

## Содержание

<b>Easy</b>	<b>2</b>
Задача А. Подстрока [2 sec, 256 mb]	2
Задача В. Дана строка [2 sec, 256 mb]	3
Задача С. Основание строки [2 sec, 256 mb]	4
<b>Medium</b>	<b>5</b>
Задача D. Суффиксный массив [2 sec, 256 mb]	5
Задача Е. Двухкратная подстрока [2 sec, 256 mb]	6
Задача F. Суффиксы [2 sec, 256 mb]	7
Задача G. От префикс-функции к z-функции [2 sec, 256 mb]	8
Задача H. От z-функции к префикс-функции [2 sec, 256 mb]	9

---

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

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

**Обратите внимание**, что ввод-вывод во всех задачах стандартный.

## Easy

### Задача А. Подстрока [2 sec, 256 mb]

Дана строка. Найдите её подстроку, которая встречается в этой строке как можно больше раз.

#### Формат входных данных

В первой строке входного файла записана непустая строка  $S$  длиной не более 100 000 символов, состоящая из маленьких букв латинского алфавита.

#### Формат выходных данных

Выведите непустую подстроку  $P$  данной строки  $S$ , встречающуюся в  $S$  как можно больше раз. Если оптимальных ответов несколько, выведите любой.

#### Примеры

stdin	stdout
a	a
abc	ab

**Задача В. Дана строка [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.

**Эту задачу нужно сдать при помощи z-функции**

**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**

stdin	stdout
abacaba	2
aba	1 5

### Задача С. Основание строки [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  $S$ .

Эту задачу нужно сдать при помощи prefix-функции

#### 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

stdin	stdout
zzz	1
bcabcab	3

## Medium

### Задача D. Суффиксный массив [2 сек, 256 mb]

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

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

### Задача Е. Двухкратная подстрока [2 sec, 256 mb]

Дана строка  $S$  длины  $n$  и число  $k$ . Найдите в строке  $S$  такую подстроку длины  $k$ , которая встречается в ней по крайней мере два раза, или выясните, что такой подстроки нет.

#### Формат входных данных

В первой строке входного файла задана строка  $S$ ; её длина  $n$  — от 1 до 100 000 символов, включительно. Во второй строке задано целое число  $k$  ( $1 \leq k \leq n$ ). Строка состоит только из маленьких букв латинского алфавита.

#### Формат выходных данных

Если подстроки длины  $k$ , встречающейся хотя бы два раза, не существует, выведите слово «NONE» в первой строке выходного файла. В противном случае выведите любую из таких подстрок в первой строке выходного файла.

#### Примеры

stdin	stdout
ast 1	NONE
blinkingblueblogger 2	in
aaaaaab 5	aaaaa

### Задача F. Суффиксы [2 сек, 256 mb]

Назовём *строкой* последовательность из маленьких букв английского алфавита. Строкой, например, является пустая последовательность «», слово «aabaf» или бесконечная последовательность букв «a».

Определим  $i$ -й *суффикс*  $S_i$  строки  $S$ : это просто строка  $S$ , из которой вырезаны первые  $i$  букв. Так, для строки  $S = \text{«aabaf»}$  суффиксы будут такими:

$$\begin{aligned} S_0 &= \text{«aabaf»} \\ S_1 &= \text{«abaf»} \\ S_2 &= \text{«baf»} \\ S_3 &= \text{«af»} \\ S_4 &= \text{«f»} \\ S_5 = S_6 = S_7 = \dots &= \text{«»} \end{aligned}$$

Суффиксы определены для всех  $i \geq 0$ .

*Циклическое расширение*  $S^*$  конечной строки  $S$  — это строка, полученная приписыванием её к самой себе бесконечное количество раз. Так,

$$\begin{aligned} S^* = S_0^* &= \text{«aabafaabafaa\dots»} \\ S_1^* &= \text{«abafabafabaf\dots»} \\ S_2^* &= \text{«bafbafbafbaf\dots»} \\ S_3^* &= \text{«afafafafafaf\dots»} \\ S_4^* &= \text{«ffffffffffff\dots»} \\ S_5^* = S_6^* = S_7^* = \dots &= \text{«»} \end{aligned}$$

По данной строке  $S$  выясните, сколько её суффиксов  $S_i$  имеют такое же циклическое расширение, как и сама строка  $S$ , то есть количество таких  $i$ , что  $S^* = S_i^*$ .

#### Формат входных данных

В первой и единственной строке входного файла задана строка  $S$ , состоящая из не менее, чем одной, и не более, чем 100 000 маленьких латинских букв «a–z».

#### Формат выходных данных

Выведите в первую строку выходного файла одно число — количество суффиксов строки  $S$ , имеющих такое же циклическое расширение, как и она сама.

#### Примеры

stdin	stdout
aa	2
ab	1
qqqq	4
xyzzxy	1

**Задача G. От префикс-функции к z-функции [2 sec, 256 mb]**

*Префикс-функция*  $p(i)$  для строки  $s = s_1s_2 \dots s_n$  определяется от позиции  $i$  ( $1 \leq i \leq n$ ) в строке так:  $p(i)$  — это максимальная длина собственного префикса строки  $s_1s_2 \dots s_i$ , равного её собственному суффиксу. Напомним, что *собственный префикс* строки  $s = s_1s_2 \dots s_n$  — это строка  $s_1s_2 \dots s_r$  для некоторого  $r < n$ . Аналогично, *собственный суффикс* строки  $s = s_1s_2 \dots s_n$  — это строка  $s_ls_2 \dots s_n$  для некоторого  $l > 1$ .

*Z-функция*  $z(i)$  для строки  $s = s_1s_2 \dots s_n$  определяется от позиции  $i$  ( $1 \leq i \leq n$ ) в строке так:  $z(1) = 0$ , а для  $i > 1$   $z(i)$  — это максимальное число такое, что строки  $s_1s_2 \dots s_{z(i)}$  и  $s_is_{i+1} \dots s_{i+z(i)-1}$  совпадают.

Даны длина строки  $n$  и значения префикс-функции  $p(1), p(2), \dots, p(n)$  для этой строки. Найдите для этой строки значения z-функции  $z(1), z(2), \dots, z(n)$ .

**Формат входных данных**

В первой строчке входного файла задано целое число  $n$  ( $1 \leq n \leq 1\,000\,000$ ). Во второй строчке заданы  $n$  чисел через пробел — значения префикс-функции  $p(1), p(2), \dots, p(n)$ . Гарантируется, что существует строка длины  $n$ , состоящая из строчных букв латинского алфавита, для которой префикс-функция от позиций  $1, 2, \dots, n$  принимает данные значения.

**Формат выходных данных**

В первой строчке выходного файла выведите  $n$  чисел через пробел — значения z-функции для строки, имеющей данную префикс-функцию.

**Примеры**

stdin	stdout
6 0 0 1 2 3 4	0 0 4 0 2 0
7 0 0 0 1 2 3 4	0 0 0 4 0 0 1
4 0 0 0 0	0 0 0 0



**Задача Н. От z-функции к префикс-функции [2 сек, 256 mb]**

*Z-функция*  $z(i)$  для строки  $s = s_1s_2 \dots s_n$  определяется от позиции  $i$  ( $1 \leq i \leq n$ ) в строке так:  $z(1) = 0$ , а для  $i > 1$   $z(i)$  — это максимальное число такое, что строки  $s_1s_2 \dots s_{z(i)}$  и  $s_is_{i+1} \dots s_{i+z(i)-1}$  совпадают.

*Префикс-функция*  $p(i)$  для строки  $s = s_1s_2 \dots s_n$  определяется от позиции  $i$  ( $1 \leq i \leq n$ ) в строке так:  $p(i)$  — это максимальная длина собственного префикса строки  $s_1s_2 \dots s_i$ , равного её собственному суффиксу. Напомним, что *собственный префикс* строки  $s = s_1s_2 \dots s_n$  — это строка  $s_1s_2 \dots s_r$  для некоторого  $r < n$ . Аналогично, *собственный суффикс* строки  $s = s_1s_2 \dots s_n$  — это строка  $s_ls_2 \dots s_n$  для некоторого  $l > 1$ .

Даны длина строки  $n$  и значения z-функции  $z(1), z(2), \dots, z(n)$  для этой строки. Найдите для этой строки значения префикс-функции  $p(1), p(2), \dots, p(n)$ .

**Формат входных данных**

В первой строчке входного файла задано целое число  $n$  ( $1 \leq n \leq 1\,000\,000$ ). Во второй строчке заданы  $n$  чисел через пробел — значения z-функции  $z(1), z(2), \dots, z(n)$ . Гарантируется, что существует строка длины  $n$ , состоящая из строчных букв латинского алфавита, для которой z-функция от позиций  $1, 2, \dots, n$  принимает данные значения.

**Формат выходных данных**

В первой строчке выходного файла выведите  $n$  чисел через пробел — значения префикс-функции для строки, имеющей данную z-функцию.

**Примеры**

stdin	stdout
6 0 0 4 0 2 0	0 0 1 2 3 4
7 0 0 0 4 0 0 1	0 0 0 1 2 3 4
4 0 0 0 0	0 0 0 0