

Задача А. Бинарные строки

Имя входного файла: `binary.in`
Имя выходного файла: `binary.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Строка называется *бинарной*, если она состоит только из символов '0' и '1'.

Рассмотрим бинарную строку w длины n . *Суффиксным массивом* строки w называется массив $a[1..n]$ такой, что строка $w[a[i]..n]$ является i -ым в лексикографическом порядке суффиксов строки w . Например, в результате сортировки суффиксов строки $w="001011"$ они будут расположены следующим образом: "001011", "01011", "011", "1", "1011", "11". Следовательно, суффиксный массив для строки w выглядит так: (1, 2, 4, 6, 3, 5).

Вам дан суффиксный массив a неизвестной строки w . Требуется восстановить строку w .

Формат входных данных

Первая строка входного файла содержит n — длину строки w ($1 \leq n \leq 300\,000$). Вторая строка содержит n различных целых чисел в диапазоне от 1 до n — суффиксный массив строки w .

Формат выходных данных

Выведите единственную строку — искомую бинарную строку w , суффиксный массив которой совпадает с массивом, заданным во входных данных. Если таких строк несколько, выведите любую из них. В случае, если таких строк не существует, выведите "Error".

Примеры

binary.in	binary.out
6	001011
1 2 4 6 3 5	

Задача В. Башни

Имя входного файла: `towers.in`
Имя выходного файла: `towers.out`
Ограничение по времени: 15 секунды
Ограничение по памяти: 256 мегабайт

Задано число n и последовательность из n чисел. Требуется рассмотреть все возможные циклические сдвиги заданной последовательности, отсортировать их в лексикографическом порядке, и вывести сумму наибольших общих префиксов соседних в этом порядке сдвигов.

Формат входных данных

Входной файл содержит не более 200 тестовых примеров. Каждый тестовый пример состоит из двух строк. Первая из них содержит целое число $1 \leq n \leq 50000$ — количество магических башен. Вторая строка содержит n чисел в интервале от 0 до 100 — заданную последовательность.

После последнего тестового примера вместо числа n идет 0.

Формат выходных данных

Для каждого тестового примера выведите одно число — искомую сумму.

Примеры

<code>towers.in</code>	<code>towers.out</code>
11 12 8 18 18 8 18 18 8 15 15 8 0	13

Задача С. Контрольное списывание

Имя входного файла: `kthsubstr.in`
Имя выходного файла: `kthsubstr.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Сегодня на уроке преподаватель Массивов Автомат Укконевич рассказывал своим ученикам про строки и всё такое. Например, он рассказал им, что строка A лексикографически меньше строки B , если либо A является префиксом B , либо B не является префиксом A , и при этом символ, стоящий в строке A на первой такой позиции, в которой эти строки различаются, имеет меньший номер в алфавите, чем символ на этой позиции в строке B .

Чтобы проверить понимание учениками нового материала, Автомат Укконевич выдал им следующее задание: среди всех различных непустых подстрок строки S найти k -ю в лексикографическом порядке.

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

Формат входных данных

В первой строке входного файла находится строка S ($|S| \leq 10^6$). Вторая строка содержит число k ($1 \leq k \leq 10^{18}$) — порядковый номер запрашиваемой подстроки.

Формат выходных данных

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

Примеры

<code>kthsubstr.in</code>	<code>kthsubstr.out</code>
<code>abacaba</code> <code>10</code>	<code>acab</code>
<code>abracadabra</code> <code>10000000000000000000</code>	<code>racadabra</code>