

## Содержание

<b>Обязательные задачи</b>	<b>3</b>
Задача 25А. Различные подстроки [0.2 sec, 256 mb]	3
Задача 25В. Дана строка [0.2 sec, 256 mb]	4
Задача 25С. Основание строки [0.2 sec, 256 mb]	5
Задача 25D. Циклические суффиксы [0.2 sec, 256 mb]	6
Задача 25Е. Десятичная дробь [0.3 sec, 256 mb]	7
Задача 25F. Сравнения подстрок [0.3 sec, 256 mb]	9
Задача 25G. Кубики [0.3 sec, 256 mb]	10
<b>Бонус</b>	<b>11</b>
Задача 25H. Архиватор [0.8 sec, 256 mb]	11
Задача 25I. Две строки [0.3 sec, 256 mb]	12
Задача 25J. Word Cover [0.3 sec, 256 mb]	13
Задача 25K. Преобразование строковых функций [0.3 sec, 256 mb]	14
Задача 25L. Преобразование функций: обратная задача [0.3 sec, 256 mb]	15

---

### **Пример работы с файлами.**

Если вы не умеете читать/выводить данные, или открывать файлы, воспользуйтесь примерами. <http://acm.math.spbu.ru/~sk1/algo/sum/>

### **Пример работы с файлами.**

В некоторых задачах большой ввод и вывод. Про ввод-вывод в C++:

[http://acm.math.spbu.ru/~sk1/algo/input-output/cpp\\_common.html](http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html)

Имеет смысл пользоваться супер быстрым вводом-выводом. Две версии:

[http://acm.math.spbu.ru/~sk1/algo/input-output/io\\_export.cpp.html](http://acm.math.spbu.ru/~sk1/algo/input-output/io_export.cpp.html)

[http://acm.math.spbu.ru/~sk1/algo/input-output/fread\\_write\\_export.cpp.html](http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html)

### **Выделение памяти.**

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

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

## Обязательные задачи

### Задача 25А. Различные подстроки [0.2 sec, 256 mb]

*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

<code>unequal.in</code>	<code>unequal.out</code>
<code>aab</code>	5
<code>dabux</code>	15

**Задача 25В. Дана строка [0.2 sec, 256 mb]**

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

**Формат входных данных**

В первой строке входного файла содержится строка  $\alpha$ , во второй — строка  $\beta$ . 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  $\beta$  в строку  $\alpha$ . Во второй строке для каждого вхождения выведите номер символа в строке  $\alpha$ , где начинается очередная строка  $\beta$ . Вхождения нужно выводить в возрастающем порядке.

**Пример**

search.in	search.out
abacaba	2
aba	1 5

**Задача 25С. Основание строки [0.2 сек, 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**

<code>basis.in</code>	<code>basis.out</code>
<code>zzz</code>	<code>1</code>
<code>bcabcab</code>	<code>3</code>

### Задача 25D. Циклические суффиксы [0.2 sec, 256 mb]

Рассмотрим строку  $S = s_1s_2s_3 \dots s_{n-1}s_n$  над алфавитом  $\Sigma$ . Циклическим расширением порядка  $t$  строки  $S$  назовем строку  $s_1s_2s_3 \dots s_{n-1}s_ns_1s_2 \dots$  из  $t$  символов; это значит, что мы приписываем строку  $S$  саму к себе, пока не получим требуемую длину, и берем префикс длины  $t$ .

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  $S = cabcab$ :

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

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

cyclic.in	cyclic.out
abracadabra	3
cabcab	5

### Задача 25Е. Десятичная дробь [0.3 сек, 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

decimal.in	decimal.out
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.



**Задача 25F. Сравнения подстрок [0.3 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  $a,b,c,d$ .  $0 \leq M \leq 10^5, 1 \leq a \leq b \leq |S|, 1 \leq c \leq d \leq |S|$

**Формат выходных данных**

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

**Пример**

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

### Задача 25G. Кубики [0.3 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 настоящие, 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  $N$  ( $1 \leq N \leq 100\,000$ ) и количество различных цветов, в которые могут быть раскрашены кубики —  $M$  ( $1 \leq M \leq 100\,000$ ). Следующая строка содержит  $N$  целых чисел от 1 до  $M$  — цвета кубиков.

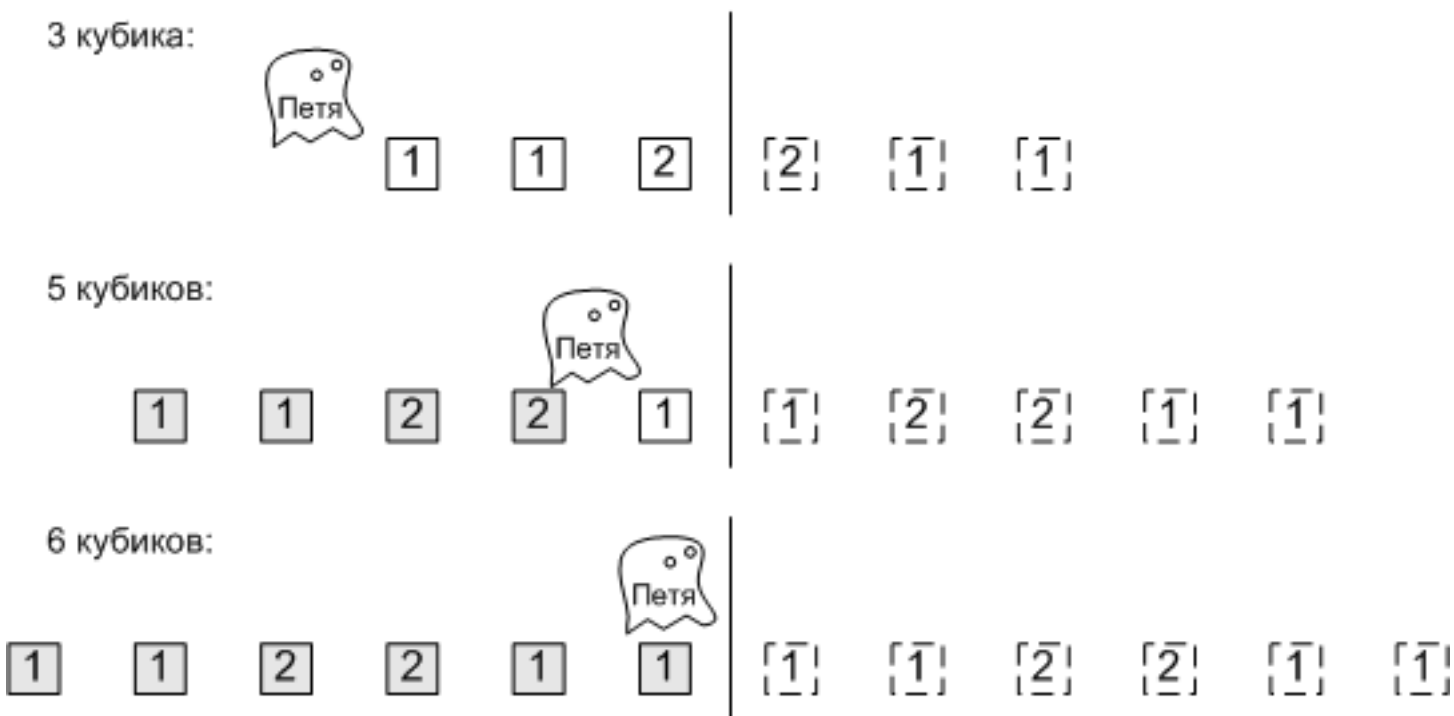
#### Формат выходных данных

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.



## Бонус

### Задача 25Н. Архиватор [0.8 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:

- «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 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

### Задача 251. Две строки [0.3 sec, 256 mb]

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

2strings.in	2strings.out
ababa bab	ab

### Задача 25J. Word Cover [0.3 sec, 256 mb]

Говорят, что строка  $\alpha$  покрывает строку  $\beta$ , если для каждой позиции в строке  $\beta$  существует вхождение  $\alpha$ , как подстроки  $\beta$ , содержащее эту позицию. Например, строка “aba” покрывает строку “abaabaababa”, но не покрывает строку “baba”. Конечно, строка покрывает саму себя. Компактностью строки  $\beta$  назовем длину самой короткой строки, которая покрывает  $\beta$ .

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 25К. Преобразование строковых функций [0.3 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:  $\pi[i] = \max\{k | 0 \leq k < i, S[1..k] = S[i-k+1..i]\}$ . 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  $N$  ( $1 \leq N \leq 200\,000$ ), 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	0 0 1 0 1 2 3 1
8 0 1 0 3 0 1 1	

**Задача 25L. Преобразование функций: обратная задача [0.3 сек, 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:  $\pi[i] = \max\{k | 0 \leq k < i, S[1..k] = S[i-k+1..i]\}$ . 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  $N$  ( $1 \leq N \leq 200\,000$ ), 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	8 0 1 0 3 0 1 1
0 0 1 0 1 2 3 1	