

# Суффиксный массив

Имя входного файла: `array.in`  
Имя выходного файла: `array.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Постройте суффиксный массив для заданой строки  $s$ .

## Формат входных данных

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

## Формат выходных данных

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

## Примеры

<code>array.in</code>	<code>array.out</code>
ababb	1 3 5 2 4
abacaba	7 5 1 3 6 2 4

# LCP для суффиксного массива

Имя входного файла: `sufflcp.in`  
Имя выходного файла: `sufflcp.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

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.

## Примеры

<code>sufflcp.in</code>	<code>sufflcp.out</code>
5	1
сасао	0
2 4 1 3 5	2
	0

## Замечание

Суффиксный массив для строки `сасао`:

асао  
ао  
сасао  
сао  
о

---

## Различные подстроки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Дана строка  $S$ . Назовем ее подстрокой строку с  $i$ -го по  $j$ -й символ ( $i \leq j$ ). Ваша задача — посчитать количество различных подстрок данной строки.

### Формат входных данных

Во входном файле находится одна строка  $S$ , состоящая не более, чем из 200 000 символов. Все символы в строке — маленькие латинские буквы.

### Формат выходных данных

В выходной файл выведите единственное число — количество различных подстрок заданной строки.

### Примеры

стандартный ввод	стандартный вывод
aaba	8

# Контрольное списывание у Юнга

Input file:           kthsubstr.in  
Output file:         kthsubstr.out  
Time limit:          2 seconds  
Memory limit:       64 megabytes

Сегодня на уроке преподаватель Массивов Автомат Укконеви́ч рассказывал своим ученикам про строки, суффиксные структуры и всё такое. Например, он рассказал им, как сравнить две строки  $A$  и  $B$  лексикографически. Если одна из них является префиксом другой, то более короткая будет лексикографически меньше, иначе необходимо сравнить символы стоящие на первой позиции, в которой они отличаются. Строка с меньшим по номеру в алфавите символом на данной позиции и будет лексикографически меньше.

Чтобы проверить понимание учениками нового материала, Автомат Укконеви́ч дал им следующее задание: найти  $k$ -ю лексикографически непустую уникальную подстроку строки  $S$ .

Так как учитель знает, что Михаил В. и Роман Б. очень любят списывать у известного в узких кругах Юнга, каждый школьник получил своё число  $k$  и вынужден был обратиться к вам за помощью.

## Input

В первой строке входного файла находится строка  $S$  ( $|S| \leq 10^5$ ). Вторая строка содержит число  $k$  ( $1 \leq k \leq 10^{18}$ ) — порядковый номер запрашиваемой подстроки.

## Output

Если ответ существует, выведите искомую подстроку строки  $S$ . В противном случае выведите её лексикографически максимальную подстроку.

## Examples

kthsubstr.in	kthsubstr.out
abacaba 10	acab
abracadabra 10000000000000000000	racadabra

---

## Поиск подподстроки в подстроке

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 4.5 секунд  
Ограничение по памяти: 256 мегабайт

Опытным участникам соревнований по спортивному программированию хорошо известна классическая задача о нахождении количества вхождений подстроки в строку. Обычно она формулируется так: дана строка-образец  $s$  и строка  $t$ , требуется найти количество индексов, начиная с которых строка  $s$  содержится в строке  $t$ .

К сожалению, для решения этой задачи уже придумано множество алгоритмов, поэтому сама по себе она может быть интересна только в качестве упражнения, но не олимпиадной задачи. Однако, как это часто бывает со стандартными задачами, её легко усложнить — представим, что нас интересуют не сами строки  $s$  и  $t$ , а некоторые их подстроки  $s[l_1 \dots r_1]$  и  $t[l_2 \dots r_2]$ .

Как вы уже, наверное, догадались, вам дано  $q$  запросов,  $i$ -й из которых задаёт некоторые подстроки  $\bar{s} = s[l_{1i} \dots r_{1i}]$  и  $\bar{t} = t[l_{2i} \dots r_{2i}]$ . Для каждого такого запроса необходимо посчитать количество вхождений строки  $\bar{s}$  в строку  $\bar{t}$ .

### Формат входных данных

Первая и вторая строки входных данных содержат строки  $s$  и  $t$  ( $1 \leq |s|, |t| \leq 2 \cdot 10^5$ ) соответственно. Строки состоят из маленьких букв английского алфавита.

В третьей строке задано одно число  $q$  ( $1 \leq q \leq 5 \cdot 10^5$ ) — количество запросов.

Каждая из следующих  $q$  строк содержит по четыре числа  $l_1, r_1, l_2$  и  $r_2$  ( $1 \leq l_1 \leq r_1 \leq |s|, 1 \leq l_2 \leq r_2 \leq |t|$ ), описывающих очередной запрос.

### Формат выходных данных

Выведите  $q$  чисел — ответы на запросы.

### Примеры

стандартный ввод	стандартный вывод
abb	3
ababababb	1
5	2
1 2 1 7	1
2 3 2 9	4
3 3 4 7	
1 2 2 4	
1 1 1 9	