

## 8Е. А-функция от строки

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 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  $\{k\}$ , что равны следующие строки:

$S_1 + S_2 + \dots + S_k$  и  $S_i + S_{i-1} + \dots + S_{i-k+1}$ , где  $S_i$  —  $\{i\}$ -ый символ строки  $\{S\}$ , а знак  $+$  означает, что символы записываются в строку непосредственно друг за другом.

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

### Входные данные

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.

### Выходные данные

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

### Пример

входные данные

Скопировать

5  
aaba

выходные данные

Скопировать

1 2 0 1 5

## 9A. Мультимножество Василия

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 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.

### Входные данные

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.

### Выходные данные

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

### Пример

входные данные	Скопировать
10 + 8 + 9 + 11 + 6 + 1 ? 3 - 8 ? 3 ? 8 ? 11	
выходные данные	Скопировать
11 10 14 13	

### Примечание

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  $3 \oplus 1 = 2$ .

## 10A. Суффиксный массив

ограничение по времени на тест: 2 s.

ограничение по памяти на тест: 256 MB

Постройте суффиксный массив для заданной строки  $s$ .

### Входные данные

Первая строка входного файла содержит строку  $s$  ( $1 \leq |s| \leq 400\,000$ ). Строка состоит из строчных латинских букв.

### Выходные данные

Выведите  $|s|$  различных чисел — номера первых символов суффиксов строки  $s$  так, чтобы соответствующие суффиксы были упорядочены в лексикографически возрастающем порядке.

### Пример

<b>входные данные</b>	<a href="#">Скопировать</a>
ababb	
<b>выходные данные</b>	<a href="#">Скопировать</a>
1 3 5 2 4	

## 10B. Циклические сдвиги

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

$k$ -м циклическим сдвигом строки  $S$  называется строка, полученная перестановкой  $k$  первых символов строки  $S$  в конец строки.

Рассмотрим все различные циклические сдвиги строки  $S$  и отсортируем их по возрастанию. Требуется вычислить  $i$ -ю строчку этого массива.

Например, для строки `abacabac` существует четыре различных циклических сдвига: нулевой (`abacabac`), первый (`bacabaca`), второй (`acabacab`) и третий (`cabacaba`). После сортировки по возрастанию получится такой массив: `abacabac`, `acabacab`, `bacabaca`, `cabacaba`.

### Входные данные

В первой строке входного файла записана строка  $S$ , длиной не более **100 000** символов с ASCII-кодами от 32 до 126. Во второй строке содержится единственное целое число  $k$  ( $1 \leq k \leq 100\,000$ ).

### Выходные данные

В выходной файл выведите  $k$ -й по возрастанию циклический сдвиг строки  $S$ , или слово `IMPOSSIBLE`, если такого сдвига не существует.

### Пример

**входные данные**

Скопировать

`abacabac`  
`4`

**выходные данные**

Скопировать

`cabacaba`

## 10D. Рефрен

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Рассмотрим последовательность  $n$  целых чисел от  $1$  до  $m$ . Подпоследовательность подряд идущих чисел называется рефреном, если произведение ее длины на количество вхождений в последовательность максимально.

По заданной последовательности требуется найти ее рефрен.

### Входные данные

Первая строка входного файла содержит два целых числа:  $n$  и  $m$  ( $1 \leq n \leq 150\,000$ ,  $1 \leq m \leq 10$ ).

Вторая строка содержит  $n$  целых чисел от  $1$  до  $m$ .

### Выходные данные

Первая строка выходного файла должна содержать произведение длины рефрена на количество ее вхождений. Вторая строка должна содержать длину рефрена. Третья строка должна содержать последовательность которая является рефреном.

### Пример

**входные данные**

Скопировать

```
9 3
1 2 1 2 1 3 1 2 1
```

**выходные данные**

Скопировать

```
9
9
1 2 1 2 1 3 1 2 1
```

## 1А. Получи дерево

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Александр решил посадить у себя в саду дерево. Он купил его в интернет-магазине, и на следующий день его обещали доставить. Но произошла какая-то ошибка, и вместо дерева Александр получил неориентированный связный граф. Оставлять в таком виде граф нельзя: в нем могут быть циклы, в том числе петли, кратные ребра — совсем некрасиво. Поэтому Александр взял топор и решил удалить некоторое количество ребер из графа.

Выведите любое дерево, которое могло получиться у Александра в результате удаления ребер из графа.

### Входные данные

Сначала вводятся 2 числа  $n$  и  $m$  — число вершин и ребер в графе ( $1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5$ ). Далее идут  $m$  пар чисел, задающих ребра. Гарантируется, что граф связный.

### Выходные данные

Выведите  $n - 1$  пару чисел — ребра, которые войдут в дерево. Ребра, как и концы каждого ребра, можно выводить в любом порядке.

### Пример

входные данные	Скопировать
4 5 1 2 2 3 3 4 1 4 1 2	
выходные данные	Скопировать
1 2 2 3 3 4	

## 1B. Компоненты связности

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Дан неориентированный граф. Необходимо посчитать количество его компонент связности и вывести их.

### Входные данные

В первой строке записаны два числа  $n$  и  $m$  ( $1 \leq n \leq 10^5$ ,  $0 \leq m \leq 10^5$ ) — количество вершин и ребер в графе, соответственно.

Далее следуют  $m$  строк, содержащих описание ребер графа. В каждой строке записаны два числа  $u$  и  $v$  ( $1 \leq u, v \leq n$ ,  $u \neq v$ ), означающие, что вершины  $u$  и  $v$  соединены ребром.

### Выходные данные

В первой строке выведите одно число — количество компонент связности.

Далее выведите сами компоненты в следующем формате: в первой строке следует вывести количество вершин в компоненте, а во второй строке — номера вершин, принадлежащих данной компоненте.

### Пример

<b>входные данные</b>	Скопировать
<pre>6 4 3 1 1 2 5 4 2 3</pre>	
<b>выходные данные</b>	Скопировать
<pre>3 3 1 3 2 2 4 5 1 6</pre>	



## 1F. Topsort

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

### Входные данные

В первой строке входного файла даны два целых числа  $N$  и  $M$  ( $1 \leq N \leq 100\,000, 0 \leq M \leq 100\,000$ ) — количества вершин и рёбер в графе соответственно. Далее в  $M$  строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

### Выходные данные

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести «-1».

### Пример

<b>входные данные</b>	Скопировать
<pre>6 6 1 2 3 2 4 2 2 5 6 5 4 6</pre>	
<b>выходные данные</b>	Скопировать
<pre>4 6 3 1 2 5</pre>	



## 11. Кратчайший путь

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Дан ориентированный взвешенный ациклический граф. Требуется найти в нем кратчайший путь из вершины  $s$  в вершину  $t$ .

### Входные данные

Первая строка входного файла содержит четыре целых числа  $n$ ,  $m$ ,  $s$  и  $t$  — количество вершин, дуг графа, начальная и конечная вершина соответственно. Следующие  $m$  строк содержат описания дуг по одной на строке. Ребро номер  $i$  описывается тремя целыми числами  $b_i$ ,  $e_i$  и  $w_i$  — началом, концом и длиной дуги соответственно ( $1 \leq b_i, e_i \leq n$ ,  $|w_i| \leq 1000$ ).

Входной граф не содержит циклов и петель.

$1 \leq n \leq 100\,000$ ,  $0 \leq m \leq 200\,000$ .

### Выходные данные

Первая строка выходного файла должна содержать одно целое число — длину кратчайшего пути из  $s$  в  $t$ . Если пути из  $s$  в  $t$  не существует, выведите «Unreachable».

### Примеры

<b>входные данные</b>	Скопировать
<pre>2 1 1 2 1 2 -10</pre>	
<b>выходные данные</b>	Скопировать
<pre>-10</pre>	

  

<b>входные данные</b>	Скопировать
<pre>2 1 2 1 1 2 -10</pre>	
<b>выходные данные</b>	Скопировать
<pre>Unreachable</pre>	

# 1К. Топологическая сортировка

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Задан ориентированный ациклический граф с  $n$  вершинами и  $m$  ребрами. Также задана перестановка вершин графа. Необходимо проверить, является ли данная перестановка топологической сортировкой.

## Входные данные

В первой строке даны два числа  $n$  и  $m$  — количество вершин и ребер в графе соответственно ( $1 \leq n, m \leq 10^5$ ). В следующих  $m$  строках заданы пары чисел  $u_i, v_i$ , означающие, что в графе есть ребро из вершины  $u_i$  в вершину  $v_i$ . В последней строке задана перестановка из  $n$  элементов.

## Выходные данные

Выведите «YES» (без кавычек), если данная перестановка является топологической сортировкой и «NO» в противном случае.

## Примеры

входные данные	Скопировать
<pre>3 3 2 3 1 3 1 2 2 1 3</pre>	
выходные данные	Скопировать
<pre>NO</pre>	

входные данные	Скопировать
<pre>3 3 3 2 1 2 3 1 3 1 2</pre>	
выходные данные	Скопировать
<pre>YES</pre>	

## 2A. Конденсация графа

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт

Вам задан ориентированный граф с  $N$  вершинами и  $M$  ребрами ( $1 \leq N \leq 200\,000$ ,  $1 \leq M \leq 200\,000$ ). Найдите компоненты сильной связности заданного графа и топологически отсортируйте его конденсацию.

### Входные данные

Граф задан во входном файле следующим образом: первая строка содержит числа  $N$  и  $M$ . Каждая из следующих  $M$  строк содержит описание ребра – два целых числа из диапазона от 1 до  $N$  – номера начала и конца ребра.

### Выходные данные

На первой строке выведите число  $K$  – количество компонент сильной связности в заданном графе. На следующей строке выведите  $N$  чисел – для каждой вершины выведите номер компоненты сильной связности, которой принадлежит эта вершина. Компоненты сильной связности должны быть занумерованы таким образом, чтобы для любого ребра номер компоненты сильной связности его начала не превышал номера компоненты сильной связности его конца.

### Пример

входные данные

Скопировать

```
10 19
1 4
7 8
5 10
8 9
9 6
2 6
6 2
3 8
9 2
7 2
9 7
4 5
3 6
7 3
6 7
10 8
10 1
2 9
2 7
```

выходные данные

Скопировать

```
2
1 2 2 1 1 2 2 2 2 1
```

## 3A. Мосты

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Дан неориентированный граф. Требуется найти все мосты в нём.

### Входные данные

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и рёбер графа соответственно ( $1 \leq n \leq 20\,000$ ,  $1 \leq m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание рёбер по одному на строке. Рёбро номер  $i$  описывается двумя натуральными числами  $b_i$ ,  $e_i$  — номерами концов ребра ( $1 \leq b_i, e_i \leq n$ ).

### Выходные данные

Первая строка выходного файла должна содержать одно натуральное число  $b$  — количество мостов в заданном графе.

На следующей строке выведите  $b$  целых чисел — номера рёбер, которые являются мостами, в возрастающем порядке. Рёбра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

### Пример

<b>входные данные</b>	Скопировать
<pre>6 7 1 2 2 3 3 4 1 3 4 5 4 6 5 6</pre>	
<b>выходные данные</b>	Скопировать
<pre>1 3</pre>	

## 3B. Точки сочленения

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

{256 мегабайт}

Дан неориентированный граф. Требуется найти все точки сочленения в нём.

### Входные данные

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и рёбер графа соответственно ( $1 \leq n \leq 20\,000$ ,  $1 \leq m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание рёбер по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i$ ,  $e_i$  — номерами концов ребра ( $1 \leq b_i, e_i \leq n$ ).

### Выходные данные

Первая строка выходного файла должна содержать одно натуральное число  $b$  — количество точек сочленения в заданном графе. На следующей строке выведите  $b$  целых чисел — номера вершин, которые являются точками сочленения, в возрастающем порядке.

### Пример

<b>входные данные</b>	Скопировать
<pre>6 7 1 2 2 3 2 4 2 5 4 5 1 3 3 6</pre>	
<b>выходные данные</b>	Скопировать
<pre>2 2 3</pre>	

## 3Е. Компоненты реберной двусвязности

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 64 мегабайта

Компонентой реберной двусвязности графа  $\langle V, E \rangle$  называется подмножество вершин  $S \subset V$ , такое что для любых различных  $u$  и  $v$  из этого множества существует не менее двух реберно не пересекающихся путей из  $u$  в  $v$ .

Дан неориентированный граф. Требуется выделить компоненты реберной двусвязности в нем.

### Входные данные

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и ребер графа соответственно ( $1 \leq n \leq 20\,000$ ,  $1 \leq m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i, e_i$  — номерами концов ребра ( $1 \leq b_i, e_i \leq n$ ).

### Выходные данные

В первой строке выходного файла выведите целое число  $k$  — количество компонент реберной двусвязности графа.

Во второй строке выведите  $n$  натуральных чисел  $a_1, a_2, \dots, a_n$ , не превосходящих  $k$ , где  $a_i$  — номер компоненты реберной двусвязности, которой принадлежит  $i$ -я вершина.

Компоненты требуется нумеровать в порядке возрастания минимального номера вершины, входящей в компоненту.

### Пример

входные данные

Скопировать

```
6 7
1 2
2 3
3 1
1 4
4 5
4 6
5 6
```

выходные данные

Скопировать

```
2
1 1 1 2 2 2
```

## 5A. Кратчайший путь в невзвешенном графе

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Дан ориентированный невзвешенный граф. Также даны стартовая и конечная вершина. Найдите и выведите кратчайший путь между этими двумя вершинами.

### Входные данные

В первой строке даны  $n$  и  $m$  — число вершин и ребер в графе ( $1 \leq n \leq 10^5, 0 \leq m \leq 2 \cdot 10^5$ ). В следующих  $m$  строках идут пары чисел, задающие ребра.

В последней строке даны  $s$  и  $t$  — номера стартовой и конечной вершины ( $1 \leq s, t \leq n, s \neq t$ ).

### Выходные данные

Если пути не существует, выведите «-1» без кавычек. Иначе в первой строке выведите длину кратчайшего пути, а в следующей — вершины пути, включая стартовую и конечную.

### Пример

<b>входные данные</b>	Скопировать
<pre>5 5 1 2 2 3 3 4 1 5 5 4 1 4</pre>	
<b>выходные данные</b>	Скопировать
<pre>2 1 5 4</pre>	



## 6A. Пифагоров экспресс (для C++)

ограничение по времени на тест: 2.5 секунд

ограничение по памяти на тест: 256 мегабайт

В стране Флатландии есть  $n$  городов, расположенных в целочисленных точках плоскости. Транспортная система Флатландии настолько развита, что между любыми двумя городами ходит экспресс имени Пифагора. С помощью него можно добраться от города с координатами  $x_1, y_1$  до города с координатами  $x_2, y_2$  за время  $(x_1 - x_2)^2 + (y_1 - y_2)^2$ . По каждой линии ходит достаточно поездов, и временем на пересадки можно пренебречь.

Сообщение с внешним миром во Флатландии продумано несколько хуже, и единственный аэропорт международного сообщения находится в городе с номером  $s$ . Вам же хочется попасть в город с номером  $t$ . Определите, за какое наименьшее время это можно сделать.

### Входные данные

В первой строке задано число городов  $n$  ( $1 \leq n \leq 2 \cdot 10^4$ ). В следующих  $n$  строках заданы координаты городов  $x_i, y_i$  ( $|x_i|, |y_i| \leq 10^4$ ). В последней строке даны  $s$  и  $t$  — номера начального и конечного города в пути ( $1 \leq s, t \leq n$ ).

### Выходные данные

Выведите одно число — минимальное время, за которое можно добраться из  $s$  в  $t$ .

### Пример

<b>входные данные</b>	Скопировать
<pre>3 0 1 0 0 1 3 2 3</pre>	
<b>выходные данные</b>	Скопировать
<pre>6</pre>	

## 6B. Флойд

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Дан ориентированный граф, в котором могут быть кратные ребра и петли. Каждое ребро имеет вес, выражающийся целым числом (возможно, отрицательным). Гарантируется, что циклы отрицательного веса отсутствуют.

Требуется посчитать матрицу длин кратчайших путей от каждой вершины до всех остальных вершин.

### Входные данные

Программа получает сначала число  $N$  ( $1 \leq N \leq 100$ ) — количество вершин графа и число  $M$  ( $0 \leq M \leq 10^4$ ) — количество ребер.

В следующих строках идет  $M$  троек чисел, описывающих ребра: начало ребра, конец ребра и вес (вес — целое число от  $-100$  до  $100$ ).

### Выходные данные

Требуется вывести матрицу  $N$  на  $N$ . В  $j$ -м столбце  $i$ -й строки следует вывести длину кратчайшего пути от вершины  $i$  до вершины  $j$ , либо число **30 000**, если не существует пути между данными вершинами.

### Пример

входные данные

Скопировать

```
6 4
1 2 10
2 3 10
1 3 100
4 5 -10
```

выходные данные

Скопировать

```
0 10 20 30000 30000 30000
30000 0 10 30000 30000 30000
30000 30000 0 30000 30000 30000
30000 30000 30000 0 -10 30000
30000 30000 30000 30000 0 30000
30000 30000 30000 30000 30000 0
```

## 7A. Остовное дерево

ограничение по времени на тест: 8 секунд

ограничение по памяти на тест: 256 мегабайт

Даны точки на плоскости, являющиеся вершинами полного графа. Вес ребра равен расстоянию между точками, соответствующими концам этого ребра. Требуется в этом графе найти остовное дерево минимального веса.

### Входные данные

Первая строка входного файла содержит натуральное число  $n$  — количество вершин графа ( $1 \leq n \leq 10\,000$ ). Каждая из следующих  $n$  строк содержит два целых числа  $x_i, y_i$  — координаты  $i$ -й вершины ( $-10\,000 \leq x_i, y_i \leq 10\,000$ ). Никакие две точки не совпадают.

### Выходные данные

Первая строка выходного файла должна содержать одно вещественное число — вес минимального остовного дерева.

### Пример

**входные данные**

Скопировать

```
2
0 0
1 1
```

**выходные данные**

Скопировать

```
1.4142135624
```

## 7B. Остовное дерево 2

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Требуется найти в связном графе остовное дерево минимального веса.

### Входные данные

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно. Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается тремя натуральными числами  $b_i$ ,  $e_i$  и  $w_i$  — номера концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 100\,000$ ).  $n \leq 200\,000$ ,  $m \leq 200\,000$ .

Граф является связным.

### Выходные данные

Первая строка выходного файла должна содержать одно натуральное число — вес минимального остовного дерева.

### Пример

<b>входные данные</b>	Скопировать
<pre>4 4 1 2 1 2 3 2 3 4 5 4 1 4</pre>	
<b>выходные данные</b>	Скопировать
<pre>7</pre>	

## 8A. Префикс-функция

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт

Вам дана строка из строчных латинских букв. От вас требуется вычислить префикс-функцию данной строки.

### Входные данные

Вводится строка, состоящая из строчных латинских букв. Длина строки не превышает  $10^6$ .

### Выходные данные

Требуется вывести префикс-функцию данной строки.

### Пример

<b>входные данные</b>	Скопировать
abacaba	
<b>выходные данные</b>	Скопировать
0 0 1 0 1 2 3	

### Примечание

Предполагается, что значение префикс-функции для первого символа равно нулю.

## 8B. N-функция

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Вам дана строка из строчных латинских букв. От вас требуется вычислить N-функцию данной строки.

### Входные данные

Вводится строка, состоящая из строчных латинских букв. Длина строки не превышает  $10^6$ .

### Выходные данные

Требуется вывести N-функцию данной строки.

### Пример

<b>входные данные</b>	Скопировать
abacaba	
<b>выходные данные</b>	Скопировать
7 0 1 0 3 0 1	

### Примечание

Предполагается, что значение N-функции для первого символа равно длине строки.

## 8С. Поиск подстроки (1 балл)

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт

Найти все вхождения строки  $T$  в строку  $S$ .

### Входные данные

Первые две строки входных данных содержат строки  $S$  и  $T$ , соответственно. Длины строк больше 0 и меньше 500 000, строки содержат только строчные латинские буквы.

### Выходные данные

Выведите номера символов, начиная с которых строка  $T$  входит в строку  $S$ , в порядке возрастания.

### Пример

<b>входные данные</b>	Скопировать
ababbababa aba	
<b>выходные данные</b>	Скопировать
0 5 7	



## 8D. Строчечки

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Мальчик Кирилл написал однажды на листе бумаги строчку, состоящую из больших и маленьких латинских букв, а после этого ушел играть в футбол. Когда он вернулся, то обнаружил, что его друг Дима написал под его строкой еще одну строчку такой же длины. Дима утверждает, что свою строчку он получил циклическим сдвигом строки Кирилла на несколько шагов вправо (циклический сдвиг строки «abcde» на 2 позиции вправо даст строку «deabc»). Однако Дима известен тем, что может случайно ошибиться в большом количестве вычислений, поэтому Кирилл в растерянности – верить ли Диме? Помогите ему! По данным строкам выведите минимальный возможный размер сдвига или  $-1$ , если Дима ошибся.

### Входные данные

Первые две строки входных данных содержат строки Кирилла и Димы, соответственно. Длины строк одинаковы, не превышают  $10^6$  и не равны 0.

### Выходные данные

Выведите единственное число – ответ на вопрос задачи.

### Пример

<b>входные данные</b>	Скопировать
zabcd abcdz	
<b>выходные данные</b>	Скопировать
4	