Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 18

По дисциплине «Основы алгоритмизации и программирования»

На тему «Работа с файлами на языке С»

Выполнил:

Студент 1 курса 9 группы

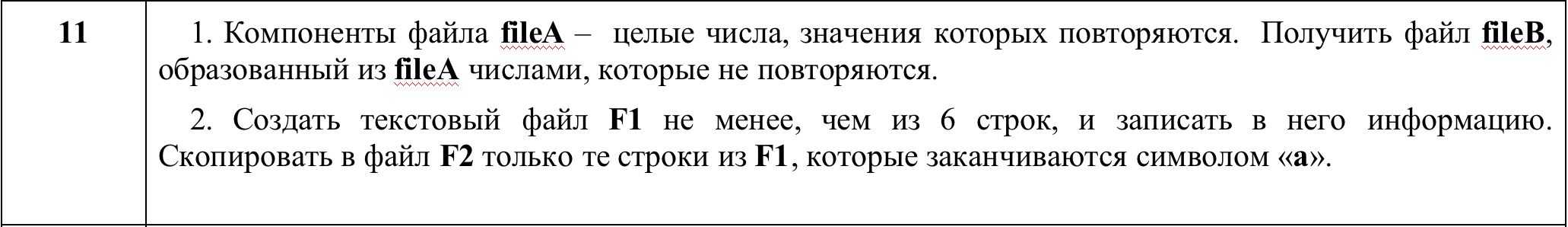
Павлович Ян Андреевич

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 11

В соответствии со своим вариантом разработать программы для условий, приведенных в таблице ниже, и изучить способы работы с файлами на ***языке С***.



1. Компоненты файла **fileA** –  целые числа, значения которых повторяются.  Получить файл **fileB**, образованный из **fileA** числами, которые не повторяются.

#include <iostream>

#include <Windows.h>

#include <stdio.h>

**int** main() {

    setlocale(LC\_ALL, "Russian"); //добавляем возможность работы с русским языком

    FILE\* fileA; //создание указателя на файл

    errno\_t errA;//переменная для ошибки

    errA = fopen\_s(&fileA, "fileA.txt", "r"); //открытие файла для чтения

**if** (errA != 0) { //если ошибка

        perror("Ошибка открытия файла!");

**return** 1;

    }

**int** unique\_numbers[100]; //создание массива для хранения единственного вхождения каждого числа

**int** num\_unique = 0; //создание переменной для вычисления кол-ва никальных чисел

**int** number; //создание переменной для текущего чистаемого элмента файла

**while** (fscanf\_s(fileA, "%d", &number) == 1) { //цикл для чтения каждого элемента файла

**int** isDuplicate = 0; //созжание контрольной переменной

**for** (**int** i = 0; i < num\_unique; i++) { //цикл для проверки вхождения уникальных чисел

**if** (unique\_numbers[i] == number) {

                isDuplicate = 1; //если найдено повторное фхождение

**break**;

            }

        }

**if** (!isDuplicate) { //добавить уникальное число в массив

            unique\_numbers[num\_unique] = number;

            num\_unique++;

        }

    }

    fclose(fileA); //закрыть файл f

    FILE\* fileB; //создание указателя на файл g

    errno\_t errB; //переменная для ошибки

    errB = fopen\_s(&fileB, "fileB.txt", "w"); //открытие файла g для чтения

**if** (errB != 0) { //если ошилка

        perror("Ошибка открытия файла!"); //вывести сообзение

        fclose(fileA); //закрыть файл f

**return** 2;

    }

**for** (**int** i = 0; i < num\_unique; i++) { //запись уникальных элементов файла f в файл g

        fprintf(fileB, "%d\t", unique\_numbers[i]);

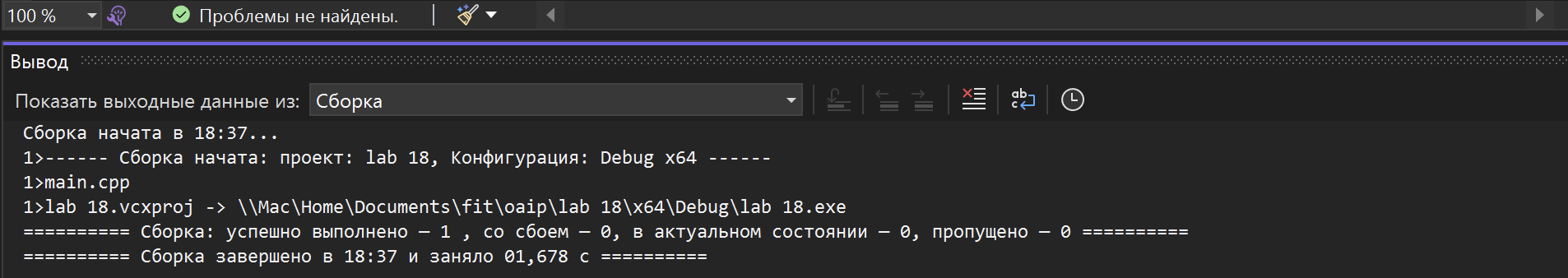
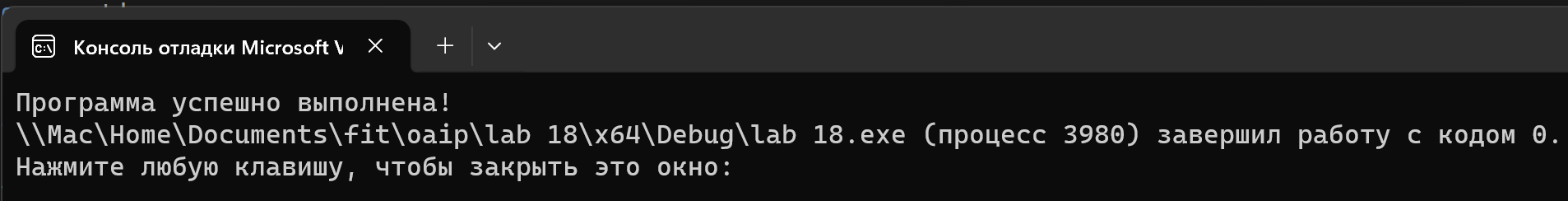
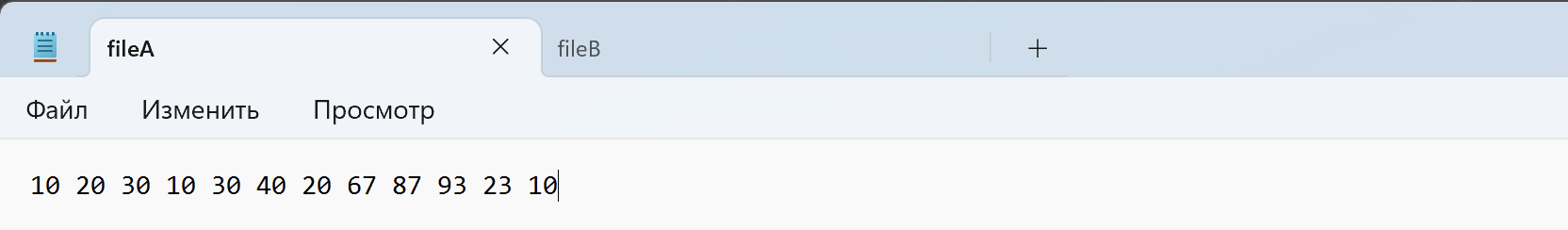
    }

    fclose(fileB); //закрыть файл g

    printf("Программа успешно выполнена!");

**return** 0;

}



2. Создать текстовый файл **F1** не менее, чем из 6 строк, и записать в него информацию. Скопировать в файл **F2** только те строки из **F1**, которые заканчиваются символом «**а**».

#include <iostream>

#include <Windows.h>

#include <stdio.h>

**int** main() {

    setlocale(LC\_ALL, "Russian"); //локализацяи консоли

    FILE\* F1; //создание указателя на файл 1

    errno\_t err1; //переменная для ошибки

    err1 = fopen\_s(&F1, "F1.txt", "r"); //открытие файла 1 для чтения

**if** (err1 != 0) { //если ошибка

        perror("Ошибка открытия файла 1!");

**return** 1;

    }

    FILE\* F2; //сощдание указателя на файл 2

    errno\_t err2; //переменная для ошибки

    err2 = fopen\_s(&F2, "F2.txt", "w");//открытие файла 2 для чтения

**if** (err2 != 0) { //если ошибка

        perror("Ошибка открытия файла 2!");

        fclose(F1); //закрыть файл 1

**return** 2;

    }

**char** line[100]; //создание массива для добавления строки

**while** (fgets(line, **sizeof**(line), F1)) { //цикл для проверки каждой строки

**if** (line[strlen(line) - 2] == 'a') { //если строка заканчивается на а

            fputs(line, F2); //добавить строку в файл 2

        }

    }

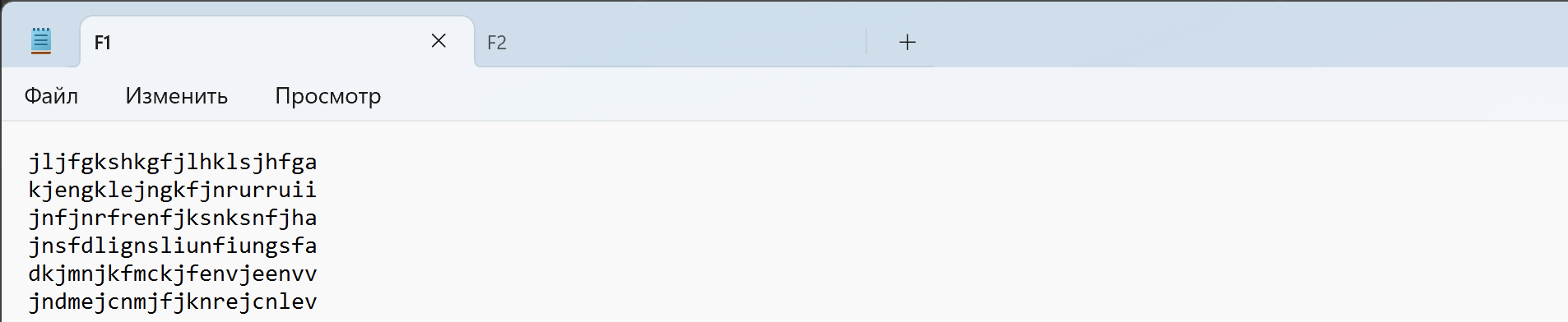
    fclose(F1); //закрыть файл 1

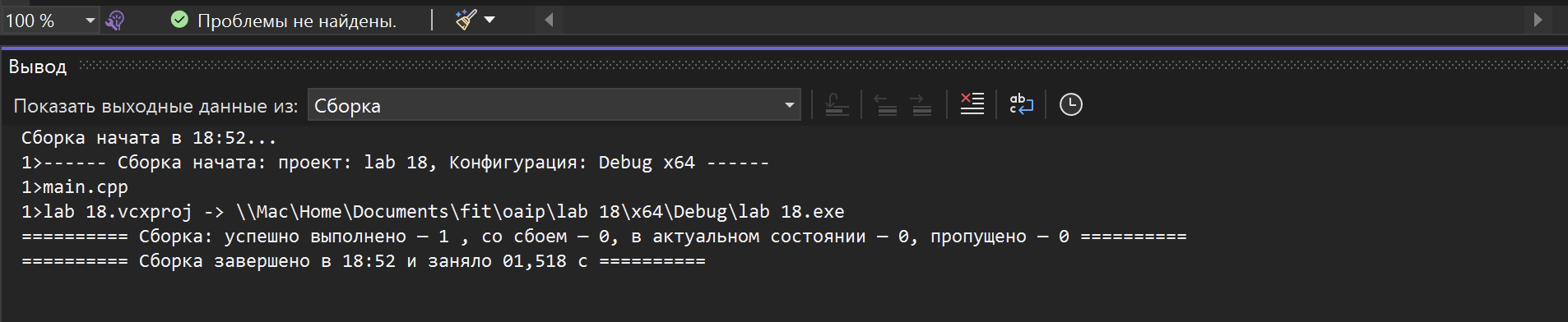
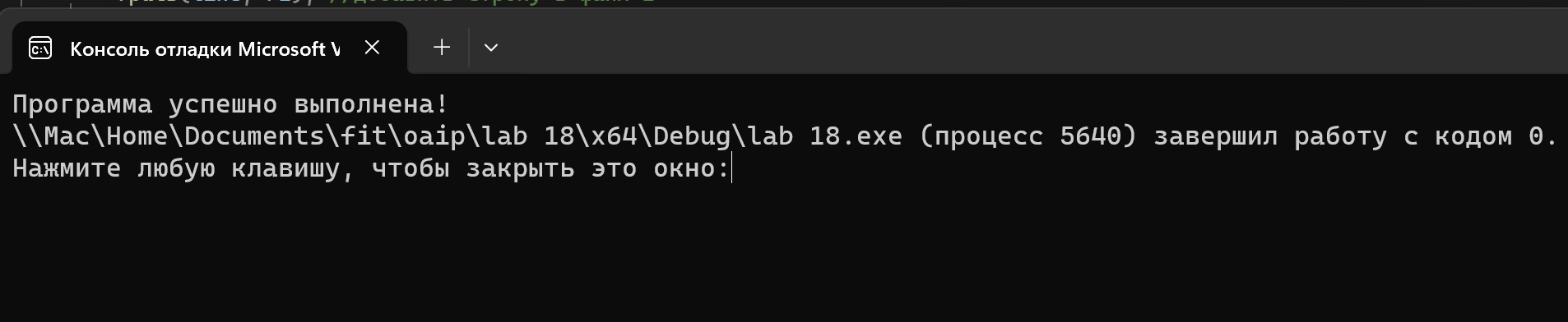
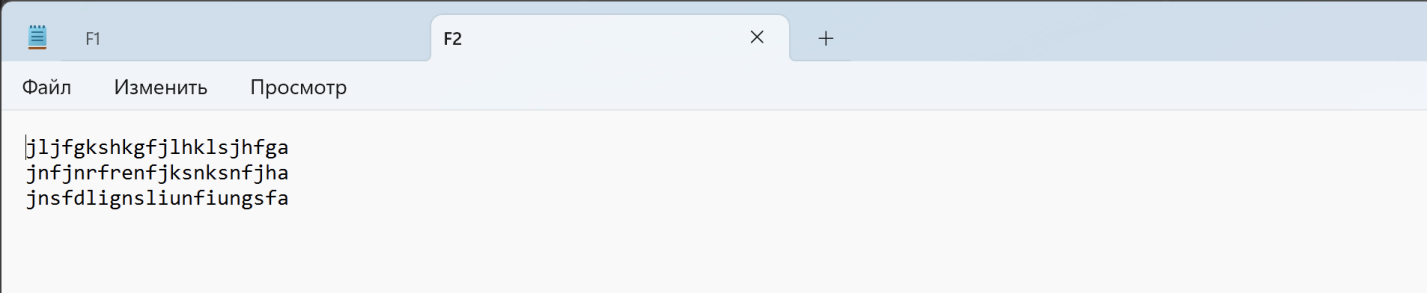
    fclose(F2); //закрыть файл 2

    printf("Программа успешно выполнена!");

**return** 0;

}

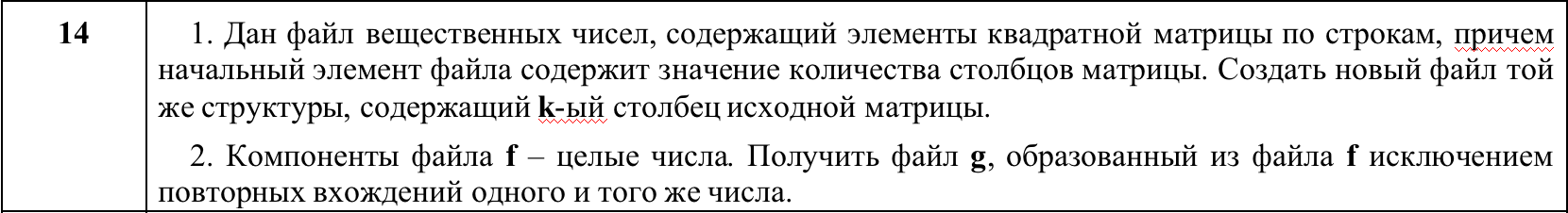




**Дополнительные задания**

Вариант 14

В соответствии со своим вариантом разработать программы для условий, приведенных в таблице ниже, и изучить способы работы с файлами на ***языке С***.



1. Дан файл вещественных чисел, содержащий элементы квадратной матрицы по строкам, причем начальный элемент файла содержит значение количества столбцов матрицы. Создать новый файл той же структуры, содержащий **k**-ый столбец исходной матрицы.

#include <iostream>

#include <Windows.h>

#include <stdio.h>

**using** **namespace** std;

**int** main() {

    setlocale(LC\_ALL, "Russian"); //локализация консоли

    FILE\* fileA; //создание указател яна файл А

    errno\_t errA; //переменная для ощибки

    errA = fopen\_s(&fileA, "FileA.txt", "r"); //открытие файла А для чтения

**if** (errA != 0) { //если ошибка

        perror("Ошибка открытия файла!");

**return** 1;

    }

**int** amm\_of\_columns; //создание переменной для подсчёта кол-ва столбцов из первого элемента

    fscanf\_s(fileA, "%d", &amm\_of\_columns); //считывание первого элемента из файла

**double** matrix[100][100]; //создание матрицы

**for** (**int** i = 0; i < amm\_of\_columns; i++) { //цикл в цикле для добваления элементов файла в матрицу

**for** (**int** j = 0; j < amm\_of\_columns; j++) {

            fscanf\_s(fileA, "%lf", &matrix[i][j]); //считывание элементов из файла

        }

    }

    fclose(fileA); //закрытие файла А

**int** k; //вовод номера искомого столбца с калвиатуры

    cout << "Введите искомый номер столбца: ";

    cin >> k;

    FILE\* fileB; //создание указателя на файл В

    errno\_t errB; //переменная для ошибки

    errB = fopen\_s(&fileB, "FileB.txt", "w"); //открытие файла В для записи

**if** (errB != 0) { //если ошибка

        perror("Ошидка открытия файла!"); //вывести сообщение

        fclose(fileA); //закрыть файл А

**return** 2;

    }

**for** (**int** i = 0; i < amm\_of\_columns; i++) { //цикл дял ввода нужных элементов матрицы в файл В

        fprintf(fileB, "%lf\n", matrix[i][k - 1]);

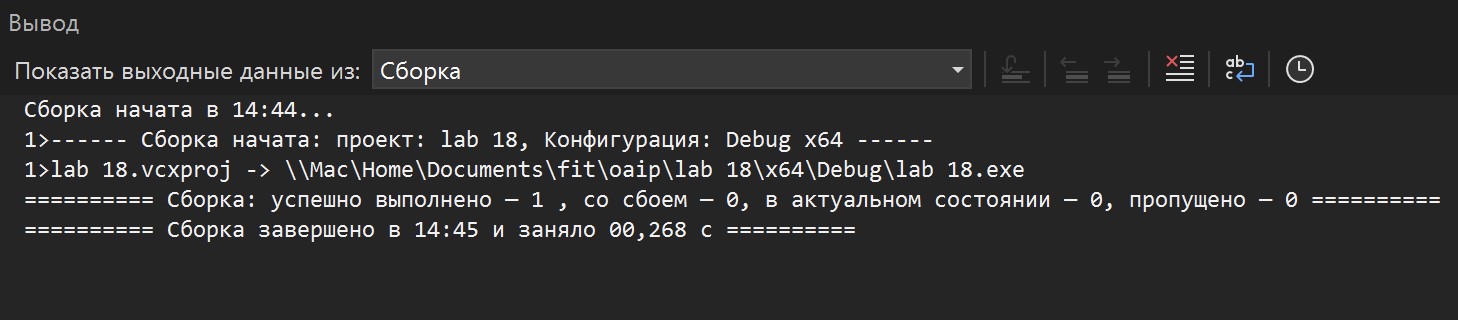
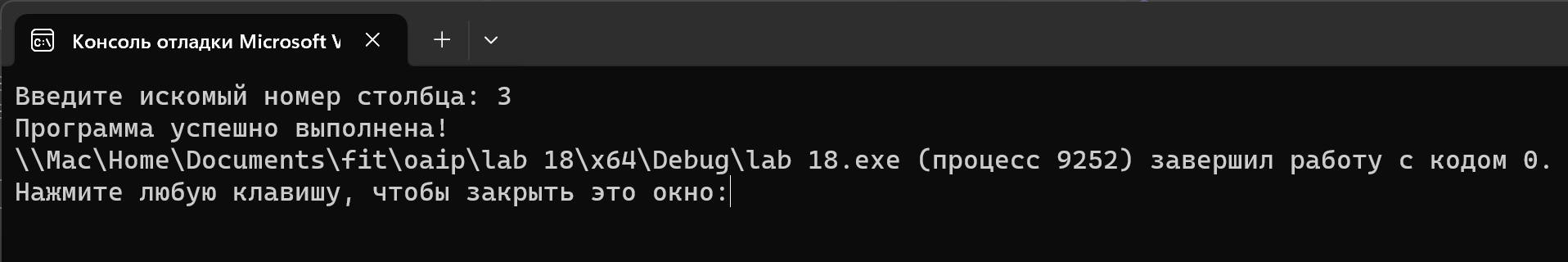
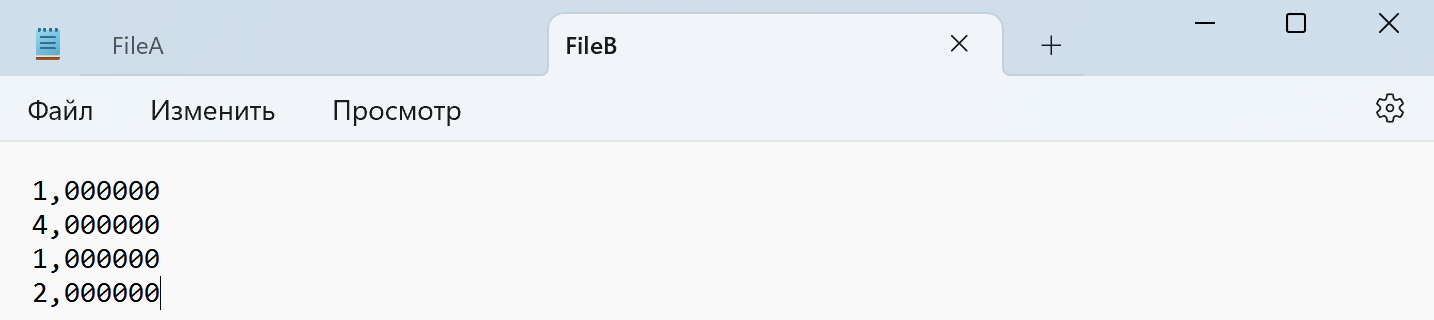
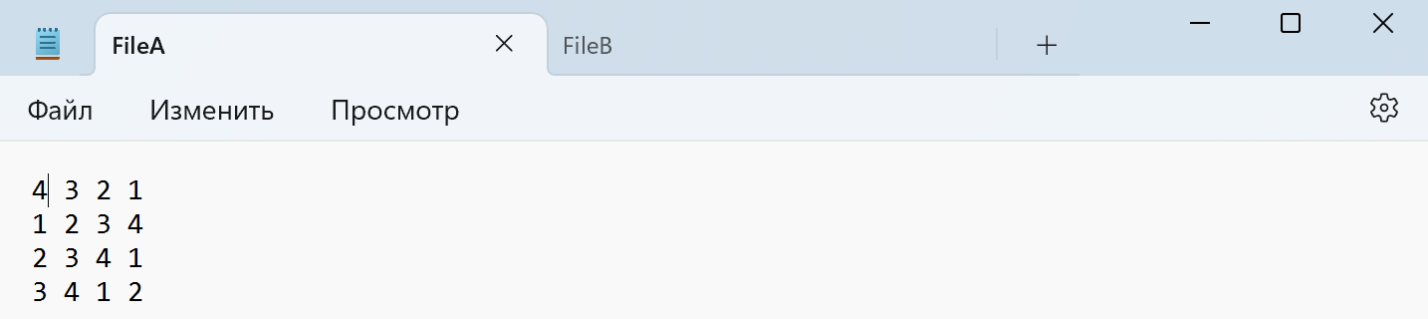
    }

    fclose(fileB); //закрытие файла В

    printf("Программа успешно выполнена!");

**return** 0;

}



1. Компоненты файла **f** – целые числа. Получить файл **g**, образованный из файла **f** исключением повторных вхождений одного и того же числа.

#include <iostream>

#include <Windows.h>

#include <stdio.h>

**int** main() {

    setlocale(LC\_ALL, "Russian"); //локализация консоли

    FILE\* f; //создание указателя на файл f

    errno\_t errF; //переменная для ошибки

    errF = fopen\_s(&f, "f.txt", "r"); //открытие файла f для чтения

**if** (errF != 0)  {//если ошибка

        perror("Ошибка открытия файла!");

**return** 1;

    }

**int** unique\_numbers[100];

**int** num\_unique = 0;

**int** number;

**while** (fscanf\_s(f, "%d", &number) == 1) { //цикл для чтения каждого элемента файла

**int** isDuplicate = 0; //созжание контрольной переменной

**for** (**int** i = 0; i < num\_unique; i++) { //цикл для проверки вхождения уникальных чисел

**if** (unique\_numbers[i] == number) {

                isDuplicate = 1; //если найдено повторное фхождение

**break**;

            }

        }

**if** (!isDuplicate) { //добавить уникальное число в массив

            unique\_numbers[num\_unique] = number;

            num\_unique++;

        }

    }

    fclose(f); //закрыть файл f

    FILE\* g; //создание указателя на файл g

    errno\_t errG; //переменная для ошибки

    errG = fopen\_s(&g, "g.txt", "w"); //открытие файла g для чтения

**if** (errG != 0) { //если ошилка

        perror("Ошибка открытия файла!");

        fclose(f); //закрыть файл f

**return** 2;

    }

**for** (**int** i = 0; i < num\_unique; i++) {  //запись уникальных элементов файла f в файл g

        fprintf(g, "%d\t", unique\_numbers[i]);

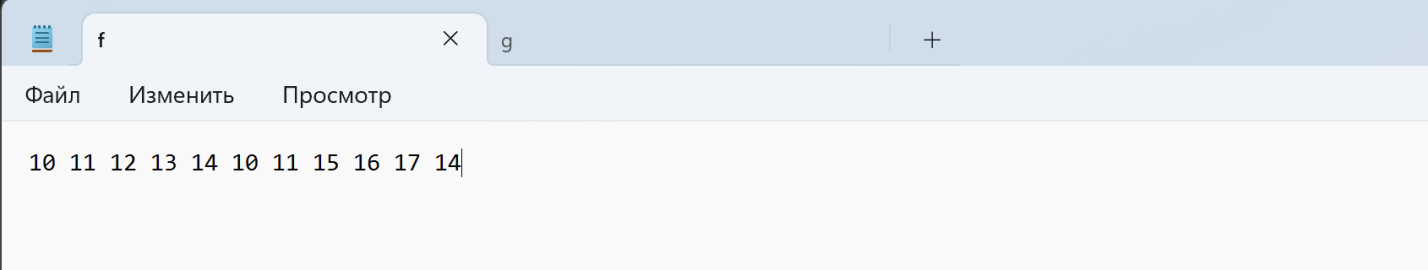
    }

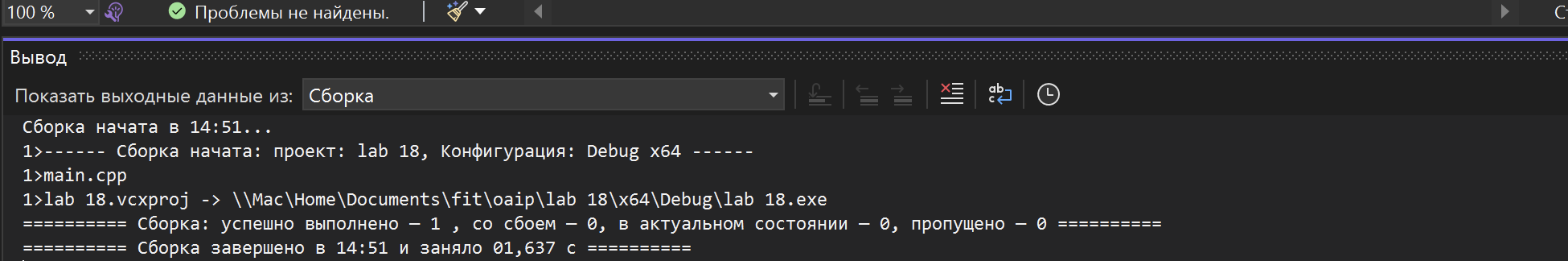
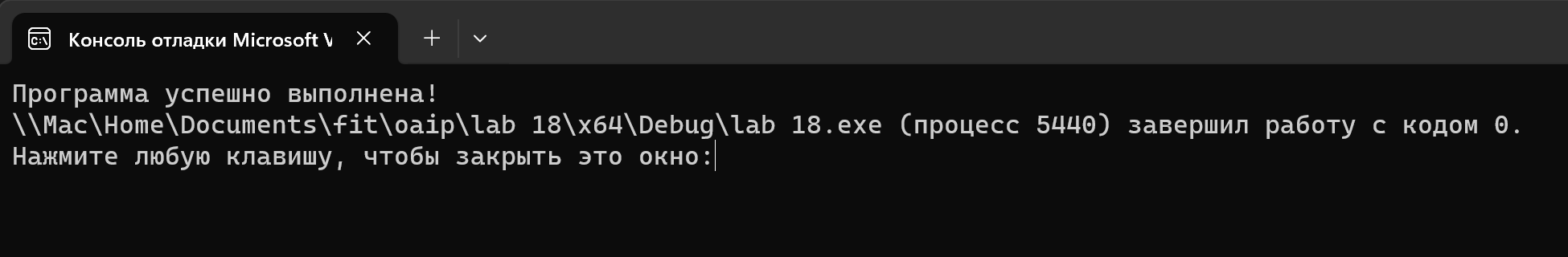
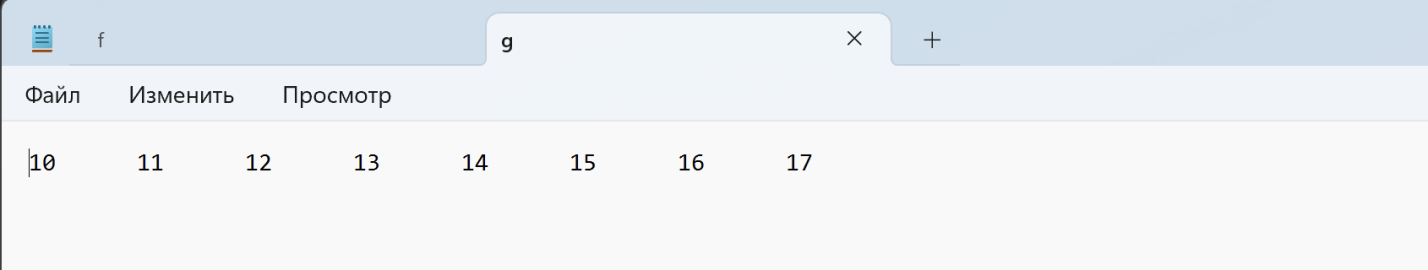
    fclose(g); //закрыть файл g

    printf("Программа успешно выполнена!");

**return** 0;

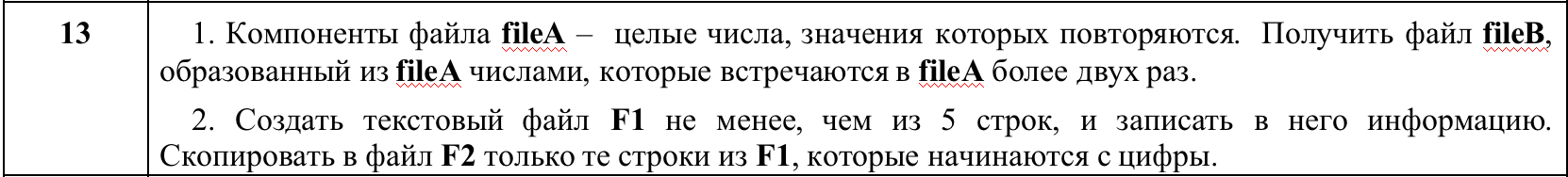
}





Вариант 13

В соответствии со своим вариантом разработать программы для условий, приведенных в таблице ниже, и изучить способы работы с файлами на ***языке С***.



1. Компоненты файла **fileA** –  целые числа, значения которых повторяются.  Получить файл **fileB**, образованный из **fileA** числами, которые встречаются в **fileA** более двух раз.

#include <iostream>

#include <Windows.h>

#include <stdio.h>

**int** main() {

    setlocale(LC\_ALL, "Russian"); // Устанавливаем локализацию консоли на русский язык

    FILE\* fileA; // Указатель на файл А

    errno\_t errA; // Переменная для ошибки при открытии файла А

    errA = fopen\_s(&fileA, "fileA.txt", "r"); // Открываем файл А для чтения

**if** (fileA == **NULL**) { // Проверка на ошибку при открытии файла А

        perror("Ошибка открытия файла fileA");

**return** 1;

    }

**int** numbers[100]; // Массив для хранения чисел из файла А

**int** count = 0; // Счетчик чисел в массиве

**int** number;

**while** (fscanf\_s(fileA, "%d", &number) == 1) { // Считываем числа из файла А и записываем в массив

        numbers[count] = number;

        count++;

    }

    fclose(fileA); // Закрываем файл А

    FILE\* fileB; // Указатель на файл B

    errno\_t errB; // Переменная для ошибки при открытии файла B

    errB = fopen\_s(&fileB, "fileB.txt", "w"); // Открываем файл B для записи

**if** (fileB == **NULL**) { // Проверка на ошибку при открытии файла B

        perror("Ошибка открытия файла fileB");

        fclose(fileA); // Закрываем файл А в случае ошибки

**return** 2;

    }

**for** (**int** i = 0; i < count; i++) { // Цикл, проходящий по всем элементам массива

**int** freq = 0; // Переменная для подсчета количества вхождений конкретного элемента

**int** curnum = numbers[i]; // Текущее число для подсчета

**for** (**int** j = 0; j < count; j++) { // Цикл для подсчета количества вхождений числа

**if** (numbers[j] == curnum) { // Если обнаружено вхождение

                freq++; // Увеличиваем счетчик

            }

        }

**if** (freq > 2) { // Если число встречается более двух раз

            fprintf(fileB, "%d\t", curnum); // Записываем число в файл B

        }

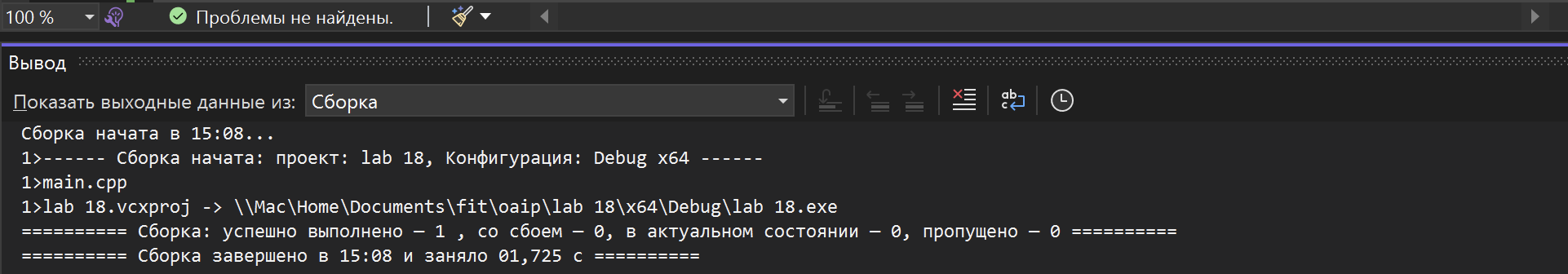
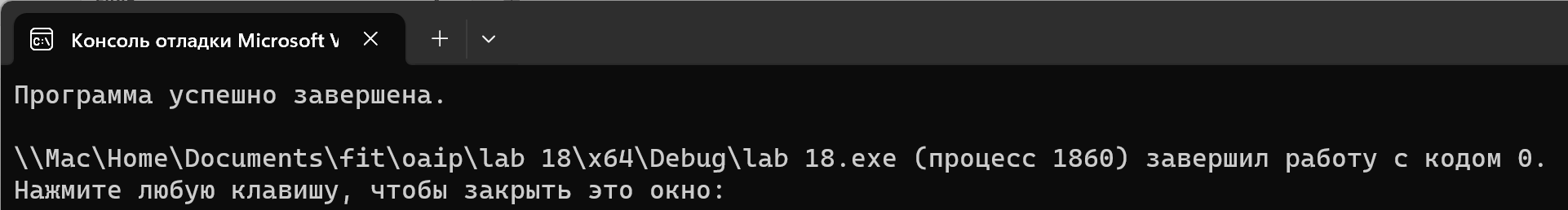
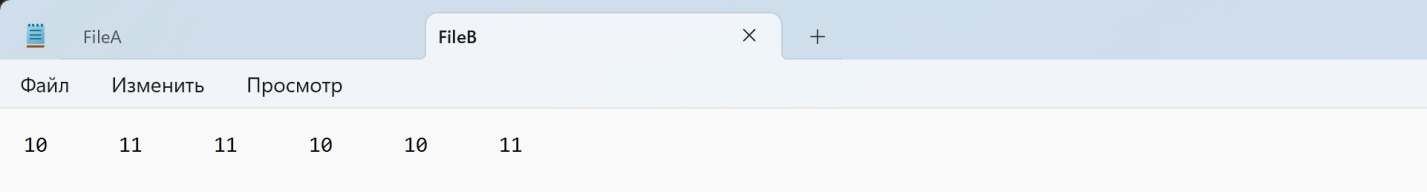
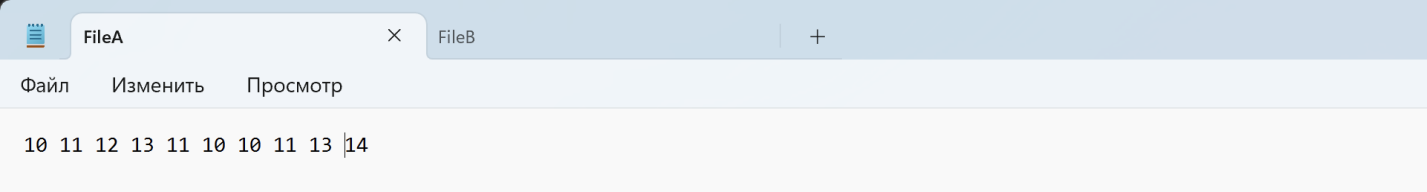
    }

    fclose(fileB); // Закрываем файл B

    printf("Программа успешно завершена.\n"); // Выводим сообщение о завершении программы

**return** 0;

}



2. Создать текстовый файл **F1** не менее, чем из 5 строк, и записать в него информацию. Скопировать в файл **F2** только те строки из **F1**, которые начинаются с цифры.

#include <iostream>

#include <Windows.h>

#include <stdio.h>

**int** main() {

    setlocale(LC\_ALL, "Russian"); //локализация консоли

    FILE\* F1; //создание указателя на файл 1

    errno\_t err1; //переменная для ошибки

    err1 = fopen\_s(&F1, "F1.txt", "r"); //открытие файла 1 для чтения

**if** (err1 != 0) { //есди есть ошибка

        perror("Ошибка открытия файла!"); //вывести сообщение об ошибке

**return** 1;

    }

    FILE\* F2; //содание указателя на файл 2

    errno\_t err2; //переменная для ошибки

    err2 = fopen\_s(&F2, "F2.txt", "w"); //открытие файла 2 для записи

**if** (err2 != 0) { //есди есть ошика

        perror("Ошибка открытия файла!");

        fclose(F1); //закрыть файл 1

**return** 2;

    }

**char** line[100]; //создание массива для хранения строки

**while** (fgets(line, **sizeof**(line), F1) != **NULL**) { //передача строки из файла в массив

**if** (line[0] > '0' && line[0] < '9') { //если первый знак - цифра

            fprintf(F2, "%s", line); //добавить строку в файл 2

        }

    }

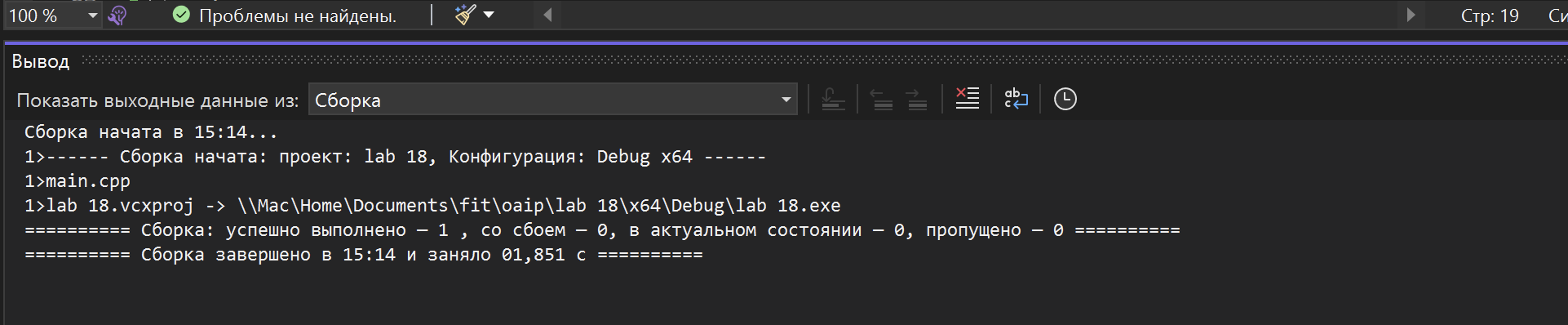
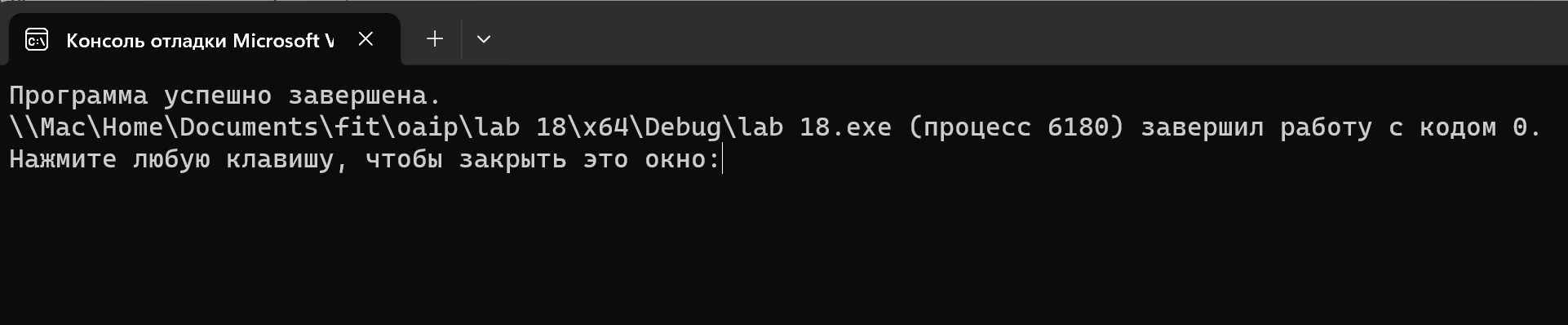
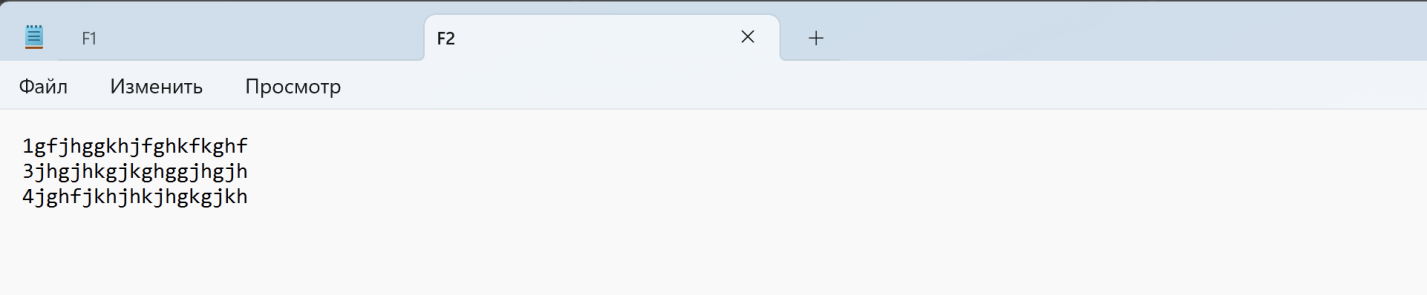
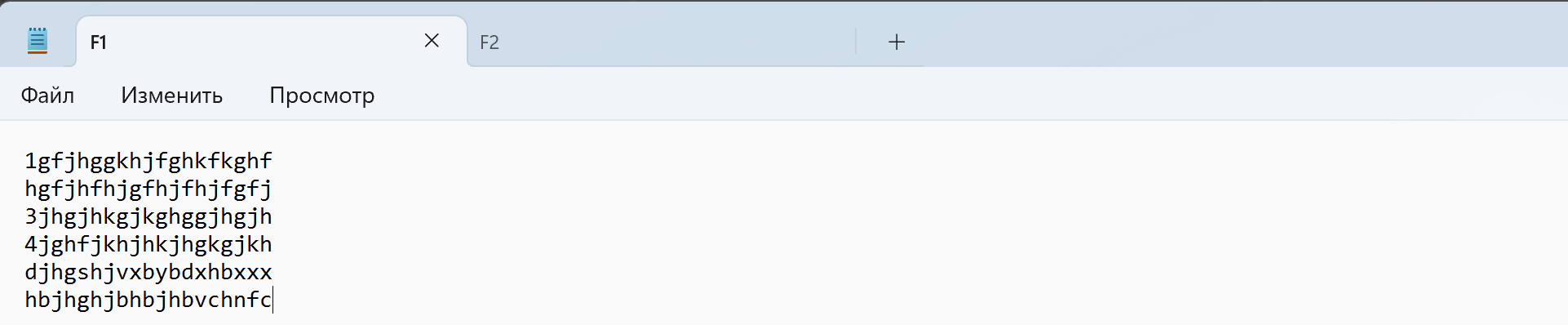
    fclose(F1); //закрыть файл 2

    fclose(F2); //закрыть файл 1

    printf("Программа успешно завершена.");

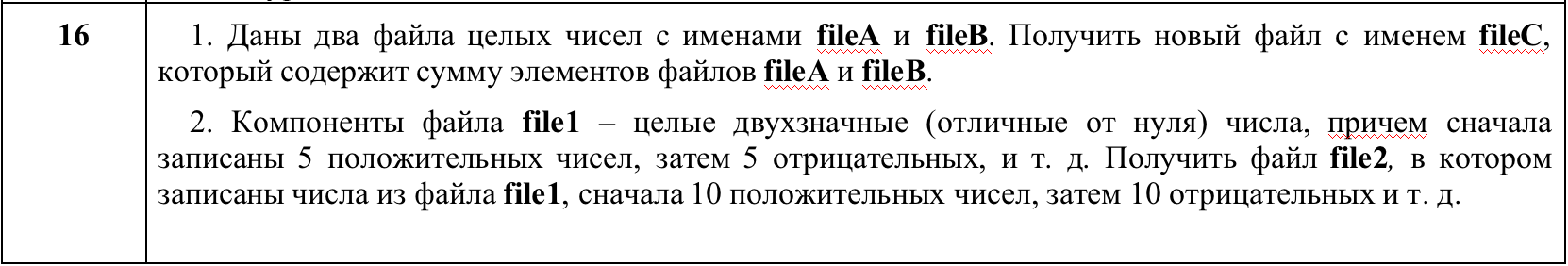
**return** 0;

}



Вариант 16

В соответствии со своим вариантом разработать программы для условий, приведенных в таблице ниже, и изучить способы работы с файлами на ***языке С***.



1. Даны два файла целых чисел с именами **fileA** и **fileB**. Получить новый файл с именем **fileC**, который содержит сумму элементов файлов **fileA** и **fileB**.

#define \_CRT\_SECURE\_NO\_DEPRECATE //для отключения предупреждений компилятора, связанных с использованием устаревших или небезопасных функций

#include <iostream>

#include <stdio.h>

**int** main() {

    setlocale(LC\_ALL, "RU"); // устанавливаем русскую локаль для корректного отображения кириллицы

    FILE\* fileA;

    errno\_t errA = fopen\_s(&fileA, "fileA1.txt", "r"); // открываем файл fileA1.txt для чтения

**if** (errA != 0) {

        perror("Не удалось открыть файл fileA1.txt"); // выводим сообщение об ошибке

**return** 1;

    }

**else** {

        printf("Файл fileA1.txt успешно открыт\n"); // выводим сообщение об успешном открытии файла

    }

    FILE\* fileB;

    errno\_t errB = fopen\_s(&fileB, "fileB1.txt", "r"); // открываем файл fileB1.txt для чтения

**if** (errB != 0) {

        perror("Не удалось открыть файл fileB1.txt"); // выводим сообщение об ошибке

        fclose(fileA); // закрываем файл fileA1.txt

**return** 2;

    }

**else** {

        printf("Файл fileB1.txt успешно открыт\n"); // выводим сообщение об успешном открытии файла

    }

    FILE\* fileC;

    errno\_t errC = fopen\_s(&fileC, "fileC1.txt", "w"); // открываем файл fileC1.txt для записи

**if** (errC != 0) {

        perror("Не удалось открыть файл fileC1.txt"); // выводим сообщение об ошибке

        fclose(fileA); // закрываем файл fileA1.txt

        fclose(fileB); // закрываем файл fileB1.txt

**return** 3;

    }

**else** {

        printf("Файл fileC1.txt успешно открыт\n"); // выводим сообщение об успешном открытии файла

    }

**int** numA, numB;

**while** (fscanf(fileA, "%d", &numA) != EOF && fscanf(fileB, "%d", &numB) != EOF) {

        fprintf(fileC, "%d ", numA + numB); // суммируем числа из fileA1.txt и fileB1.txt, записываем результат в fileC1.txt

    }

    fclose(fileA); // закрываем файл fileA1.txt

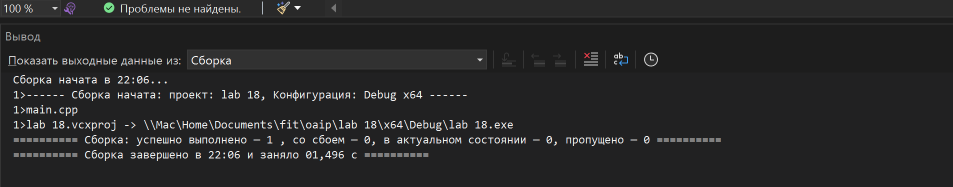
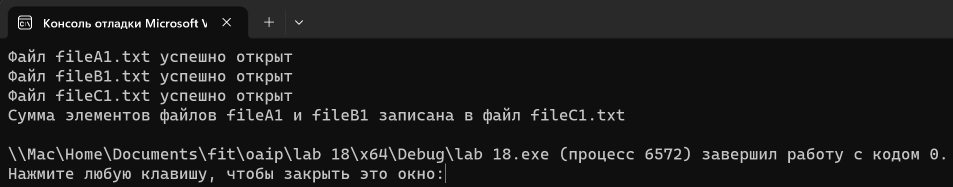
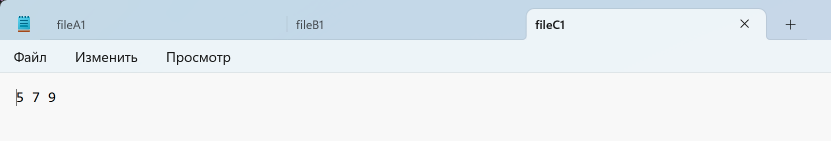
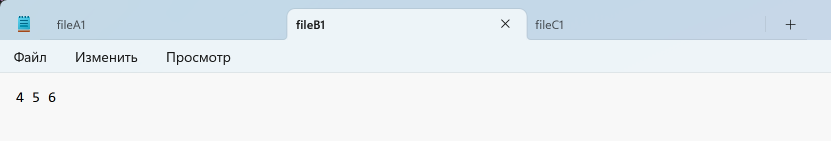
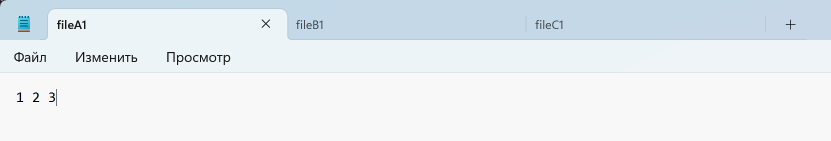
    fclose(fileB); // закрываем файл fileB1.txt

    fclose(fileC); // закрываем файл fileC1.txt

    printf("Сумма элементов файлов fileA1 и fileB1 записана в файл fileC1.txt\n"); // выводим сообщение об успешном выполнении программы

**return** 0;

}



2. Компоненты файла**file1** – целые двухзначные (отличные от нуля) числа, причем сначала записаны 5 положительных чисел, затем 5 отрицательных, и т. д. Получить файл **file2***,* в котором записаны числа из файла **file1**,сначала 10 положительных чисел, затем 10 отрицательных и т. д.

#define \_CRT\_SECURE\_NO\_DEPRECATE //для отключения предупреждений компилятора, связанных с использованием устаревших или небезопасных функций

#include <iostream>

#include <Windows.h>

#include <stdio.h>

**int** main() {

    setlocale(LC\_ALL, "Russian"); // Установка русской локали для корректного отображения кириллицы

    FILE\* file1; // Указатель на файл 1

    errno\_t err1; // Переменная для ошибки

    err1 = fopen\_s(&file1, "file1.txt", "r"); // Открытие файла 1 для чтения

**if** (err1 != 0) { // Если есть ошибка

        perror("Ошибка открытия файла!"); // Вывести сообщение об ошибке

**return** 1;

    }

    FILE\* file2; // Указатель на файл 2

    errno\_t err2; // Переменная для ошибки

    err2 = fopen\_s(&file2, "file2.txt", "w"); // Открытие файла 2 для записи

**if** (err2 != 0) { // Если есть ошибка

        perror("Ошибка открытия файла!");

        fclose(file1); // Закрыть файл 1

**return** 2;

    }

**int** num;

**int** positive[10];

**int** negative[10];

**int** count\_positive = 0;

**int** count\_negative = 0;

    // Чтение чисел из файла file1

**while** (fscanf(file1, "%d", &num) != EOF) {

**if** (num >= 0 && count\_positive < 10) {

            positive[count\_positive++] = num; // Сохранение положительных чисел

        }

**else** **if** (num < 0 && count\_negative < 10) {

            negative[count\_negative++] = num; // Сохранение отрицательных чисел

        }

    }

    // Запись положительных чисел в файл file2

**for** (**int** i = 0; i < count\_positive; ++i) {

        fprintf(file2, "%d ", positive[i]);

    }

    // Запись отрицательных чисел в файл file2

**for** (**int** i = 0; i < count\_negative; ++i) {

        fprintf(file2, "%d ", negative[i]);

    }

    fclose(file1); // Закрыть файл 1

    fclose(file2); // Закрыть файл 2

    printf("Числа из файла file1.txt переписаны в файл file2.txt\n");

**return** 0;

}

