

Задача А. Код Хаффмана

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

Заданы числа p_1, p_2, \dots, p_n .

Предположив, что имеется текст, содержащий p_1 символов c_1 , p_2 символов c_2 , и т. д., постройте код Хаффмана и найдите суммарное число битов, необходимое для кодирования такого текста.

Формат входного файла

Первая строка входного файла содержит число n ($2 \leq n \leq 1000$). Вторая строка содержит n целых чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq 10^9$).

Формат выходного файла

Выведите одно число — число битов, необходимое для кодирования текста с заданным во входном файле количеством вхождений каждого символа.

Пример

<code>huffman.in</code>	<code>huffman.out</code>
10 1 2 3 4 5 6 7 8 9 10	173

Задача В. Преобразование Барроуза-Уиллера

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

Реализуйте преобразование Барроуза-Уиллера.

Рассмотрим строку s , состоящую из строчных латинских букв.

Отсортируем в лексикографическом порядке все ее циклические сдвиги. Выпишем последние буквы получившихся строк в порядке сортировки.

Формат входного файла

Входной файл содержит строку, содержащую не более 1000 строчных букв латинского алфавита.

Формат выходного файла

Выведите результат преобразования Барроуза-Уиллера.

Пример

<code>bwt.in</code>	<code>bwt.out</code>
abacaba	bcabaaa

Задача С. Move To Front

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

Реализуйте преобразование МТФ.

Рассмотрим строку из строчных латинских букв.

Исходно буквы от 'a' до 'z' организованы в список в алфавитном порядке. По очереди рассматриваются слова из латинских букв. Для каждой буквы кодируемой строки выполняется следующее:

- Выводится ее номер в списке (нумерация с 1).
- Она перемещается на первую позицию в списке.

Формат входного файла

Входной файл содержит строку, содержащую не более 1000 строчных букв латинского алфавита.

Формат выходного файла

Пусть длина строки во входном файле равна n . Выведите n чисел от 1 до 26, которые будут выведены при преобразовании Move To Front.

Пример

<code>mtf.in</code>	<code>mtf.out</code>
abacaba	1 2 2 3 2 3 2

Задача D. Алгоритм LZW

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

Реализуйте кодирование в алгоритме LZW.

Рассмотрим строку s , состоящую из строчных латинских букв.

Исходно имеется словарь, содержащий символы от 'a' до 'z' с кодами от 0 до 25, соответственно.

Алгоритм поддерживает текущий буфер t , исходно инициализированный пустой строкой. Последовательно рассматриваются символы строки s . Пусть очередной символ строки равен c .

Если строка t есть в словаре, то t присваивается tc и обработка символа завершается.

Иначе выводится код t и строка tc помещается в словарь с минимальным свободным кодом. После этого t присваивается значение c и обработка символа завершается.

После просмотра всех символов код оставшегося t также выводится.

Формат входного файла

Входной файл содержит строку, содержащую не более 1000 строчных букв латинского алфавита.

Формат выходного файла

Выведите коды, которые выводятся по мере выполнения алгоритма.

Пример

<code>lzw.in</code>	<code>lzw.out</code>
abacaba	0 1 0 2 26 0