

Задача А. Суффиксный массив

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

Данна строка, требуется построить суффиксный массив для этой строки. Суффиксный массив — лексикографически отсортированный массив всех суффиксов строки. Каждый суффикс задается целым числом — позицией начала.

Строка s лексикографически меньше строки t , если есть такое i , что $s_i < t_i$ и $s_j = t_j$ для всех $j < i$. Или, если такого i не существует и строка s короче строки t .

Здесь s_i — код i -го символа строки s .

Формат входных данных

Файл состоит из единственной строки. Длина текста не превосходит 250 000. Коды всех символов в тексте от 32 до 127.

Формат выходных данных

Выведите N чисел — суффиксный массив данной строки.

Примеры

| <code>suffarray.in</code> | <code>suffarray.out</code> |
|---------------------------|----------------------------|
| 11 MISSISSIPPI | 11 8 5 2 1 10 9 7 4 6 3 |
| 5 AAAAA | 5 4 3 2 1 |

Задача В. Количество подстрок

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

Дана зацикленная строка S . Подсчитайте количество различных подстрок данной строки.

Зацикленная строка значит, что после последнего символа следует первый, таким образом у зацикленной строки "aba" есть подстроки "aa", "baa", "aab" но нет подстроки "baab", так как каждый символ должен входить не более одного раза в подстроку. Можете представлять, что строка записана на окружности, и в качестве подстроки подходит любая дуга.

Формат входных данных

Входной файл содержит одну строку, состоящую из строчных латинских букв ($1 \leq |S| \leq 100,000$).

Формат выходных данных

Выведите одно число — количество различных подстрок зацикленной строки S .

Примеры

| <code>subc.in</code> | <code>subc.out</code> |
|----------------------|-----------------------|
| aba | 8 |
| aaaaa | 5 |

Задача С. 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 D. 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

| <code>refrain.in</code> | <code>refrain.out</code> |
|-------------------------|--------------------------|
| 9 3 | 9 |
| 1 2 1 2 1 3 1 2 1 | 3 |
| | 1 2 1 |