

Задача А. Башни

Имя входного файла: `towers.in`
Имя выходного файла: `towers.out`
Ограничение по времени: 15 секунд
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

towers.in	towers.out
11	13
12 8 18 18 8 18 18 8 15 15 8	
0	

Задача В. Рефрен

Имя входного файла: `refrain.in`
Имя выходного файла: `refrain.out`
Ограничение по времени: 2 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.

Примеры

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

Задача С. Двоичный суффиксный массив

Имя входного файла: `binary.in`
Имя выходного файла: `binary.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Двоичное слово — это слово, состоящее только из символов 0 и 1.

Рассмотрим двоичное слово w длины n . Суффиксный массивом слова w называется массива $a[1..n]$ такой, $w[a[i]..n]$ — i в лексикографическом порядке суффикс строки w . Например, пусть $w = \text{“001011”}$, тогда суффиксы строки w упорядочены следующим образом: “001011”, “01011”, “011”, “1”, “1011”, “11”, а суффиксный массив равен (1, 2, 4, 6, 3, 5). Вам дан суффиксный массив a двоичного слова w . Восстановите w .

Формат входных данных

Первая строка входных данных содержит число n — длина w ($1 \leq n \leq 300\,000$). Вторая строка содержит n различных целых чисел от 1 до n — суффиксный массив двоичного слова w .

Формат выходных данных

Выведите такое двоичное слово w , что его суффиксный массив совпадает с заданным во входных данных массивом. Если таких слов несколько, то разрешается вывести любое из них. Если не существует такого слова w , выведите “**Error**”.

Примеры

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