Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа № 3

По дисциплине «Основы алгоритмизации и программирования»

На тему «Работа с файлами на языке С++»

Выполнил:

Студент 1 курса 10 группы

Жамойдо Артём Игоревич

Преподаватель: асс. Андронова М.В.

Минск, 2024

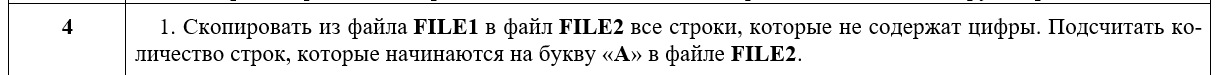
В соответствии со своим вариантом разработать программы для работы с файлами на ***языке С++.***

Для первой программы необходимо предварительно создать текстовый файл FILE1из нескольких строк и записать в него данные.

Во второй программе ввод информации с клавиатуры и вывод в консольное окно осуществить в главной функции, а запись в файл и чтение из файла − в функциях пользователя.

Встроенные функции для работы со строками не использовать.

**Вариант 4**



#include <iostream>

#include <fstream>

#include <cctype>

#include <string>

using namespace std;

int main()

{

setlocale(0, "ru");

// Открытие файлов

ifstream file1("FILE1.txt");

ofstream file2("FILE2.txt");

if (!file1.is\_open() || !file2.is\_open()) // Проверяем, удалось ли открыть файлы

{

cerr << "Не удалось открыть файлы." << endl;

exit(1); // Завершить программу с кодом ошибки 1

}

string line; // Функция для копирования строк без цифр из одного файла в другой

int count = 0;

while (getline(file1, line)) // Проверка на содержание цифр

{

bool hasDigit = false;

for (char c : line) // Проверка каждого символа в строке

{

if (isdigit(c))

{

hasDigit = true;

break;

}

}

if (!hasDigit) // Если строка не содержит цифр, копируем ее в файл FILE2

{

file2 << line << endl;

if (!line.empty() && toupper(line[0]) == 'A') // Подсчет строк, начинающихся на букву 'A'

{

count++;

}

}

}

//Закрытие файлов

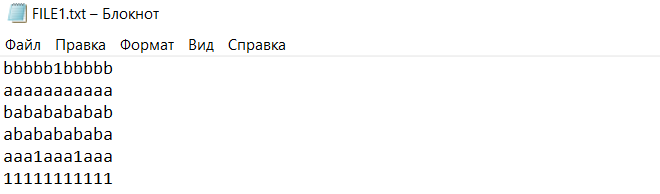
file1.close();

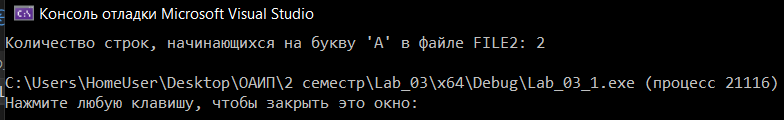
file2.close();

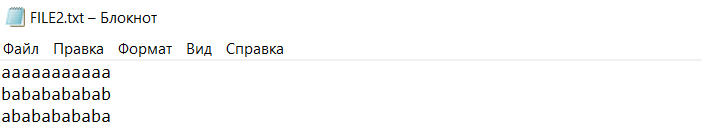
cout << "Количество строк, начинающихся на букву 'A' в файле FILE2: " << count << endl;

return 0;

}









#include <iostream>

#include <fstream>

#include <sstream>

#include <string>

using namespace std;

//Прототипы

void writeText(const string& text);

int findLongestWord(const string& file);

void writeText(const string& text) // Функция для записи текста в файл

{

ofstream outFile("file.txt");

if (!outFile.is\_open())

{

cerr << "Не удалось открыть файл для записи." << endl;

exit(1); // Завершить программу с кодом ошибки 1

}

outFile << text; // Записать текст в файл

outFile.close(); // Закрыть файл

}

int findLongestWord(const string& file) // Функция для поиска длины самого длинного слова в файле

{

ifstream inFile(file);

if (!inFile.is\_open())

{

cerr << "Не удалось открыть файл для чтения." << endl;

exit(1); // Завершить программу с кодом ошибки 1

}

string line;

string word;

int maxLength = 0;

while (inFile >> line) // Чтение из файла по словам и нахождение самого длинного слова

{

istringstream iss(line); // Создание потока для разбора строки на слова

while (iss >> word)

{

int length = word.length(); // Длина текущего слова

if (length > maxLength)

{

maxLength = length;

}

}

}

inFile.close(); // Закрыть файл

return maxLength;

}

int main()

{

setlocale(0, "ru");

string inputString;

cout << "Введите строку: ";

getline(cin, inputString); // Считать строку от пользователя

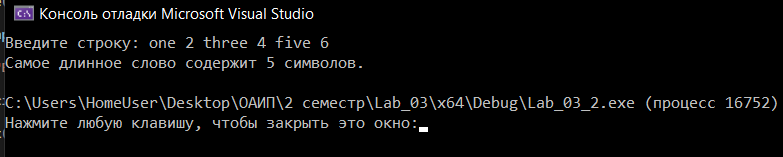
writeText(inputString); // Записать введенную строку в файл

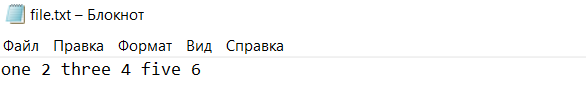
int maxLength = findLongestWord("file.txt");

cout << "Самое длинное слово содержит " << maxLength << " символов." << endl;

return 0;

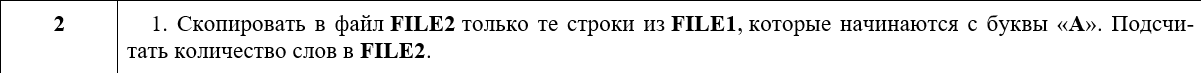
}





Доп. задачи

**Вариант 2**

****

#include <iostream>

#include <fstream>

#include <string>

#include <sstream>

using namespace std;

int main()

{

setlocale(0, "ru");

// Открытие файлов

ifstream file1("FILE1.txt");

ofstream file2("FILE2.txt");

if (!file1.is\_open() || !file2.is\_open()) // Проверяем, удалось ли открыть файлы

{

cerr << "Не удалось открыть файлы." << endl;

exit(1); // Завершить программу с кодом ошибки 1

}

string line; // Функция для копирования строк из одного файла в другой

string word;

int count = 0;

while (getline(file1, line))

{

if (!line.empty() && toupper(line[0]) == 'A') // Проверка строк, начинающихся на букву 'A'

{

file2 << line << endl;

istringstream iss(line); // Создание потока для разбора строки на слова

while (iss >> word)

{

count++;

}

}

}

//Закрытие файлов

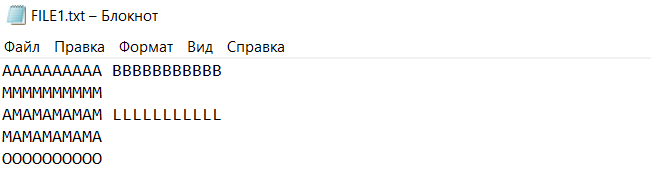
file1.close();

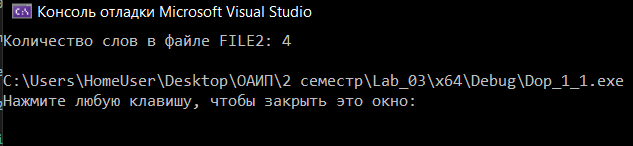
file2.close();

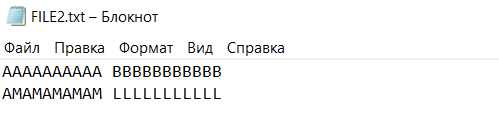
cout << "Количество слов в файле FILE2: " << count << endl;

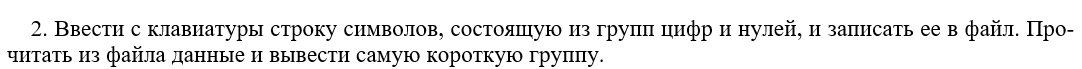
return 0;

}

****

****

****

****

#include <iostream>

#include <fstream>

#include <sstream>

#include <string>

#include <climits>

using namespace std;

// Прототипы

void writeText(const string& text);

string findShortestGroup(const string& file);

// Функция для записи строки в файл

void writeText(const string& text)

{

ofstream outFile("file.txt");

if (!outFile.is\_open())

{

cerr << "Не удалось открыть файл для записи." << endl;

exit(1); // Завершить программу с кодом ошибки 1

}

outFile << text; // Записать строку в файл

outFile.close(); // Закрыть файл

}

// Функция для поиска самой короткой группы в файле

string findShortestGroup(const string& file)

{

ifstream inFile(file);

if (!inFile.is\_open())

{

cerr << "Не удалось открыть файл для чтения." << endl;

exit(1); // Завершить программу с кодом ошибки 1

}

string line;

string shortestGroup;

int minLength = INT\_MAX; // Инициализируем длину самой короткой группы значением INT\_MAX

while (inFile >> line) // Чтение строки из файла

{

int currentLength = 0; // Длина текущей группы

string currentGroup;

istringstream iss(line); // Создание потока для разбора строки

char currentChar;

while (iss >> currentChar) // Читаем данные по одному символу из файла

{

if (currentChar == '0' || currentChar == '1')

{

currentLength++;

currentGroup += currentChar;

}

else

{

if (currentLength < minLength && !currentGroup.empty())

{

minLength = currentLength;

shortestGroup = currentGroup;

}

currentLength = 0;

currentGroup = "";

}

}

if (currentLength < minLength && !currentGroup.empty())

{

minLength = currentLength;

shortestGroup = currentGroup;

}

}

inFile.close(); // Закрыть файл

return shortestGroup;

}

int main()

{

setlocale(0, "ru");

string inputString;

cout << "Введите строку из групп цифр и нулей: ";

getline(cin, inputString);

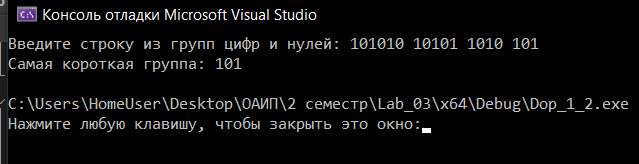
writeText(inputString); // Записать строку в файл

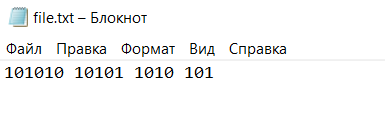
string shortestGroup = findShortestGroup("file.txt"); // Найти самую короткую группу в файле

cout << "Самая короткая группа: " << shortestGroup << endl;

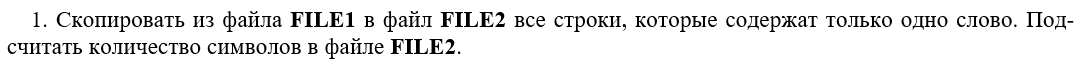
return 0;

}





**Вариант 11**



#include <iostream>

#include <fstream>

#include <cstring>

#include <sstream>

using namespace std;

int main()

{

setlocale(0, "ru");

// Открытие файлов

ifstream inFile("FILE1.txt");

ofstream outFile("FILE2.txt");

if (!inFile.is\_open() || !outFile.is\_open()) // Проверяем, удалось ли открыть файлы

{

cerr << "Не удалось открыть файлы." << endl;

exit(1); // Завершаем программу с ошибкой

}

string line; // Переменная для хранения строки из файла

int charCount = 0;

while (getline(inFile, line)) // Читаем файл по строкам

{

bool oneWord = true;

for (char c : line) // Проверяем каждый символ в строке

{

if (c == ' ') // Если найден пробел, значит это не одно слово

{

oneWord = false;

break;

}

}

// Если строка состоит из одного слова, записываем ее в выходной файл

if (oneWord)

{

outFile << line << endl;

charCount += line.length(); // Увеличиваем общее количество символов

}

}

cout << "Количество символов в FILE2: " << charCount << endl;

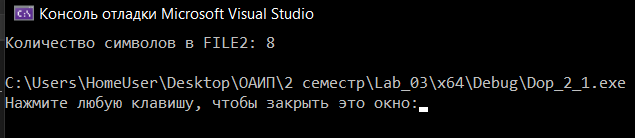
// Закрытие файлов

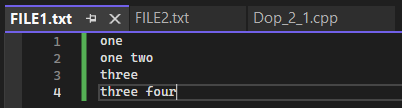
inFile.close();

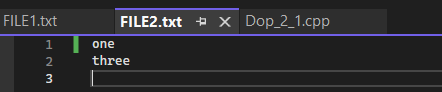
outFile.close();

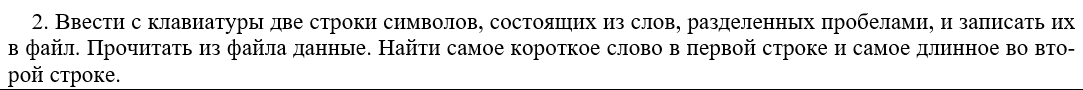
return 0;

}









#include <iostream>

#include <fstream>

#include <cstring>

#include <sstream>

using namespace std;

// Прототипы

void writeText(const string& str1, const string& str2);

void findWords(string& shortestWord, string& longestWord);

// Функция для записи строк в файл

void writeText(const string& str1, const string& str2)

{

ofstream outFile("file.txt");

// Проверяем удалось ли открыть файл

if (!outFile.is\_open())

{

cerr << "Не удалось открыть файл для записи." << endl;

exit(1);

}

outFile << str1 << endl;

outFile << str2 << endl;

// Закрываем файл

outFile.close();

}

// Функция для чтения строк из файла

void findWords(string& shortestWord, string& longestWord)

{

ifstream inFile("file.txt");

// Проверяем удалось ли открыть файл

if (!inFile.is\_open())

{

cerr << "Не удалось открыть файл для чтения." << endl;

exit(1);

}

string line1, line2;

// Читаем строки из файла

if (getline(inFile, line1) && getline(inFile, line2))

{

// Находим самое короткое слово в первой строке

istringstream iss1(line1);

iss1 >> shortestWord;

string word;

while (iss1 >> word)

{

if (word.length() < shortestWord.length())

{

shortestWord = word;

}

}

// Находим самое длинное слово во второй строке

istringstream iss2(line2);

iss2 >> longestWord;

while (iss2 >> word)

{

if (word.length() > longestWord.length())

{

longestWord = word;

}

}

}

// Закрываем файл

inFile.close();

}

int main()

{

setlocale(0, "ru");

string input1, input2;

cout << "Введите первую строку: ";

getline(cin, input1);

cout << "Введите вторую строку: ";

getline(cin, input2);

writeText(input1, input2); // Запись строк в файл

string shortestWord, longestWord;

// Чтение строк из файла и поиск самых короткого и длинного слова

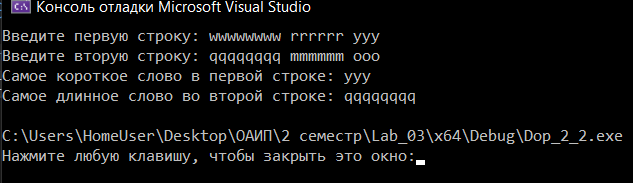
findWords(shortestWord, longestWord);

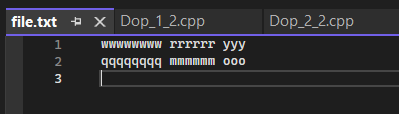
cout << "Самое короткое слово в первой строке: " << shortestWord << endl;

cout << "Самое длинное слово во второй строке: " << longestWord << endl;

return 0;

}





**Вариант 1**



#include <iostream>

#include <fstream>

#include <sstream>

using namespace std;

int main()

{

setlocale(0, "ru");

// Открываем файлы

ifstream inFile("FILE1.txt");

ofstream outFile("FILE2.txt");

// Проверяем удалось ли открыть файл

if (!inFile.is\_open() || !outFile.is\_open())

{

cerr << "Ошибка открытия файлов." << endl;

exit(1);

}

string line;

int lineCount = 0;

while (getline(inFile, line)) // Читаем каждую строку из FILE1

{

lineCount++;

if (lineCount % 2 == 0) // Проверяем четность строки

{

outFile << line << endl;

}

}

// Закрытие файлов

inFile.close();

outFile.close();

// Подсчет размера файлов

ifstream inFile2("FILE1.txt", ios::binary | ios::ate);

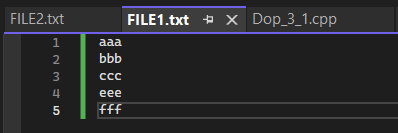
ifstream outFile2("FILE2.txt", ios::binary | ios::ate);

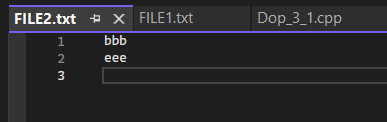
cout << "Размер файла FILE1 в байтах: " << inFile2.tellg() << endl;

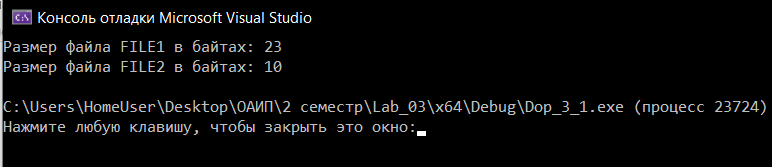
cout << "Размер файла FILE2 в байтах: " << outFile2.tellg() << endl;

return 0;

}









#include <iostream>

#include <fstream>

#include <sstream>

using namespace std;

// Прототипы

void writeText(const string& data, const string& filename);

string readOddNumbers(const string& filename);

// Функция для записи строки в файл

void writeText(const string& data, const string& filename)

{

ofstream outFile(filename);

if (!outFile.is\_open())

{

cerr << "Ошибка открытия файла." << endl;

exit(1);

}

outFile << data; // Записываем данные в файл

outFile.close();

}

// Функция для чтения данных из файла

string readOddNumbers(const string& filename)

{

ifstream inFile(filename); // Открываем файл для чтения

if (!inFile.is\_open())

{

cerr << "Ошибка открытия файла." << endl;

exit(1);

}

string data;

getline(inFile, data); // Читаем строку из файла

string result = "";

string number;

istringstream iss(data); // Создаем поток для чтения данных

// Читаем числа из строки и проверяем на четность

while (iss >> number)

{

// Проверяем, содержит ли слово только цифры

if (number.find\_first\_not\_of("0123456789") == string::npos)

{

int num = stoi(number);

if (num % 2 != 0)

{

result += number + " "; // Добавляем нечетное число к результату

}

}

}

inFile.close();

return result;

}

int main()

{

setlocale(0, "ru");

string input;

cout << "Введите строку символов: ";

getline(cin, input);

writeText(input, "file.txt"); // Записываем введенную строку в файл

string oddNumbers = readOddNumbers("file.txt"); // Читаем нечетные числа из файла

if (!oddNumbers.empty())

{

cout << "Нечетные числа строки: " << oddNumbers << endl;

}

else

{

cout << "В строке нет нечетных чисел." << endl;

}

return 0;

}

