

## Задача А. Общая подстрока

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

Заданы две строки, состоящие из нулей и единиц. Рассмотрим все строки, которые являются подстроками обеих данных строк. Найдите среди них  $k$ -ю 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  $T$  и такое, что для всех  $j < i$  выполняется  $S_j = T_j$ , но  $S_i < T_i$ .

### Формат входных данных

Первые две строки содержат заданные строки, непустые и 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  $k$ -ю в лексикографическом порядке общую подстроку заданных строк.

### Пример

common.in	common.out
0100	01
0010	
3	

## Задача В. Конкатенация

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

Дана строка  $S$ , состоящая из строчных букв английского алфавита. Рассмотрим строку  $T(S)$ , представляющую собой конкатенацию всех подстрок  $S$  в лексикографическом порядке.

Например, если  $S = aba$ , её подстроки — это  $\{a, b, a, ab, ba, aba\}$ , подстроки в лексикографическом порядке — это  $\{a, a, ab, aba, b, ba\}$ , и таким образом,  $T(S) = aaabababba$ .

Найдите  $i$ -й символ строки  $T(S)$ .

### Формат входных данных

Ввод состоит из одного или более тестовых случаев.

Каждый тест начинается строкой, содержащей натуральное число  $m$ , которое задаёт число запросов. Следующая строка содержит строку  $S$  ( $1 \leq |S| \leq 5000$ ). Следующая строка содержит  $m$  целых чисел  $a_i$  ( $1 \leq a_i \leq |T(S)|$ ), задающих запросы.

Ввод будет завершён тестом с  $m = 0$ , который не требуется обрабатывать.

Сумма  $m$  по всем тестовым случаям во вводе не превысит 5000.

Сумма длин всех строк  $S$  также не превысит 5000.

### Формат выходных данных

Для каждого тестового случая выведите строку из  $m$  символов: ответы на запросы. Следуйте формату вывода, указанному в примере, как можно точнее.

### Пример

concat.in	concat.out
10	Case #1: aaabababba Case #2: x
aba	
1 2 3 4 5 6 7 8 9 10	
1	
x	
1	
0	

## Задача С. Ненокку

Имя входного файла: nenokku.in  
Имя выходного файла: nenokku.out  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Очень известный автор не менее известной книги решил написать продолжение своего произведения. Он писал все свои книги на компьютере, подключённом к интернету. Из-за такой неосторожности мальчику Ненокку удалось получить доступ к ещё не написанной книге. Каждый вечер мальчик

пробирался на компьютер писателя и скачивал на свой компьютер новые записи.

Ненокку, скачав на свой компьютер очередную главу, заинтересовался, а использовал ли хоть раз писатель слово «книга». Но он не любит читать книги (он лучше почитает что-нибудь в интернете), и поэтому он просит вас узнать про несколько слов, есть они в тексте произведения.

### Формат входных данных

Каждая строка входных данных имеет один из следующих форматов.

- «? *word*» — проверить существование подстроки *word* в произведении. При проверке большие и маленькие буквы следует считать **одинаковыми**. Здесь *word* — это непустой набор из не более 50 английских букв.
- «A *text*» — добавить строку *text* в конец произведения. Здесь *text* — это непустой набор из не более  $10^5$  английских букв.

Писатель только начал работать над произведением, поэтому он не мог написать более  $10^5$  букв. Суммарная длина всех входных данных не превосходит 20 мегабайт.

### Формат выходных данных

В ответ на каждый вопрос выведите «YES», если такое слово есть в тексте, и «NO» в противном случае. Большие и маленькие буквы следует считать **одинаковыми**.

### Пример

nenokku.in	nenokku.out
? love	NO
? is	NO
A Loveis	YES
? love	NO
? WHO	YES
A Whoareyou	
? is	

### Задача D. 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 её длины на количество вхождений в последовательность максимально.

По заданной последовательности требуется найти её рефрен.

### Формат входных данных

Первая строка 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

В первой строке выведите произведение длины рефрена на количество его вхождений, во второй строке — длину рефрена, а в третьей — последовательность, которая является рефреном. Если есть несколько различных рефренов, выведите любой из них.

### Пример

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

### Задача E. Суффиксное дерево

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

Дана строка  $s$ . Постройте сжатое суффиксное дерево для строки  $s$ , содержащее минимально возможное количество вершин, и выведите его.

### Формат входных данных

В первой строке записана строка  $s$  ( $1 \leq |s| \leq 10^5$ ). Гарантируется, что последний символ строки — знак доллара («\$», ASCII-код 36), а остальные символы — маленькие английские буквы.

### Формат выходных данных

Пронумеруйте вершины дерева от 0 до  $n - 1$  в порядке обхода в глубину, обходя поддеревья в порядке лексикографической сортировки исходящих из вершины рёбер. Используйте ASCII-коды символов для определения их порядка.

В первой строке выведите число  $n$  — количество вершин дерева. В следующих  $n - 1$  строках выведите описание вершин дерева, кроме корня, в порядке увеличения их номеров.

Описание вершины дерева  $v$  состоит из трёх целых чисел:  $p, l, r$ . Здесь  $p$  ( $0 \leq p \leq n, p \neq v$ ) — номер родителя текущей вершины. На ребре, ведущем из  $p$  в  $v$ , написана подстрока  $s_l s_{l+1} \dots s_{r-1}$  ( $0 \leq l < r \leq |s|$ , символы строки нумеруются с нуля).

### Примеры

suftree.in	suftree.out
aaa\$	7 0 3 4 0 0 1 2 3 4 2 1 2 4 3 4 4 2 4
b\$	3 0 1 2 0 0 2
ababa\$	10 0 5 6 0 0 1 2 5 6 2 1 3 4 5 6 4 3 6 0 1 3 7 5 6 7 3 6

### Задача F. Слова

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

Дан набор из  $n$  различных слов. Для каждого слова узнайте, сколько раз оно встречается как подстрока во всех остальных словах.

### Формат входных данных

В первой строке записано целое число  $n$  ( $1 \leq n \leq 10\,000$ ). В следующих  $n$  строках записаны слова. Каждое слово непусто и состоит из не более чем 20 маленьких букв английского алфавита. Все слова различны.

### Формат выходных данных

Выведите  $n$  строк, по одному числу на строке. В  $i$ -й строке должно быть записано, сколько раз  $i$ -е слово встречается в других словах как подстрока.

### Примеры

words.in	words.out
1 word	0
2 aba abacaba	2 0
5 less lesss session s ss	1 0 0 10 4