

## Задача А. 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 В. 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 С. Максимальная общая подстрока

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Дана строка  $A = a_1a_2 \dots a_n$  и строка  $B = b_1b_2 \dots b_m$ . Нужно узнать длину их максимальной общей подстроки.

### Формат входных данных

В первых двух строках записаны строки  $A$  и  $B$ , состоящие из строчных латинских букв. Эти строки непустые и имеют длину не более 30 000 символов.

### Формат выходных данных

В выходной файл выведите длину их максимальной общей подстроки.

### Примеры

стандартный ввод	стандартный вывод
abacaba acabaca	5

## Задача D. Сравнение подстрок

Имя входного файла: `substrcmp.in`  
 Имя выходного файла: `substrcmp.out`  
 Ограничение по времени: 1.5 секунд  
 Ограничение по памяти: 256 мегабайт

Дана строка  $S = s_1 s_2 \dots s_n$  и множество запросов вида  $(l_1, r_1, l_2, r_2)$ . Для каждого запроса требуется ответить, какая из подстрок больше —  $s_{l_1} \dots s_{r_1}$  или  $s_{l_2} \dots s_{r_2}$ .

### Формат входных данных

В первой строке записана строка  $S$ , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число  $q$  ( $1 \leq q \leq 100\,000$ ) — количество запросов. В каждой из следующих  $q$  строк записаны числа  $l_1, r_1, l_2, r_2$  ( $1 \leq l_1 \leq r_1 \leq |S|$ ;  $1 \leq l_2 \leq r_2 \leq |S|$ ).

### Формат выходных данных

Для каждого запроса выведите «=», если соответствующие подстроки равны, «>», если первая подстрока больше и «<», если первая подстрока меньше.

### Примеры

substrcmp.in	substrcmp.out
abacaba	=
3	<
1 3 5 7	>
1 3 3 5	
4 7 2 5	
ab	<
2	<
1 1 2 2	
1 1 1 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

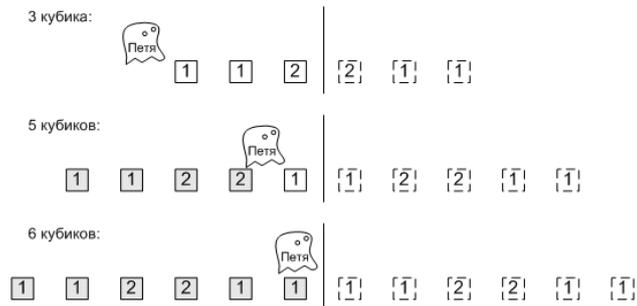
<code>palindrome.in</code>	<code>palindrome.out</code>
<code>aaa</code>	3
<code>abba</code>	2
<code>омах</code>	0

## Задача F. Кубики

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

Привидение Петя любит играть со своими кубиками. Он любит выкладывать их в ряд и разглядывать своё 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 у него может быть кубиков. 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 два целых числа:  $N$  ( $1 \leq N \leq 100\,000$ ) 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

В выходной файл выведите в порядке возрастания все такие  $K$ , что у Пети может быть  $K$  кубиков.

### Примеры

<code>cubes.in</code>	<code>cubes.out</code>
6 2 1 1 2 2 1 1	3 5 6

## Задача G. 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 Н. 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>
<code>aaaa</code>	<code>4</code>
<code>Caabdaaaa</code>	<code>1 2 6 7</code>

## Задача I. 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	