

## Лабораторная работа 2-3. Алгоритмы на строках

### А. Сравнения подстрок

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

**ввод: стандартный ввод**

**вывод: стандартный вывод**

Дана строка. Нужно уметь отвечать на запросы вида: равны ли подстроки  $[a..b]$  и  $[c..d]$ .

#### Входные данные

Сперва строка  $S$  (не более  $10^5$  строчных латинских букв). Далее число  $M$  — количество запросов.

В следующих  $M$  строках запросы  $a, b, c, d$ .  $0 \leq M \leq 10^5$ ,  $1 \leq a \leq b \leq |S|$ ,  $1 \leq c \leq d \leq |S|$

#### Выходные данные

$M$  строк. Выведите Yes, если подстроки совпадают, и No иначе.

#### Пример

входные данные	Скопировать
trololo 3 1 7 1 7 3 5 5 7 1 1 1 5	
выходные данные	Скопировать
Yes Yes No	

## В. Префикс-функция

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

**ввод: стандартный ввод**

**вывод: стандартный вывод**

Постройте префикс-функцию для заданной строки  $s$ .

### Входные данные

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

### Выходные данные

Выведите значения префикс-функции строки  $s$  для всех индексов  $1, 2, \dots, |s|$ .

### Пример

#### входные данные

[Скопировать](#)

aaaAAA

#### выходные данные

[Скопировать](#)

0 1 2 0 0 0

## С. Z-функция

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

**ввод: стандартный ввод**

**вывод: стандартный вывод**

Постройте Z-функцию для заданной строки  $s$ .

### Входные данные

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

### Выходные данные

Выведите значения Z-функции строки  $s$  для индексов  $2, 3, \dots, |s|$ .

### Примеры

<b>входные данные</b>	Скопировать
aaaAAA	
<b>выходные данные</b>	Скопировать
2 1 0 0 0	

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

## D. Быстрый поиск подстроки в строке

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

**ввод: стандартный ввод**

**вывод: стандартный вывод**

Даны строки  $p$  и  $t$ . Требуется найти все вхождения строки  $p$  в строку  $t$  в качестве подстроки.

### Входные данные

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

### Выходные данные

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.

### Пример

<b>входные данные</b>	Скопировать

aba  
abaCaba

выходные данные

Скопировать

2  
1 5

## Е. Поиск периода

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

**ввод: стандартный ввод**

**вывод: стандартный вывод**

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

### Входные данные

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

### Выходные данные

Выведите длину искомой строки  $t$ .

### Примеры

входные данные

Скопировать

abcabcabc

выходные данные

Скопировать

3

входные данные

Скопировать

abacaba

выходные данные

Скопировать

7

## F. Подстроки-3

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

**ввод:** стандартный ввод

**вывод:** стандартный вывод

Даны  $K$  строк из маленьких латинских букв. Требуется найти их наибольшую общую подстроку.

### Входные данные

В первой строке число  $K$  ( $1 \leq K \leq 10$ ).

В следующих  $K$  строках — собственно  $K$  строк (длины строк от 1 до 10 000).

### Выходные данные

Наибольшая общая подстрока.

### Пример

<b>входные данные</b>	Скопировать
3 abacaba mycabarchive acabistrue	
<b>выходные данные</b>	Скопировать
cab	

## G. Множественный поиск

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

**ввод:** search4.in

**вывод:** search4.out

Дан массив строк  $s_i$  и строка  $t$ . Требуется для каждой строки  $s_i$  определить, встречается ли она в  $t$  как подстрока.

### Входные данные

Первая строка входного файла содержит целое число  $n$  — число элементов в  $s$  ( $1 \leq n \leq 10^6$ ). Следующие  $n$  строк содержат по одной строке  $s_i$ . Сумма длин всех строк из  $s$  не превосходит  $10^6$ . Последняя строка входного файла содержит  $t$  ( $1 \leq t \leq 10^6$ ). Все строки состоят из строчных латинских букв.

### Выходные данные

Для каждой строки  $s_i$  выведите «YES», если она встречается в  $t$  и «NO» в противном случае. Строки нумеруются в порядке появления во входном файле.

### Пример

<b>входные данные</b>	Скопировать
3 abc abcdr abcde xabcdef	
<b>выходные данные</b>	Скопировать
YES NO YES	

## Н. Множественный поиск 2

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

**ввод:** search5.in

**вывод:** search5.out

Дан массив строк  $s_i$  и строка  $t$ . Требуется для каждой строки  $s_i$  определить, сколько раз она встречается в  $t$  как подстрока.

### Входные данные

Первая строка входного файла содержит целое число  $n$  — число элементов в  $s$  ( $1 \leq n \leq 10^6$ ). Следующие  $n$  строк содержат по одной строке  $s_i$ . Сумма длин всех строк из  $s$  не превосходит  $10^6$ . Последняя строка входного файла содержит  $t$  ( $1 \leq t \leq 10^6$ ). Все строки состоят из строчных латинских букв.



## Выходные данные

Для каждой строки  $s_i$  выведите одно число: сколько раз она встречается в  $t$ . Строки нумеруются в порядке появления во входном файле.

### Пример

<b>входные данные</b>	Скопировать
3 abc abcdr abcde xabcdef	
<b>выходные данные</b>	Скопировать
1 0 1	

## I. Множественный поиск 3

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 512 мегабайт

**ввод:** search6.in

**вывод:** search6.out

Дан массив строк  $s_i$  и строка  $t$ . Требуется для каждой строки  $s_i$  найти самое левое и самое правое вхождение в  $t$  как подстроки.

## Входные данные

Первая строка входного файла содержит целое число  $n$  — число элементов в  $s$  ( $1 \leq n \leq 10^6$ ). Следующие  $n$  строк содержат по одной строке  $s_i$ . Сумма длин всех строк из  $s$  не превосходит  $10^6$ . Последняя строка входного файла содержит  $t$  ( $1 \leq t \leq 10^6$ ). Все строки состоят из строчных латинских букв.

## Выходные данные

Для каждой строки  $s_i$  выведите два числа: индексы самой левой и самой правой позиции, в которых она встречается в  $t$ . Если строка не встречается в  $t$  ни разу, выведите -1 -1. Строки нумеруются в порядке появления во входном файле. Позиции нумеруются с 0.

### Пример

**Входные данные**[Скопировать](#)

```
3
ab
bcd
abde
abcdab
```

**Выходные данные**[Скопировать](#)

```
0 4
1 1
-1 -1
```

## Ж. Суффиксный массив

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 512 мегабайт

**ввод:** array.in

**вывод:** array.out

Постройте суффиксный массив для заданной строки  $s$ , для каждой двух соседних суффиксов найдите длину максимального общего префикса.

**Входные данные**

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

**Выходные данные**

В первой строке выведите  $|s|$  различных чисел — номера первых символов суффиксов строки  $s$  так, чтобы соответствующие суффиксы были упорядочены в лексикографически возрастающем порядке. Во второй строке выведите  $|s| - 1$  чисел — длины наибольших общих префиксов.

**Пример****Входные данные**[Скопировать](#)

```
ababb
```

**Выходные данные**[Скопировать](#)

```
1 3 5 2 4
2 0 1 1
```



## К. Количество подстрок

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 512 мегабайт

**ввод:** count.in  
**вывод:** count.out

Вычислите количество различных подстрок строки  $s$ .

### Входные данные

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

### Выходные данные

Выведите одно число — ответ на задачу.

### Пример

<b>входные данные</b>	Скопировать
ababb	
<b>выходные данные</b>	Скопировать
11	

## Л. Циклические сдвиги

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 512 мегабайт

**ввод:** shifts.in  
**вывод:** shifts.out

$k$ -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.

### Примеры

<b>входные данные</b>	Скопировать
<code>abacabac</code> <code>4</code>	
<b>выходные данные</b>	Скопировать
<code>cabacaba</code>	

  

<b>входные данные</b>	Скопировать
<code>abacabac</code> <code>5</code>	
<b>выходные данные</b>	Скопировать
<code>IMPOSSIBLE</code>	

## М. Наибольшая общая подстрока

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 512 мегабайт

**ввод:** `common.in`

**вывод:** `common.out`

Найдите наибольшую общую подстроку строк  $s$  и  $t$ .

### Входные данные

Первая строка входного файла содержит строку  $s$ , вторая —  $t$  ( $1 \leq |s|, |t| \leq 100,000$ ). Строки состоят из строчных латинских букв.

### Выходные данные

Выведите одну строку — наибольшую общую подстроку строк  $s$  и  $t$ . В случае, если ответ не единственный, выведите минимальный лексикографически.

### Пример

#### входные данные

[Скопировать](#)

```
bababb  
zabacabba
```

#### выходные данные

[Скопировать](#)

```
aba
```

---

[Codeforces](#) (c) Copyright 2010-2018 Михаил Мирзаянов  
Соревнования по программированию 2.0