Задача 1. Частота встречаемости символов

Источник:	базовая
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Необходимо посчитать, сколько раз каждый символ встречается в текстовом файле.

Формат входных данных

Во входном файле записан некоторый непустой текст. Размер файла не превосходит 2.1 Mb.

Формат выходных данных

В выходной файл необходимо для каждого символа, который встречается в файле, в отдельной строке вывести информацию в следующем формате:

$a : b$

Здесь a — код символа в десятичном формате, b — положительное число, равное количеству этих символов в файле.

Символы выводить в порядке возрастания кодов.

Пример

input.txt	output.txt
кол около колокола	10 : 1 32 : 2 224 : 1 234 : 4 235 : 4 238 : 7

Задача 2. Частота встречаемости байтов

Источник:	базовая
Имя входного файла:	`input.bin`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Составить программу подсчета байтов в бинарном файле.

Формат входных данных

Во входном файле записана непустая последовательность байтов. Размер файла не превышает 2.1 Mb.

Формат выходных данных

В выходной файл необходимо для каждого символа, который встречается в файле, вывести в отдельной строке информацию в следующем формате:

$a : b$

Здесь a — код символа в десятичном формате, b — положительное число, равное количеству этих символов в файле.

Символы выводить в порядке возрастания кодов.

Примеры

input.bin	output.txt
EA EE EB 20 EE EA EE EB EE 20 EA EE EB EE EA EE EB E0 0D 0A	10 : 1 13 : 1 32 : 2 224 : 1 234 : 4 235 : 4 238 : 7

Задача 3. Коды Хаффмана для текста

Источник:	основная
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Необходимо построить коды Хаффмана для символов, которые встречаются в текстовом файле.

При построении дерева Хаффмана требуется придерживаться следующих правил:

1. Упорядочить символы, которые встречаются в файле по убыванию частот встречаемости. Если частоты совпадают, упорядочить эти символы между собой по возрастанию кодов.

2. При построении новой вершины в качестве левого сына брать предпоследнюю вершину в списке вершин, а в качестве правого — последнюю.

3. Затем эту новую вершину вставить в общий список (или ряд) так, чтобы ее частота встречаемости была строго меньше встречаемости предшествующей и больше либо равна встречаемости следующей за ней вершины.

4. При построении кодов левому сыну приписывать 0, а правому — 1.

В этом случае обеспечивается то свойство, что построенные коды символов с одинаковой частотой встречаемости будут упорядочены в лексико-графическом порядке.

Формат входных данных

Во входном файле записан некоторый непустой текст. Размер файла не превосходит 2.1 Mb.

Формат выходных данных

В выходной файл необходимо для каждого символа, который встречается в файле, в отдельной строке вывести информацию в следующем формате:

$a : s$

Здесь a — код символа в десятичном формате, s — строка, состоящая из нулей и единиц — код Хаффмана для этого символа.

Символы выводить в порядке возрастания кодов.

Далее необходимо вывести информацию о результатах кодирования: информационную емкость, длину входного файла, длину выходного кода и значение избыточности кода. Эти данные выводить в формате, указанном в примере. Вещественные числа выводить с 6 знаками после точки.

Пример

input.txt	output.txt
кол около колокола	10 : 0100 32 : 011 224 : 0101 234 : 10 235 : 11 238 : 00 Information capacity = 2.266270 Input file size = 19 Code length = 44 Redundancy = 0.021383

Задача 4. Коды Хаффмана для бинарного файла

Источник:	основная
Имя входного файла:	`input.bin`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Составить программу подсчета байтов в бинарном файле.

При построении дерева Хаффмана требуется придерживаться дисциплины, описанной в задаче 3.

Формат входных данных

Во входном файле записана непустая последовательность байтов. Размер файла не превышает 2.1 Mb.

Формат выходных данных

В выходной файл необходимо для каждого символа, который встречается в файле, в отдельной строке вывести информацию в следующем формате:

$a : s$

Здесь a — код символа в десятичном формате, s — строка, состоящая из нулей и единиц — код Хаффмана для этого символа.

Символы выводить в порядке возрастания кодов.

Далее необходимо вывести информацию о результатах кодирования: информационную емкость, длину входного файла, длину выходного кода и значение избыточности кодирования. Эти данные выводить в формате, указанном в примере. Вещественные числа выводить с 6 знаками после точки.

Примеры

input.bin
`EA EE EB 20 EE EA EE EB EE 20 EA EE EB EE EA EE EB E0 0D 0A`
output.txt

`10 : 0101`
`13 : 0110`
`32 : 0100`
`224 : 0111`
`234 : 10`
`235 : 11`
`238 : 00`
`Information capacity = 2.439354`
`Input file size = 20`
`Code length = 50`
`Redundancy = 0.024258`

Задача 5. Дерево - текст

Источник:	основная
Имя входного файла:	`input.bin`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Построить дерево Хаффмана для заданного бинарного файла.

При его построении требуется придерживаться дисциплины, описанной в задаче 3.

Формат входных данных

Во входном файле записана непустая последовательность байтов. Размер файла не превышает 2.1 Mb.

Формат выходных данных

В выходной файл необходимо вывести построенное дерево Хаффмана в текстовом виде.

Вершины построенного дерева Хаффмана выдавать в порядке префиксного обхода.

Если текущая вершина — не лист, то нужно выдать 0, в противном случае нужно выдать 1, а затем выдать последовательность из восьми литер (нулей и единиц) — код символа, которому соответствует данная вершина.

Пример

input.bin
EA EE EB 20 EE EA EE EB EE 20 EA EE EB EE EA EE EB E0 0D 0A
output.txt
00111101110001001000001000010100100001101111100000011110101011101011

Задача 6. Дерево - код

Источник:	основная
Имя входного файла:	`input.bin`
Имя выходного файла:	`output.bin`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Построить дерево Хаффмана для заданного бинарного файла.

При его построении требуется придерживаться дисциплины, описанной в задаче 3.

Формат входных данных

Во входном файле записана непустая последовательность байтов. Размер файла не превышает 2.1 Mb.

Формат выходных данных

В выходной файл необходимо вывести построенное дерево Хаффмана в бинарном виде.

Первый байт выходного файла должен содержать количество различных символов, которые встречаются во входном файле.

Далее должно быть записано дерево в порядке, описанном в задаче 5. Но вместо литеры ‘0’ или ‘1’ нужно использовать 1 бит. Если последний байт заполнен не полностью, то «лишние» биты в нем должны быть нулями.

Пример

input.bin
EA EE EB 20 EE EA EE EB EE 20 EA EE EB EE EA EE EB E0 0D 0A
output.bin
07 3D C4 82 14 86 F8 1E AF 58

Задача 7. Код - дерево

Источник:	основная
Имя входного файла:	`input.bin`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Восстановить дерево Хаффмана, записанное в заданном бинарном файле и выдать в текстовый файл таблицу кодов.

Формат входных данных

Во входном файле записано дерево Хаффмана в формате, описанном в задаче 6.

Формат выходных данных

В выходной файл необходимо вывести для каждого символа его код в формате, описанном в задаче 4.

Пример

input.bin
`07 3D C4 82 14 86 F8 1E AF 58`
output.txt

`10 : 0101`
`13 : 0110`
`32 : 0100`
`224 : 0111`
`234 : 10`
`235 : 11`
`238 : 00`

Задача 8. Архивация - текст

Источник:	основная
Имя входного файла:	`input.bin`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Составить программу архивации заданного файла. Код записать в текстовый файл в виде последовательности нулей и единиц.

Закодированный файл записывать в следующем формате.

Заголовок архива содержит длину входного файла (4 байта), количество различных символов (1 байт), дерево Хаффмана в формате, описанном в предыдущих задачах.

Далее идет закодированный входной файл. Последний байт как в дереве, так и в коде дополнить нулями, если он до конца не заполнен.

Формат входных данных

Во входном файле записана непустая последовательность байтов. Размер файла не превышает 2.1 Mb.

Формат выходных данных

В выходной файл необходимо вывести последовательность нулей и единиц, соответствующую заархивированному входному файлу.

Примеры

input.bin
EA EE EB 20 EE EA EE EB EE 20 EA EE EB EE EA EE EB E0 0D 0A
output.txt
TODO

Задача 9. Архивация - код

Источник:	основная
Имя входного файла:	`input.bin`
Имя выходного файла:	`output.bin`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Составить программу архивации заданного файла методом Хаффмана. Код записать в бинарный файл.

Закодированный файл записывать в следующем формате.

Заголовок архива содержит длину входного файла (4 байта), количество различных символов (1 байт), дерево Хаффмана в формате, описанном в предыдущих задачах.

Далее идет закодированный входной файл. Последний байт как в дереве, так и в коде дополнить нулями, если он до конца не заполнен.

Формат входных данных

Во входном файле записана непустая последовательность байтов. Размер файла не превышает 2.1 Mb.

Формат выходных данных

В выходной файл необходимо вывести заархивированный входной файл.

Примеры

input.bin
EA EE EB 20 EE EA EE EB EE 20 EA EE EB EE EA EE EB E0 0D 0A
output.bin
TODO

Задача 10. Декодирование

Источник:	основная
Имя входного файла:	`input.bin`
Имя выходного файла:	`output.bin`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Составить программу декодирования заархивированного файла методом Хаффмана. Результат записать в бинарный файл.

Закодированный файл записывать в следующем формате.

Заголовок архива содержит длину входного файла (4 байта), количество различных символов (1 байт), дерево Хаффмана в формате, описанном в предыдущих задачах.

Далее идет закодированный входной файл. Последний байт как в дереве, так и в коде дополнить нулями, если он до конца не заполнен.

Формат входных данных

Во входном файле записан закодированный файл в формате, описанном в предыдущей задаче

Формат выходных данных

В выходной файл необходимо вывести разархивированный входной файл.

Примеры

input.bin
TODO
output.bin
EA EE EB 20 EE EA EE EB EE 20 EA EE EB EE EA EE EB E0 0D 0A