

# Блиц

## Задача А. Суффиксный массив

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

Данна строка, требуется построить суффиксный массив для этой строки. Суффиксный массив — лексикографически отсортированный массив всех суффиксов строки. Каждый суффикс задается целым числом — позицией начала.

Строка  $s$  лексикографически меньше строки  $t$ , если есть такое  $i$ , что  $s_i < t_i$  и  $s_j = t_j$  для всех  $j < i$ . Или, если такого  $i$  не существует и строка  $s$  короче строки  $t$ .

Здесь  $s_i$  — код  $i$ -го символа строки  $s$ .

### Формат входных данных

Файл состоит из единственной строки. Эта строка — **английский литературный текст**. Длина текста не превосходит  $10^5$ . Коды всех символов в тексте от 32 до 127.

### Формат выходных данных

Выведите  $N$  чисел — суффиксный массив данной строки.

### Пример

suffarray.in	suffarray.out
99 bottles of beer.	14 3 11 19 2 1 15 4 16 17 9 13 8 12 5 18 10 7 6

## Задача В. Различные подстроки

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

*Подстрокой* строки  $s = s_1s_2 \dots s_n$  называется непрерывная подпоследовательность символов этой строки  $s_is_{i+1}s_{i+2} \dots s_{j-1}s_j$ .

Дана строка. Сколько различных подстрок, не считая пустой, она содержит?

### Формат входных данных

В первой строке входного файла задана строка длины от 1 до 100 символов, включительно. Строка состоит из строчных букв латинского алфавита.

### Формат выходных данных

В первой строке выходного файла выведите одно число — количество различных подстрок данной строки, не считая пустой.

### Примеры

unequal.in	unequal.out
aab	5
dabxy	15

## Задача С. Сравнения подстрок

Имя входного файла: `substrcmp.in`  
Имя выходного файла: `substrcmp.out`  
Ограничение по времени: 1 секунда  
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
trololo	Yes
3	Yes
1 7 1 7	No
3 5 5 7	
1 1 1 5	

## Задача D. Основание строки

Имя входного файла: `basis.in`  
Имя выходного файла: `basis.out`  
Ограничение по времени: 0.4 секунды  
Ограничение по памяти: 256 мебибайт

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

В первой и единственной строке входного файла записана строка, которая содержит только латинские буквы, длина строки не превышает 50 000 символов.

### Формат выходных данных

В выходной файл выведите ответ на задачу.

### Пример

<code>basis.in</code>	<code>basis.out</code>
zzz	1
bcabcab	3

## Задача E. Циклические суффиксы

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

Рассмотрим строку  $S = s_1s_2s_3 \dots s_{n-1}s_n$  над алфавитом  $\Sigma$ . *Циклическим расширением* порядка  $m$  строки  $S$  назовем строку  $s_1s_2s_3 \dots s_{n-1}s_ns_1s_2 \dots$  из  $m$  символов; это значит, что мы приписываем строку  $S$  саму к себе, пока не получим требуемую длину, и берем префикс длины  $m$ .

*Циклической строкой*  $\tilde{S}$  назовем бесконечное циклическое расширение строки  $S$ .

Рассмотрим суффиксы циклической строки  $\tilde{S}$ . Очевидно, существует не более  $|S|$  различных суффиксов:  $(n + 1)$ -ый суффикс совпадает с первым,  $(n + 2)$ -ой совпадает со вторым, и так далее. Более того, различных суффиксов может быть даже меньше. Например, если  $S = \text{abab}$ , первые четыре суффикса циклической

строки  $\tilde{S}$  — это:

$$\begin{aligned}\tilde{S}_1 &= \text{ababababab} \dots \\ \tilde{S}_2 &= \text{bababababa} \dots \\ \tilde{S}_3 &= \text{ababababab} \dots \\ \tilde{S}_4 &= \text{bababababa} \dots\end{aligned}$$

Здесь существует всего два различных суффикса, в то время как  $|S| = 4$ .

Отсортируем первые  $|S|$  суффиксов  $\tilde{S}$  лексикографически. Если два суффикса совпадают, первым поставим суффикс с меньшим индексом. Теперь нас интересует следующий вопрос: на каком месте в этом списке стоит сама строка  $\tilde{S}$ ?

Например, рассмотрим строку  $S = \text{cabcab}$ :

$$\begin{aligned}(1) \quad \tilde{S}_2 &= \text{abcabcabca} \dots \\ (2) \quad \tilde{S}_5 &= \text{abcabcabca} \dots \\ (3) \quad \tilde{S}_3 &= \text{bcabcabcab} \dots \\ (4) \quad \tilde{S}_6 &= \text{bcabcabcab} \dots \\ (5) \quad \tilde{S}_1 &= \text{cabcabcabc} \dots \\ (6) \quad \tilde{S}_4 &= \text{cabcabcabc} \dots\end{aligned}$$

Здесь циклическая строка  $\tilde{S} = \tilde{S}_1$  находится на пятом месте.

Вам дана строка  $S$ . Ваша задача — найти позицию циклической строки  $\tilde{S}$  в описанном порядке.

### Формат входных данных

Во входном файле записана единственная строка  $S$  ( $1 \leq |S| \leq 1\,000\,000$ ), состоящая из прописных латинских букв.

### Формат выходных данных

В выходной файл выведите единственное число — номер строки  $\tilde{S}$  в описанном порядке среди первых  $|S|$  суффиксов.

### Примеры

<code>cyclic.in</code>	<code>cyclic.out</code>
abracadabra	3
cabcab	5

## Простые задачи

### Задача F. LCP для суффиксного массива

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

Дана строка длины  $N$  и отсортированный массив суффиксов этой строки (т.е. суффиксный массив), вам нужно вычислить LCP. При сортировке строка **a** считается меньше строки **aa**. LCP — наибольший общий префикс двух последовательных суффиксов в суффиксном массиве.

#### Формат входных данных

В первой строке число  $N$  ( $1 \leq N \leq 10^5$ ). На второй строке файла дана  $N$  строчных латинских букв. В третьей строке  $N$  чисел от 1 до  $N$  — суффиксный массив (числом  $i$  кодируется суффикс, начинающийся с  $i$ -го символа).

#### Формат выходных данных

Выведите  $N - 1$  число — значения LCP.

#### Пример

sufflcp.in	sufflcp.out
5 сасао 2 4 1 3 5	1 0 2 0

#### Замечание

Суффиксный массив для строки `сасао`:

асао  
ао  
сасао  
сао  
о

### Задача G. Словарь

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

Дан набор слов и текст, требуется определить для каждого слова, присутствует ли оно в тексте как подстрока.

#### Формат входных данных

В первой строке дан текст (не более  $10^6$  строчных латинских букв). Далее дано число  $M$  — количество слов в словаре.

В следующих  $M$  строках записаны слова (не более 30 строчных латинских букв). Слова различны и отсортированы в лексикографическом порядке.

Суммарная длина слов в словаре не более  $10^5$ .

#### Формат выходных данных

$M$  строк вида Yes, если слово присутствует, и No иначе.

#### Пример

dictionary.in	dictionary.out
trololo	No
3	Yes
abacabadabacaba	Yes
olo	
trol	

### Задача H. Циклические сдвиги

Имя входного файла: `shifts.in`  
Имя выходного файла: `shifts.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 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`.

### Формат входных данных

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  $k$ .

### Формат выходных данных

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

### Пример

shifts.in	shifts.out
abacabac 4	cabacaba
abacabac 5	IMPOSSIBLE

## Задача I. 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 20 000 символов.

### Формат выходных данных

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

substr.in	substr.out
aaaa	4
abacaba	21

# Классические задачи

## Задача J. Общая подстрока

Имя входного файла: **common.in**  
Имя выходного файла: **common.out**  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Заданы две строки, состоящие из 0 и 1. Рассмотрим все строки, которые являются подстроками обеих данных строк. Найдите среди них  $k$ -ую в лексикографическом порядке.

Строка  $S$  меньше строки  $T$  в лексикографическом порядке, если выполняется одно из двух условий:

- $S$  является префиксом  $T$ ;
- существует  $i$ , не превышающее длин строк  $S$  и  $T$ , такое что для  $j < i$  выполняется  $S[j] = T[j]$  и  $S[i] < T[i]$ .

### Формат входных данных

Первые две строки входного файла содержат заданные строки, длиной не более 4000 символов каждая. Третья строка содержит целое положительное число  $k$ , не превышающее количества общих подстрок двух заданных строк.

### Формат выходных данных

Выведите в выходной файл  $k$ -ую в лексикографическом порядке общую подстроку заданных строк.

### Пример

common.in	common.out
0100 0010 3	01

## Задача К. Поиск набора образцов

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

Напишите программу, которая для каждой строки из заданного набора  $S$  проверяет, верно ли, что она содержит как подстроку одну из строк из набора  $T$ .

### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  — количество строк в наборе  $T$  ( $n \leq 1000$ ). Каждая из следующих  $n$  строк содержит непустую строку длины не более 80-ти символов.

Оставшаяся часть файла содержит строки из набора  $S$ . Каждая строка состоит из ASCII символов с кодами от 32 до 126 включительно. Строка может быть пустой и ее длина не превышает 250-ти символов.

Гарантируется, что размер входного файла не превышает 1 Мбайт.

### Формат выходных данных

В выходной файл выведите все строки из набора  $S$  (в том порядке, в котором они находятся во входном файле), содержащие как подстроку по крайней мере одну строку из набора  $T$ .

### Пример

console.in	console.out
3 gr sud abc lksh sudislavl kostroma summer group a'	sudislavl group a'

## Задача Л. Палиндромы

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

Строка называется палиндромом, если она одинаково читается как слева направо, так и справа налево. Например, `abba` — палиндром, а `omax` — 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

palindrome.in	palindrome.out
aaa	3
abba	2
omax	0

## Задача М. Рефрен

Имя входного файла: `refrain.in`  
Имя выходного файла: `refrain.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 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.

### Формат входных данных

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  $m$ .

### Формат выходных данных

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

refrain.in	refrain.out
9 3	9
1 2 1 2 1 3 1 2 1	3
	1 2 1

## Если вам стало скучно

### Задача N. Под-бор

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

*Бором* называется подвешенное дерево, на каждом из рёбер которого написано по символу, причём символы, написанные на рёбрах, выходящих из общей вершины-родителя, различны. Будем называть направление от родителя к детям “вниз”. Назовем *вхождением строки s в бор* такую вершину бора, от которой можно пройти несколько шагов вниз таким образом, что встретившиеся символы образуют строку *s*.

Даны бор и несколько строк, найдите сумму количеств вхождений этих строк в этот бор.

#### Формат входных данных

В первой строке входного файла записано единственное число  $n$ ,  $1 \leq n \leq 100\,000$  — количество вершин бора. В следующих  $n$  строках описаны вершины бора. В  $(i + 1)$ -й строке описаны дети  $i$ -й вершины: число  $k_i$  ее детей, затем  $k_i$  пар из номера вершины-ребёнка и символа, написанного на соответствующем ребре. Номер родителя всегда меньше номера ребёнка; корнем бора является вершина номер 1.

В  $(n + 2)$ -й строке записано количество  $m$  ( $1 \leq m \leq 100\,000$ ) строк для поиска. В следующих  $m$  строках перечислены сами строки. Входные строки непусты, а их суммарная длина не превышает 100 000 символов.

Все символы, написанные на рёбрах, а также все символы, составляющие строки — маленькие латинские буквы.

### Формат выходных данных

Выведите одно число — сумму количеств вхождений.

#### Пример

trie.in	trie.out
7	9
2 2 a 4 b	
2 3 a 6 b	
0	
1 5 b	
1 7 b	
0	
0	
4	
b	
bb	
bbb	
bb	

### Задача O. Суффиксный пулемёт

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

Или зачёт, или автомат.

---

Ганнибал Ректор

Теоретическая подготовка новобранцев армии Поссилтума включала в себя не только занятия по военному праву, но и начала криптографии. Лекции читал майор Мега Байт, не чуждый солдатского юмора. Гвидо и Нунцио, в чьё задание входил развал армии Поссилтума изнутри, решили на этом сыграть, внося путаницу в терминологию. В начале очередной лекции Нунцио поднял руку и спросил:

— Вот вы на прошлой лекции рассказывали про конечные автоматы. А про конечные пулемёты расскажете?

Мега Байт не растерялся.

— Суффиксный пулемёт — это конечный автомат, принимающий все суффиксы данной строки (от нулевого до  $L$ -го включительно, где  $L$  — длина строки), и только их. Сержант Гвидо!

— Я, господин майор!

— Вы сможете отличить автомат от пулемёта?

— Так точно, господин майор!

— Вам дан конечный автомат. Требуется проверить, является ли он суффиксным пулемётом данной строки.

К сожалению, написание программ такого типа не входило в обязанности Гвидо и Нунцио как в Синдикате, так и в корпорации М. И. Ф. Так что соответствующую программу придётся писать Вам.

### Формат входных данных

Во входном файле задан один или несколько тестовых наборов. В первой строке каждого набора заданы количество состояний автомата  $N$ , количество переходов  $M$ , а также количество принимающих состояний  $T$  ( $1 \leq T \leq N \leq 50\,000$ ,  $1 \leq M \leq 100\,000$ ). Во второй строке через пробел заданы  $T$  различных чисел в пределах от 1 до  $N$  — принимающие состояния автомата, в возрастающем порядке. В последующих  $M$  строках заданы переходы в виде  $a_i b_i c_i$ , где  $1 \leq a_i, b_i \leq n$ , а  $c_i$  — маленькая буква латинского алфавита. Переход производится из состояния  $a_i$  в состояние  $b_i$  по букве  $c_i$ . Из каждого состояния  $a_i$  есть не более одного перехода по символу  $c_i$ . Последняя строка описания набора — это строка  $S$ , для которой автомат должен являться пулемётом. Она состоит только из маленьких латинских букв, и ее длина лежит в пределах от 1 до 50 000 включительно. Кроме того, сумма всех  $N$  и суммарная длина всех строк, для которых необходимо произвести проверку, не превосходит 50 000, а сумма всех  $M$  не превосходит 100 000.

Файл заканчивается фиктивным набором, в котором  $N = M = T = 0$ .

Начальным состоянием автомата является первое. Если при интерпретации какой-то строки в автомате отсутствует соответствующий переход, то автомат вываливается по ошибке и строку не принимает. Таким образом, строка принимается, только если при её интерпретации были найдены все переходы, и по их завершении автомат оказался в принимающем состоянии (при этом неважно, были по пути принимающие состояния, или нет).

### Формат выходных данных

Выведите в выходной файл, является ли данный автомат пулемётом, следуя

### Пример

suffix.in
2 1 2
1 2
1 2 a
a
2 2 2
1 2
1 1 a
1 2 b
ab
0 0 0
suffix.out
Automaton 1 is a machinegun.
Automaton 2 is not a machinegun.

## Задача Р. Обратная префикс-функция

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

Для строки  $S$  определим ее префикс-функцию:  $\pi[i] = \max\{k | 0 \leq k < i, S[1..k] = S[i - k + 1..i]\}$  для всех  $1 \leq i \leq N$ , где  $N$  — длина строки. Например, для  $S = abacabaa$  ее префикс-функция имеет вид:  $[0, 0, 1, 0, 1, 2, 3, 1]$ .

Ваша задача — по заданной префикс-функции восстановить строку. В качестве символов строки разрешается использование  $M$  первых строчных букв латинского алфавита.

### Формат входных данных

Входной файл состоит из одного или более наборов входных данных.

В первой строке каждого набора записаны два целых числа  $N$ ,  $M$  ( $N \geq 1, 1 \leq M \leq 26$ ). Во второй строке записана последовательность целых чисел  $\pi[1], \pi[2], \dots, \pi[N]$ . Все числа в последовательности целые неотрицательные, не превосходящие  $10^6$ .

Сумма значений  $N$  по всем наборам не превосходит  $10^6$ , количество наборов входных данных не превосходит  $10^5$ .

### Формат выходных данных

Выведите в первую строку выходного файла YES, если существует искомое слово, и NO в противном случае. В случае положительного ответа выведите во вторую строку выходного файла выведите искомое слово. Если решений несколько, выведите любое.

### Пример

inverse.in	inverse.out
8 3	YES
0 0 1 0 1 2 3 1	abacabaa
1 1	NO
10	

### Пример

2strings.in	2strings.out
ababa bab	ab

## Задача Q. Две строки

Имя входного файла: 2strings.in  
Имя выходного файла: 2strings.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мебибайт

Вам заданы две строки длиной не более 50 000 символов. Назовем строку хорошей, если она удовлетворяет условию, что если дописать ее в конец самой себе достаточно много раз, то в полученной строке будут содержаться в качестве подстрок обе заданные строки.

Например, для строк **ababa** и **bab** строка **ab** является хорошей — действительно, дописав ее в конец себе два раза, мы получим строку **ababab**, которая содержит обе заданные строки в качестве подстрок.

Для двух заданных строк найдите самую короткую хорошую строку.

### Формат входных данных

Входной файл содержит две заданные строки. Строки состоят из символов с ASCII-кодами от 33 до 127. Длина каждой из них не превышает 50 000.

### Формат выходных данных

Выведите в выходной файл ответ на задачу. Если существует несколько различных оптимальных хороших строк, то выведите любую.