// C++ · Лабораторная 07

Файлы

Обработка числового файла, текстовой матрицы и подсчёт строк и слов в тексте.

Цель и порядок работы

Освоить основные алгоритмы работы с типизированными и текстовыми файлами.

Выполните три задания своего варианта: числовой файл, матрица либо дополнительное файловое преобразование, затем подсчёт строк и слов. Размеры файлов определяйте при чтении. Каждое задание оформите отдельной функцией.

1. Считайте размеры создаваемых файлов заранее неизвестными и определяйте их при чтении.
2. При чтении из файла не используйте массивы. Рабочие файлы использовать разрешено.
3. Сохраните условия своего варианта: терминатор ввода, способ формирования исходных данных и требуемое преобразование.
4. Составьте схему алгоритма, реализуйте программу C++, выполните компиляцию и отладку.
5. Проведите структурное тестирование на обычных, граничных и некорректных входах, проверьте решения и логические условия.
6. Покажите содержимое файлов до и после обработки и оформите отчёт.

Требования к коду, заготовка отчёта, тестирование и блок-схемы →

Скачать текущие задания и варианты+ в PDF

Исходные материалы: Методичка №8 · PDF

Перед началом: Лекция 10 · файловые потоки · Лекция 1 · ввод и вывод · Лекция 2 · циклы · Лекция 4 · строки

> Используйте стандартные файловые потоки C++. Чтение проверяется по результату операции, а не предварительной проверкой eof(). Сначала изучите раздел о файлах по ссылке ниже.

Задания по вариантам

Выполните все задания назначенного варианта.

Общие требования к вариантам

Во всех вариантах выполните три задания по своим заголовкам. В вариантах 1–15 второе задание — матрица в текстовом файле; в вариантах+ вместо него дано дополнительное файловое преобразование. Третье задание общее: строки и слова в текстовом файле.

Размеры файлов определяйте при чтении. Весь числовой файл и всю матрицу в массив не загружайте: используйте накопители, повторные проходы и рабочие файлы. Для текстового задания разрешена одна текущая строка std::string. Исходник сохраняйте до успешного завершения и проверки результата.

«Типизированный файл» в этой работе — двоичный файл значений int одной и той же программы и платформы. Используйте std::ifstream / std::ofstream с std::ios::binary и проверяйте успех read / write. Не рассчитывайте на переносимость его байтового формата между платформами.

Для текстового файла матрицы: одна строка файла — одна строка матрицы, числа разделены пробелами. Проверяйте одинаковое количество столбцов. Размеры исходной генерируемой матрицы задавайте константами вне main; случайное заполнение можно заменить ручным для воспроизводимого теста.

Индексы и позиции записей начинаются с 0. При равных экстремумах выбирайте первое вхождение; вторые экстремумы — вторые различные значения. Границы удаляемого участка не удаляются, если это не сказано отдельно. Если нужных значений нет, сообщите об этом и сохраните файл.

Вариант 1

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Определить минимальное и максимальное числа и их позиции в файле (нумерация данных в файле начинается с нуля). Предусмотреть случай, когда таких чисел в файле нет. Удалить из файла все числа, находящиеся между минимальным и максимальным.

Задание 2

Создать текстовый файл, в который построчно записать квадратный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размер массива n задать константой. Границы диапазона v_min и v_max задать с консоли. Прочитать из созданного файла k-ю строку и k-й столбец и вывести их на экран. Значение k задать с консоли. При чтении из файла считать размер массива неизвестным. Определить его при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант_2

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Определить минимальное и максимальное числа и их позиции в файле (нумерация данных в файле начинается с нуля). Предусмотреть случай, когда таких чисел в файле нет. Оставить в файле только числа, находящиеся между минимальным и максимальным.

Задание 2

Создать текстовый файл, в который построчно записать квадратный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размер массива n задать константой. Границы диапазона v_min и v_max задать с консоли. Прочитать из созданного файла все диагонали, параллельные главной, и вывести их на экран построчно. При чтении из файла считать размер массива неизвестным. Определить его при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант 3

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Определить вторые по значению минимальное и максимальное числа и их позиции в файле (нумерация данных в файле начинается с нуля). Предусмотреть случай, когда таких чисел в файле нет. Оставить в файле только числа, находящиеся между ними.

Задание 2

Создать текстовый файл, в который построчно записать квадратный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размер массива n задать константой. Границы диапазона v_min и v_max задать с консоли. Прочитать из созданного файла все диагонали, параллельные

побочной, и вывести их на экран построчно. При чтении из файла считать размер массива неизвестным. Определить его при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант 4

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Определить вторые по значению минимальное и максимальное числа и их позиции в файле (нумерация данных в файле начинается с нуля). Предусмотреть случай, когда таких чисел в файле нет. Удалить из файла все числа, находящиеся между ними.

Задание 2

Создать текстовый файл, в который построчно записать прямоугольный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размеры массива n и m задать константами. Границы диапазона v_min и v_max задать с консоли. Прочитать из созданного файла r-ю строку и p-й столбец и вывести их на экран. Значения r и p задать с консоли. При чтении из файла считать размеры массива неизвестными. Определить их при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять

не нужно.

Вариант 5

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Определить первое по порядку четное и последнее нечетное числа и их позиции в файле (нумерация данных в файле начинается с нуля). Предусмотреть случай, когда таких чисел в файле нет. Удалить из файла все числа, находящиеся между ними.

Задание 2

Создать текстовый файл, в который построчно записать прямоугольный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размеры массива n и m задать константами. Границы диапазона v_min и v_max задать с консоли. Вывести на экран строку с максимальным элементом и столбец с минимальным элементом файла. При чтении из файла считать размеры массива неизвестными. Определить их при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант 6

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Определить первое по порядку нечетное и последнее четное числа и их позиции в файле (нумерация данных в файле начинается с нуля). Предусмотреть случай, когда таких чисел в файле нет. Удалить из файла все числа, находящиеся между ними.

Задание 2

Создать текстовый файл, в который построчно записать прямоугольный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размеры массива n и m задать константами. Границы диапазона v_min и v_max задать с консоли. Вывести на экран строку с минимальным элементом и столбец с максимальным элементом файла. При чтении из файла считать размеры массива неизвестными. Определить их при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант 7

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Определить первое по порядку положительное и последнее отрицательное числа и их позиции в файле (нумерация

данных в файле начинается с нуля). Предусмотреть случай, когда таких чисел в файле нет. Оставить в файле только числа, находящиеся между ними.

Задание 2

Создать текстовый файл, в который построчно записать прямоугольный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размеры массива n и m задать константами. Границы диапазона v_min и v_max задать с консоли. Вывести на экран строку с максимальной суммой элементов и столбец с минимальной суммой элементов. При чтении из файла считать размеры массива неизвестными. Определить их при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант 8

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Определить первое по порядку отрицательное и последнее положительное числа и их позиции в файле (нумерация данных в файле начинается с нуля). Предусмотреть случай, когда таких чисел в файле нет. Оставить в файле только числа, находящиеся между ними.

Задание 2

Создать текстовый файл, в который построчно записать прямоугольный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размеры массива n и m задать

константами. Границы диапазона v_min и v_max задать с консоли. Вывести на экран строку с минимальной суммой элементов и столбец с максимальной суммой элементов. При чтении из файла считать размеры массива неизвестными. Определить их при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант 9

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Определить первое по порядку положительное и последнее отрицательное числа и их позиции в файле (нумерация данных в файле начинается с нуля). Предусмотреть случай, когда таких чисел в файле нет. Удалить из файла все числа, находящиеся между ними.

Задание 2

Создать текстовый файл, в который построчно записать прямоугольный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размеры массива n и m задать константами. Границы диапазона v_min и v_max задать с консоли. Определить максимальный и минимальный элементы в файле и поменять их местами. При чтении из файла считать размеры массива неизвестными. Определить их при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант 10

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Определить первое по порядку отрицательное и последнее положительное числа и их позиции в файле (нумерация данных в файле начинается с нуля). Предусмотреть случай, когда таких чисел в файле нет. Удалить из файла все числа, находящиеся между ними.

Задание 2

Создать текстовый файл, в который построчно записать прямоугольный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размеры массива n и m задать константами. Границы диапазона v_min и v_max задать с консоли. Определить второй максимальный и второй минимальный элементы в файле и поменять их местами. При чтении из файла считать размеры массива неизвестными. Определить их при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант 11

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Отсортировать файл по возрастанию.

Задание 2

Создать текстовый файл, в который построчно записать прямоугольный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размеры массива n и m задать константами. Границы диапазона v_min и v_max задать с консоли. Обнулить все элементы файла, находящиеся между его максимальным и минимальным элементами. При чтении из файла считать размеры массива неизвестными. Определить их при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант 12

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Отсортировать файл по убыванию.

Задание 2

Создать текстовый файл, в который построчно записать прямоугольный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размеры массива n и m задать константами. Границы диапазона v_min и v_max задать с консоли. Определить первый четный и последний нечетный элементы в файле и поменять их местами. При чтении из файла считать размеры массива неизвестными. Определить их при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант 13

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Перевернуть элементы файла – сделать “поплавок”.

Задание 2

Создать текстовый файл, в который построчно записать прямоугольный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размеры массива n и m задать константами. Границы диапазона v_min и v_max задать с консоли. Определить первый положительный и последний отрицательный элементы в файле и поменять их местами. При чтении из файла считать размеры массива неизвестными. Определить их при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант 14

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Удалить из файла его максимальный и минимальный элементы.

Задание 2

Создать текстовый файл, в который построчно записать прямоугольный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размеры массива n и m задать константами. Границы диапазона v_min и v_max задать с консоли. Определить первый нечетный и последний четный элементы в файле и поменять их местами. При чтении из файла считать размеры массива неизвестными. Определить их при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант 15

Задание 1

Создать типизированный файл целых чисел. Записать в него с консоли последовательность целых чисел, конец последовательности (терминатор) – ноль, который не записывается в файл. Удалить из файла вторые по величине максимальный и минимальный элементы.

Задание 2

Создать текстовый файл, в который построчно записать прямоугольный массив matrix, заполненный целыми случайными числами в диапазоне от v_min до v_max. Размеры массива n и m задать константами. Границы диапазона v_min и v_max задать с консоли. Определить первый отрицательный и последний положительный элементы в файле и поменять их местами. При чтении из файла считать размеры массива неизвестными. Определить их при чтении файла.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте

std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Вариант 16+ · Фильтрация и протокол

Числовой файл

Создайте двоичный файл целых чисел из ввода до 0, не записывая терминатор. Потоково перепишите в новый файл только значения из введённого закрытого интервала. Исходный файл сохраняйте.

Дополнение+

Выведите количество прочитанных, сохранённых и отброшенных записей, сумму сохранённых. Сравните количество записей при повторном чтении результата. Массивы для чтения не используйте.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Дополнение для варианта+: найдите самую длинную строку и выведите её длину и содержимое. При равной длине выбирайте первую. Для этой проверки используйте ASCII-текст; длину определяйте через size(). Если файл пуст, сообщите, что строк нет.

Вариант 17+ · Серии одинаковых записей

Числовой файл

Создайте файл целых чисел. Найдите максимальную длину подряд идущих одинаковых значений и позицию начала первой такой серии. Достаточно одного последовательного прохода.

Дополнение+

Создайте текстовый файл пар «значение длина» для всех серий. Затем восстановите второй числовой файл и последовательным сравнением убедитесь в совпадении с исходным. Проверьте пустой файл.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Дополнение для варианта+: найдите самую длинную строку и выведите её длину и содержимое. При равной длине выбирайте первую. Для этой проверки используйте ASCII-текст; длину определяйте через size(). Если файл пуст, сообщите, что строк нет.

Вариант 18+ · Слияние файлов

Два потока

Создайте два неубывающих файла целых чисел. Проверьте порядок чтением, затем слейте их в третий файл без загрузки в массив. В памяти храните только текущую запись каждого входа.

Дополнение+

Записывайте каждое различное значение один раз и выведите количество удалённых повторов. Исходные файлы сохраняйте; ошибки открытия и неполную двоичную запись считайте отказом.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Дополнение для варианта+: найдите самую длинную строку и выведите её длину и содержимое. При равной длине выбирайте первую. Для этой проверки используйте ASCII-текст; длину определяйте через size(). Если файл пуст, сообщите, что строк нет.

Вариант 19+ · Матрица в текстовом файле

Потоковая статистика

Прочитайте неизвестную заранее прямоугольную таблицу целых чисел: строки разделены переводом строки, числа — пробелами. Проверьте одинаковое положительное число столбцов без массива; при необходимости делайте повторные проходы.

Дополнение+

В новый текстовый файл запишите таблицу, в которой строка с наибольшей суммой переставлена в начало. Остальные строки сохраняют порядок. При равенстве сумм выберите первую. Исходный файл не перезаписывайте.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая

последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Дополнение для варианта+: найдите самую длинную строку и выведите её длину и содержимое. При равной длине выбирайте первую. Для этой проверки используйте ASCII-текст; длину определяйте через size(). Если файл пуст, сообщите, что строк нет.

Вариант 20+ · Безопасное преобразование и сверка

Рабочий файл

Создайте числовой файл и удалите все записи строго между первыми вхождениями минимума и максимума через запись результата в отдельный рабочий файл. При совпадении экстремумов оставьте данные без изменений.

Дополнение+

Не заменяйте исходник: проверьте чтение и закрытие результата, затем отдельным проходом проверьте число записей и сохранность обоих экстремумов. Выведите полный протокол. При любой ошибке исходный файл должен остаться доступным.

Задание 3 · Строки и слова в файле

Создайте текстовый файл из нескольких строк. Имя файла введите с клавиатуры. Напишите отдельную функцию, которая открывает файл и считает общее количество строк и слов. Слово — непустая последовательность символов между пробелами, табуляцией или переводами строк; числа тоже считаются словами. Читайте файл последовательно, не сохраняя весь текст в массиве.

Пустая строка учитывается в количестве строк, но не добавляет слов. Для пустого файла оба результата равны 0. Если файл открыть не удалось, выведите понятное сообщение. Для строки используйте std::getline, для выделения слов из строки — std::istringstream из <sstream>, как в разделе лекции о файлах.

Пример: файл из двух строк «Hello world» и «Welcome to C++» содержит 2 строки и 5 слов. Проверьте также пустой файл, одну пустую строку и несколько пробелов между словами. Исходный файл менять не нужно.

Дополнение для варианта+: найдите самую длинную строку и выведите её длину и содержимое. При равной длине выбирайте первую. Для этой проверки используйте ASCII-текст; длину определяйте через size(). Если файл пуст, сообщите, что строк нет.

Возможные вопросы к защите

Попробуйте ответить своими словами и показать соответствующее место в своей программе. Под вопросом можно открыть ориентир для ответа и перейти к нужной теме лекции.

▼ Почему цикл чтения должен зависеть от успешности операции?

Сравните проверку результата чтения и предварительную проверку eof(). Покажите отличие чистого конца от неверного формата.

Повторить: Лекция 10 · файловые потоки

▼ Как обработать файл неизвестной длины без массива?

Опишите последовательный обход с накопителями; при необходимости используйте повторный проход или рабочий файл.

Повторить: Лекция 2 · циклы · Лекция 1 · переменные

▼ Как посчитать строки и слова в текстовом файле?

Строки читайте через std::getline, слова — через строковый поток. Пустая строка увеличивает только счётчик строк; для пустого файла оба счётчика равны нулю.

Повторить: Лекция 10 · файловые потоки

▼ Чем номер записи отличается от позиции потока?

Объясните, почему tellg() не является порядковым номером числа в текстовом файле; покажите отдельный счётчик записей.

Повторить: Лекция 10 · файловые потоки

▼ Как проверить сохранение файла и что происходит при раннем return?

Объясните автоматическое закрытие потока и явную проверку после close(). Автоматическое освобождение само по себе не подтверждает успешную запись.

Повторить: Лекция 10 · файловые потоки · Лекция 9 · владение ресурсами

▼ Какие проверки нужны при обработке строк и таблицы из файла?

Проверьте формат чисел, число строк и столбцов, ошибки чтения и границы индексов.

Повторить: Лекция 4 · строки · Лекция 6 · матрицы
