

## Задача А. Z-функция

Имя входного файла: **z-function.in**  
Имя выходного файла: **z-function.out**  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 64 мегабайта

Дана непустая строка  $S$ , длина которой  $N$  не превышает  $10^6$ . Будем считать, что элементы строки нумеруются от 1 до  $N$ .

Требуется для всех  $i$  от 1 до  $N$  вычислить её z-функцию  $z[i]$ .

### Формат входных данных

Одна строка длины  $N$ ,  $0 < N \leq 10^6$ , состоящая из маленьких латинских букв.

### Формат выходных данных

Выведите  $N$  чисел — значения z-функции для каждой позиции, разделённые пробелом.

### Примеры

| <b>z-function.in</b> | <b>z-function.out</b>  |
|----------------------|------------------------|
| abra                 | 11 0 0 1 0 1 0 4 0 0 1 |

## Задача В. Префикс-функция

Имя входного файла: `prefix-function.in`  
Имя выходного файла: `prefix-function.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 64 мегабайта

Дана непустая строка  $S$ , длина которой  $N$  не превышает  $10^6$ . Будем считать, что элементы строки нумеруются от 1 до  $N$ .

Требуется для всех  $i$  от 1 до  $N$  вычислить её префикс-функцию  $\pi[i]$ .

### Формат входных данных

Одна строка длины  $N$ ,  $0 < N \leq 10^6$ , состоящая из маленьких латинских букв.

### Формат выходных данных

Выведите  $N$  чисел — значения префикс-функции для каждой позиции, разделённые пробелом.

### Примеры

| <code>prefix-function.in</code> | <code>prefix-function.out</code> |
|---------------------------------|----------------------------------|
| <code>abracadabra</code>        | 0 0 0 1 0 1 0 1 2 3 4            |

## Задача С. Много строк

Имя входного файла: `musubstr.in`  
Имя выходного файла: `musubstr.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Даны  $K$  строк из маленьких латинских букв. Требуется найти 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

| <code>musubstr.in</code>                   | <code>musubstr.out</code> |
|--|---------------------------|
| 3<br>abacaba<br>mycabarchive<br>acabistrue | cab                       |

## Задача D. Основание строки (Z функция)

Имя входного файла: **basis.in**  
Имя выходного файла: **basis.out**  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

**Решите эту задачу Z функцией.**

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 непустая строка, которая содержит только латинские буквы, длина строки не превышает 50 000 символов.

### Формат выходных данных

В выходной файл выведите ответ на задачу.

### Примеры

| <b>basis.in</b> | <b>basis.out</b> |
|-----------------|------------------|
| zzz             | 1                |
| bcabcab         | 3                |

## Задача Е. Основание строки (Префикс функция)

Имя входного файла: **basis.in**  
Имя выходного файла: **basis.out**  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

**Решите эту задачу Префикс функцией.**

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 непустая строка, которая содержит только латинские буквы, длина строки не превышает 50 000 символов.

### Формат выходных данных

В выходной файл выведите ответ на задачу.

### Примеры

| <b>basis.in</b> | <b>basis.out</b> |
|-----------------|------------------|
| zzz             | 1                |
| bcabcab         | 3                |

## Задача F. Аполлинария

Имя входного файла: apollinaria.in  
Имя выходного файла: apollinaria.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Паша завтра читает лекцию по базовым алгоритмам на строках. В частности, ему придется рассказать ЛКШатам про  $Z$ -функцию. Сам он когда-то знал, что это такое, но с тех пор успел многое забыть.

Готовясь к лекции, он открыл <http://ru.wikipedia.org/wiki/Z-функция> и даже [http://e-maxx.ru/algo/z\\_function](http://e-maxx.ru/algo/z_function). Теперь Паша вспомнил, что  $Z$ -функция от строки  $S$  – это массив  $z$ , каждый элемент которого  $z_i$  равен длинейшему префиксу подстроки, начинающейся с позиции  $i$  в строке  $S$ .

Так, например, для строки `aabcaabxaaaaz`  $Z$ -функция это  $[12, 1, 0, 0, 3, 1, 0, 0, 2, 2, 1, 0]$ . Ее суммарное значение равно 22.

В группе у Паши учится девочка Аполлинария, которая любит задавать каверзные вопросы и, чтобы не попасть впросак, Паша прорешивает различные задачи по теме лекции. Сейчас он задумался над очередной. В заданной строке требуется переставить буквы и сделать так, чтобы суммарное значение  $Z$ -функции было минимальным.

### Формат входных данных

В входном файле содержится непустая строка, состоящая из строчных латинских букв. Гарантируется, что каждая буква встречается не более 1000 раз.

### Формат выходных данных

В выходной файл выведите строку состоящую из тех же символов, что и входная строка. Причем суммарное значение  $Z$ -функции для выходной строки должно быть минимальным. Если ответов несколько, выведите **минимальную** лексикографически строку.

### Примеры

| apollinaria.in | apollinaria.out |
|----------------|-----------------|
| apollinaria    | naaaillopr      |

## Задача G. Обратная префикс-функция

Имя входного файла: **inverse.in**  
Имя выходного файла: **inverse.out**  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

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

Ваша задача — по заданной префикс-функции восстановить строку. В качестве символов строки разрешается использование  $M$  первых строчных букв латинского алфавита.

### Формат входных данных

Входной файл состоит из одного или более набора входных данных.

В первой строке каждого набора записаны два целых числа  $N, M$  ( $N \geq 1, 1 \leq M \leq 26$ ). Во второй строке записана последовательность целых чисел  $\pi[1], \pi[2], \dots, \pi[N]$ . Все числа в последовательности целые неотрицательные, не превосходящие  $10^6$ .

Сумма значений  $N$  по всем наборам не превосходит  $10^6$ , количество наборов входных данных не превосходит  $10^5$ .

### Формат входных данных

Входной файл состоит из одного или более набора входных данных.

В первой строке каждого набора записаны два целых числа  $N, M$  ( $N \geq 1, 1 \leq M \leq 26$ ). Во второй строке записана последовательность целых чисел  $\pi[1], \pi[2], \dots, \pi[N]$ . Все числа в последовательности целые неотрицательные, не превосходящие  $10^6$ .

Сумма значений  $N$  по всем наборам не превосходит  $10^6$ , количество наборов входных данных не превосходит  $10^5$ .

### Формат выходных данных

Выведите в первую строку выходного файла YES, если существует искомое слово, и NO в противном случае. В случае положительного ответа выведите во вторую строку выходного файла выведите искомое слово. Если решений несколько, выведите любое.

### Примеры

| inverse.in                          | inverse.out           |
|-------------------------------------|-----------------------|
| 8 3<br>0 0 1 0 1 2 3 1<br>1 1<br>10 | YES<br>abacabaa<br>NO |

## Задача Н. Период строки

Имя входного файла: `period.in`  
Имя выходного файла: `period.out`  
Ограничение по времени: 0.5 секунда  
Ограничение по памяти: 256 мегабайт

Дана строка  $s$ . Требуется найти минимальную по длине строку  $t$ , такую что  $s$  представима в виде конкатенации одной или нескольких строк  $t$ .

### Формат входных данных

Первая строка входного файла содержит  $s$  ( $1 \leq |s| \leq 5 \cdot 10^6$ ). Стока состоит из букв латинского алфавита.

### Формат выходных данных

Выведите длину искомой строки  $t$ .

### Примеры

| <code>period.in</code> | <code>period.out</code> |
|------------------------|-------------------------|
| <code>abcabcabc</code> | 3                       |

## Задача I. Неточное совпадение

Имя входного файла: `inexact-matching.in`  
Имя выходного файла: `inexact-matching.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Даны строки  $p$  и  $t$ . Требуется найти все вхождения строки  $p$  в строку  $t$  в качестве подстроки с точностью до возможного несовпадения одного символа.

### Формат входных данных

Первая строка входного файла содержит  $p$ , вторая —  $t$  ( $1 \leq |p|, |t| \leq 10^6$ ). Строки состоят из букв латинского алфавита.

### Формат выходных данных

В первой строке выведите количество вхождений строки  $p$  в строку  $t$ . Во второй строке выведите в возрастающем порядке номера символов строки  $t$ , с которых начинаются вхождения  $p$ . Символы нумеруются с единицы.

### Примеры

| <code>inexact-matching.in</code> | <code>inexact-matching.out</code> |
|----------------------------------|-----------------------------------|
| aaaa                             | 4                                 |
| Caaabdaaaa                       | 1 2 6 7                           |

## Задача J. Преобразование строковых функций

Имя входного файла: **trans.in**  
Имя выходного файла: **trans.out**  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Одной умной лягушке приснилась какая-то функция от строки. На утро она вспомнила и записала определение: 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  $[8, 0, 1, 0, 3, 0, 1, 1]$ .

На следующей день ей приснилось другая функция и опять же на утро записала определение: 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:  $[0, 0, 1, 0, 1, 2, 3, 1]$ .

Позже ее заинтересовало есть ли связь между ее снами, вдруг она оракул, поэтому она пытается из первой функции построить вторую. Помогите ей в этом.

### 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.

### Примеры

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

## Задача К. Преобразование строковых функций: обратная задача

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Для строки  $S$  определим  $Z$ -функцию следующим образом:  $Z[i] = \text{lcp}(S, S[i..|S|])$ , где  $\text{lcp}(S_1, S_2)$  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

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 8                | 8 0 1 0 3 0 1 1   |
| 0 0 1 0 1 2 3 1  |                   |