ababc** длина 5

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

Задача 4L. Архиватор [0.8 sec (1.5 sec), 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:

- «`c`»: дописать к текущей строке символ c.
- «`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 4000 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

archiver.in	archiver.out
abcdqwertyqwertyu	16 12 a b c d q w e r t y 5 6 u