

## Задача А. Данна строка

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

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 В. Сравнения подстрок

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

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
trolo	Yes
3	Yes
1 7 1 7	No
3 5 5 7	
1 1 1 5	

## Задача С. Преобразование строковых функций

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

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  $S = abacaba$   $Z$ -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  $S = abacaba$  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

Выполните  $N$  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 D. Преобразование строковых функций: обратная задача

Имя входного файла: `invtrans.in`Имя выходного файла: `invtrans.out`

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

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

<code>invtrans.in</code>	<code>invtrans.out</code>
8	8 0 1 0 3 0 1 1
0 0 1 0 1 2 3 1	

## Задача E. Кубики

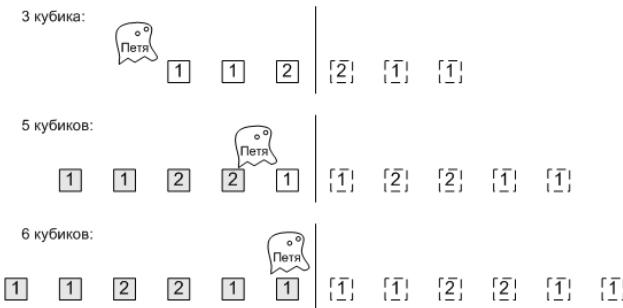
Имя входного файла: `cubes.in`Имя выходного файла: `cubes.out`

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 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 Петя! 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

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

## Задача F. Word Cover

Имя входного файла: `cover.in`Имя выходного файла: `cover.out`

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

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 “babab”. 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
abaabaaababa	1 2 3 4 5 3 4 5 3 10 3

## Задача G. Подстроки

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

Дана строка  $s$ . Вам требуется подсчитать количество её различных подстрок.  
Пустую строку учитывать не следует.

### Формат входных данных

В единственной строке входного файла содержится данная строка  $s$ , состоящая из строчных латинских букв. Длина строки не превосходит 4 000 символов.

### Формат выходных данных

В единственной строке выходного файла выведите единственное число — количество различных подстрок  $s$ .

substr.in	substr.out
aaaa	4
abacaba	21

## Задача H. Incrementator

Имя входного файла: incrementator.in  
Имя выходного файла: incrementator.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мебибайт

Ваша задача — написать программу, моделирующую простое устройство, которое умеет прибавлять целые значения к целочисленным переменным.

### Формат входных данных

Входной файл состоит из одной или нескольких строк, описывающих операции. Стока состоят из названия переменной и числа, которое к этой переменной надо добавить. Все числа не превосходят 100 по абсолютной величине. Изначально все

переменные равны нулю. Названия переменных состоят из не более чем 100 000 маленьких латинских букв. Размер входного файла не превосходит 2 мегабайта.

### Формат выходных данных

Для каждой операции выведите на отдельной строке значение соответствующей переменной после выполнения операции.

### Примеры

incrementator.in	incrementator.out
a 2	2
b 3	3
a -1	1
c 4	4
b 17	20
xyz 23	23