

## Задача А. 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>	<code>0 0 0 1 0 1 0 1 2 3 4</code>

## Задача В. А-функция

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

Дана строка  $S$ , состоящая из  $N$  символов. Определим функцию  $A(i)$  от первых  $i$  символов этой строки следующим образом:

$A(i)$  = максимально возможному  $k$  такому, что равны следующие две строки:  $S[1] + S[2] + S[3] + \dots + S[k] = S[i] + S[i - 1] + S[i - 2] + \dots + S[i - k + 1]$ , где  $S[i]$  —  $i$ -ый символ строки  $S$ , а знак  $+$  означает, что символы записываются в строчку непосредственно друг за другом.

Напишите программу, которая вычислит значения функции  $A$  для заданной строки для всех возможных значений  $i$  от 1 до  $N$ .

### Формат входных данных

В первой строке входного файла записано одно число  $N$ .  $1 \leq N \leq 200\,000$ . Во второй строке записана строка длиной  $N$  символов, состоящая только из больших и/или маленьких латинских букв.

### Формат выходных данных

В выходной файл выведите  $N$  чисел — значения функции  $A(1), A(2), \dots, A(N)$ .

### Примеры

<code>afunction.in</code>	<code>afunction.out</code>
5 aabaa	1 2 0 1 5

## Задача С. 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  $s$  ( $1 \leq |s| \leq 5 \cdot 10^6$ ). Строка 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>	<code>3</code>

## Задача D. Циклическая строка

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

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

На вход программы поступает строка, которая содержит только латинские буквы, длина строки не превышает 50 000 символов.

### Формат выходных данных

Требуется вывести одно число — ответ на вопрос задачи.

### Примеры

стандартный ввод	стандартный вывод
z	1
сac	2

## Задача Е. Палиндромы

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

Строка называется палиндромом, если она одинаково читается как слева направо, так и справа налево. Например, `abba` — палиндром, а `омах` — нет. Для строки  $\alpha$  будем обозначать  $\alpha[i..j]$  ее подстроку длины  $j - i + 1$  с  $i$ -й по  $j$ -ю позицию включительно (позиции нумеруются с единицу). Для заданной строки  $\alpha$  длины  $N$  ( $1 \leq N \leq 100\,000$ ) требуется подсчитать число  $q$  пар  $(i, j)$ ,  $1 \leq i < j \leq n$ , таких что  $\alpha[i..j]$  является палиндромом.

### Формат входных данных

Входной файл содержит одну строку  $\alpha$  длины  $N$ , состоящую из маленьких латинских букв.

### Формат выходных данных

В выходной файл выведите искомое число  $q$ .

### Примеры

<code>palindrome.in</code>	<code>palindrome.out</code>
<code>aaa</code>	3
<code>abba</code>	2
<code>омах</code>	0

## Задача F. Последнее слово Джека

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

Джек недавно прочитал на заборе интересное и новое для него слово. Оно настолько понравилось Джеку, что он захотел сам придумать ещё какое-нибудь интересное слово. Но только ничего у него не вышло — все придуманные им слова состояли из префиксов исходного слова и поэтому не приносили радости. Он стал придумывать всё более и более длинные слова, но ни одно из них не было оригинальным. . .

И вот настало время Джеку сказать своё последнее слово.

### Формат входных данных

Первая строка содержит интересное слово, которое было написано на заборе. Вторая строка содержит последнее слово Джека. Длины слов не превосходят 75 000, слова непустые и состоят из строчных латинских букв.

### Формат выходных данных

Если Джек так ничего и не придумал своего, выведите первой строкой `No`. В этом случае покажите Джеку, как разбить его последнее слово на несколько частей, каждая из которых является исходным словом или его непустым префиксом — выведите все эти части во второй строке, разделяя их пробелом. Если же такого разбиения нет, и последнее слово было за Джеком, выведите единственной строкой `Yes`.

### Примеры

<code>prefix.in</code>	<code>prefix.out</code>
<code>abracadabra abrabracada</code>	<code>No abr abracada</code>
<code>abracadabra arbadacarba</code>	<code>Yes</code>

## Задача G. 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
Caabdaaaa	1 2 6 7

## Задача Н. Шок контент

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

Вам даны две последовательности длины  $N$ :  $a = \{a_0, \dots, a_{N-1}\}$  и  $b = \{b_0, \dots, b_{N-1}\}$ , состоящие из целых положительных чисел.

Ёж выберет целое число  $k$ , такое что  $0 \leq k < N$  и целое положительное число  $x$ . После этого он создаст новую последовательность длины  $N$   $a' = \{a'_0, \dots, a'_{N-1}\}$  так, что  $a'_i = a_{(i+k) \bmod N} \oplus x$ .

Найдите все пары  $(k, x)$  такие, что  $a'$  будет равняться  $b$ .

### Формат входных данных

Первая строка входных данных содержит целое число  $N$  ( $1 \leq N \leq 2 \cdot 10^5$ ) — количество элементов в последовательностях.

Вторая строка содержит  $N$  целых чисел  $a_i$  ( $0 \leq a_i \leq 2^{30}$ ) — элементы последовательности  $a$ .

Третья строка содержит  $N$  целых чисел  $b_i$  ( $0 \leq b_i \leq 2^{30}$ ) — элементы последовательности  $b$ .

### Формат выходных данных

Выведите все пары  $(k, x)$ , что  $a'$  будет равняться  $b$ .

На каждой строке вывода должна содержаться ровно 1 пара.

Пары должны идти в порядке возрастания  $k$  (в случае равенства  $k$  в порядке возрастания  $x$ )

Если подходящих пар не существует, то вывод должен быть пустым.

### Примеры

стандартный ввод	стандартный вывод
3 0 2 1 1 2 3	1 3
5 0 0 0 0 0 2 2 2 2 2	0 2 1 2 2 2 3 2 4 2
6 0 1 3 7 6 4 1 5 4 6 2 3	2 2 5 5
2 1 2 0 0	

## Задача I. К-я строка

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

Реализуйте структуру данных, которая поддерживает следующие операции:

- добавить в словарь строку  $S$ ;
- найти в словаре  $k$ -ю строку в лексикографическом порядке.

Известно, что изначально словарь пуст.

### Формат входных данных

Первая строка входного файла содержит натуральное число  $N$  — количество команд ( $N \leq 10^5$ ). Последующие  $N$  строк содержат по одной команде каждая.

Команда записывается либо в виде числа  $k$ , либо в виде строки  $S$ , которая может состоять только из строчных латинских букв. Гарантируется, что при запросе  $k$ -й строки она существует. Также гарантируется, что сумма длин всех добавляемых строк не превышает  $10^5$ .

### Формат выходных данных

Для каждого числового запроса  $k$  выходной файл должен содержать  $k$ -ю в лексикографическом порядке строчку из словаря на момент запроса. Гарантируется, что суммарная длина строк в выходном файле не превышает  $10^5$ .

### Примеры

<code>kthstr.in</code>	<code>kthstr.out</code>
7	tolstoy
pushkin	gogol
lermontov	
tolstoy	
gogol	
gorkiy	
5	
1	

## Задача J. Мультимножество Василия

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

У автора уже закончились истории про Василия, поэтому он просто написал формальную постановку задачи.

У вас есть  $q$  запросов и мультимножество  $A$ , изначально содержащее только число 0. Запросы бывают трёх видов:

- «+  $x$ » — добавить в мультимножество  $A$  число  $x$ .
- «-  $x$ » — удалить одно вхождение числа  $x$  из мультимножества  $A$ . Гарантируется, что хотя бы одно число  $x$  в этот момент присутствует в мультимножестве.
- «?  $x$ » — вам даётся число  $x$ , требуется вычислить  $\max_{y \in A} x \oplus y$ , то есть максимальное значение побитового исключающего ИЛИ (также известно как XOR) числа  $x$  и какого-нибудь числа  $y$  из мультимножества  $A$ .

Мультимножество — это множество, в котором разрешается несколько одинаковых элементов.

### Формат входных данных

В первой строке входных данных содержится число  $q$  ( $1 \leq q \leq 200\,000$ ) — количество запросов, которые требуется обработать Василию.

Каждая из последующих  $q$  строк входных данных содержит один трёх символов «+», «-» или «?» и число  $x_i$  ( $1 \leq x_i \leq 10^9$ ). Гарантируется, что во входных данных встречается хотя бы один запрос «?».

Обратите внимание, что число 0 всегда будет присутствовать в мультимножестве.

### Формат выходных данных

На каждый запрос типа «?» выведите единственное целое число — максимальное значение побитового исключающего ИЛИ для числа  $x_i$  и какого-либо числа из мультимножества  $A$ .

### Примеры

стандартный ввод	стандартный вывод
10	11
+ 8	10
+ 9	14
+ 11	13
+ 6	
+ 1	
? 3	
- 8	
? 3	
? 8	
? 11	

### Замечание

После первых пяти операций в мультимножестве  $A$  содержатся числа 0, 8, 9, 11, 6 и 1.

Ответом на шестой запрос будет число  $11 = 3 \oplus 8$  максимальное из чисел  $3 \oplus 0 = 3$ ,  $3 \oplus 9 = 10$ ,  $3 \oplus 11 = 8$ ,  $3 \oplus 6 = 5$  и  $3 \oplus 1 = 2$ .

## Задача К. Цепочка слов

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.25 секунд  
Ограничение по памяти: 256 мегабайт

Цепочкой слов длины  $n$  назовем последовательность слов  $w_1, w_2, \dots, w_n$  такую, что для  $1 \leq i < n$  слово  $w_i$  является собственным префиксом слова  $w_{i+1}$ .

Напомним, что слово  $u$  длины  $k$  называется собственным префиксом слова  $v$  длины  $l$ , если  $l > k$  и первые  $k$  букв слова  $v$  совпадают со словом  $u$ .

Задано множество слов  $S = \{s_1, s_2, \dots, s_m\}$ . Найдите максимальную длину цепочки слов, которую можно построить, используя (возможно, не все) слова этого множества.

### Формат входных данных

Первая строка ввода содержит целое число  $m$  ( $1 \leq m \leq 255$ ). Каждая из последующих  $m$  строк содержит по одному слову из множества  $S$ .

Все слова не пусты, имеют длину, не превосходящую 255 символов, и состоят только из строчных букв латинского алфавита.

### Формат выходных данных

Выведите ответ на задачу.

### Примеры

стандартный ввод	стандартный вывод
3 a ab abc	3
5 a ab bc bcd add	2