

Задача А. 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 входного файла

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 выходного файла

Для каждого тестового примера выведите одно число — искомую сумму.

Примеры

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

Задача В. 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 входного файла

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 выходного файла

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 1 2 4 6 3 5 | 001011 |

Задача С. Простые строки

Имя входного файла: `lyndon.in`
Имя выходного файла: `lyndon.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Простой строкой (или строкой Линдона) называется строка, которая лексикографически строго меньше всех своих циклических сдвигов. Например, “`abac`” является строкой Линдона, но “`abbab`” — не является.

Вам даны два числа, n и m . Рассмотрим все строки Линдона, состоящие не более, чем из n символов, каждый из которых является одной из первых m букв английского алфавита. Занумеруем их в лексикографическом порядке, начиная с 1. Ваша задача состоит в том, чтобы найти строки с номерами от l до r , включительно.

Формат входного файла

Входные данные состоят из одного или **нескольких** тестовых примеров. Каждый тестовый пример состоит из четырех натуральных чисел n , m , l и r ($1 \leq n \leq 30$; $2 \leq m \leq 26$; $1 \leq l \leq r \leq 10^7$; $r - l < 10^5$). Гарантируется, что существует хотя бы r строк Линдона из n символов, каждый из которых является одной из первых m букв английского алфавита.

Формат выходного файла

Для каждого тестового примера выведите его номер, а затем искомые строки Линдона, по одной на строке, упорядоченные лексикографически.

Примеры

| <code>lyndon.in</code> | <code>lyndon.out</code> |
|------------------------|---|
| 4 3 7 12 4 3 31 32 | Case 1: aac aacb aacc ab abac abb Case 2: bccc c |

Note

Существует 32 строки Линдона при $n = 4$ и $m = 3$: a, aaab, aaac, aab, aabb, aabc, aac, aacb, aacc, ab, abac, abb, abbb, abbc, abc, abcb, abcc, ac, acb, acbb, acbc, acc, acsb, accs, b, bbbc, bbc, bbcc, bc, bcc, bccc, c.