Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 18

По дисциплине “Основы алгоритмизации и программирования”

На тему “**Лабораторная работа № 18. Работа с файлами на языке С**”

Выполнил:

Студент 1 курса 6 группы

Кравченко Сергей Сергеевич

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 8

|  |  |
| --- | --- |
| **8** | 1. Дан файл целых чисел с элементами **A(i), i = 0, ..., N – 1** (**N** − размер файла). Заменить исходное расположение его элементов на следующее: **A(0), A(N – 1), A(1), A(N – 2), A(2), ... .**  2. Даны два файла целых чисел, содержащие элементы матрицы **A** размерности **n** x **n** и **B** размерности **n** x **1** по строкам. Начальный элемент каждого файла содержит количество столбцов соответствующей матрицы. Создать файл **C**, содержащий произведение матриц **А** и **В**. |

1.

#include <stdio.h>

#include <iostream>

#include <locale>

#include <cstdlib>

#include <ctime>

using namespace std;

int main()

{

// Инициализация генератора случайных чисел и локали

srand(time(0));

setlocale(LC\_ALL, "");

int a;

int array[8];

errno\_t errf1, errf2;

FILE\* f = nullptr, \* f2 = nullptr;

// Открытие файла "file1.txt" для записи

errf1 = fopen\_s(&f, "file1.txt", "w");

// Проверка на успешное открытие файла

if (errf1 != 0)

{

perror("Невозможно создать файл\n");

return EXIT\_FAILURE;

}

// Запись случайных чисел в файл "file1.txt"

for (a = 0; a < 8; a++)

{

fprintf(f, "%d, ", rand() % 10);

}

printf("Данные записаны в file1.txt\n");

// Закрытие файла "file1.txt"

fclose(f);

f = nullptr;

// Повторное открытие файла "file1.txt" для чтения

errf1 = fopen\_s(&f, "file1.txt", "r");

// Проверка на успешное открытие файла

if (errf1 != 0)

{

perror("Невозможно создать файл\n");

return EXIT\_FAILURE;

}

// Открытие файла "file2.txt" для записи

errf2 = fopen\_s(&f2, "file2.txt ", "w");

// Проверка на успешное открытие файла

if (errf2 != 0)

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

// Чтение данных из файла "file1.txt"

for (a = 0; a < 8; a++)

{

fscanf\_s(f, "%d, ", &array[a]);

}

// Вывод содержимого файла "file1.txt" на экран

cout << "Содержимое file1.txt" << endl;

for (int i = 0; i < 8; i++) {

cout << array[i] << ", ";

}

int arr\_size = 8;

// Запись данных из файла "file1.txt" в файл "file2.txt" в обратном порядке

cout << endl << endl << "Содержимое file2.txt" << endl;

for (a = 0; a < arr\_size / 2; a++)

{

fprintf(f2, "%d, ", array[a]);

cout << array[a] << ", ";

fprintf(f2, "%d, ", array[arr\_size - a - 1]);

cout << array[arr\_size - a - 1] << ", ";

}

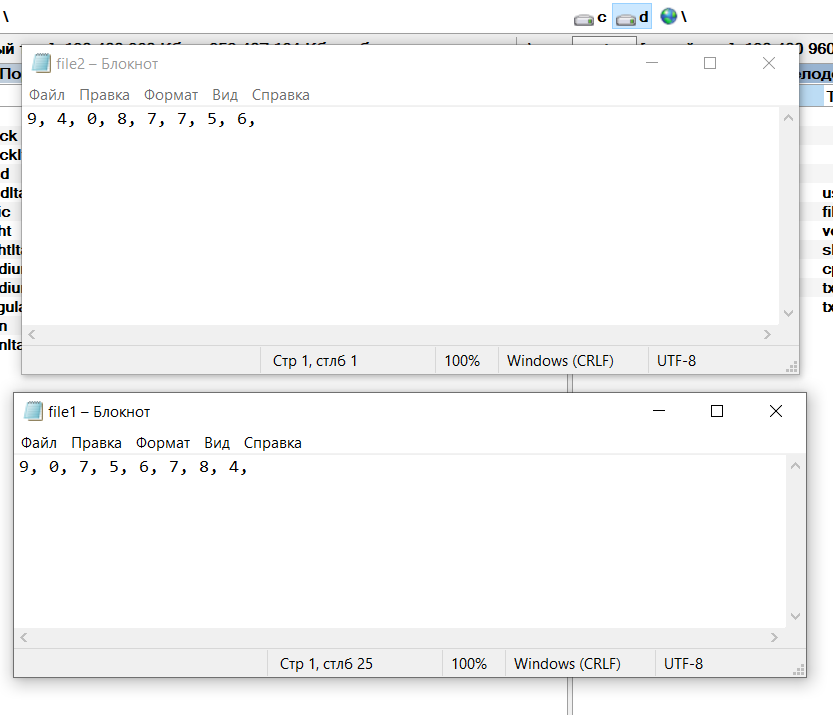
// Закрытие обоих файлов

fclose(f);

fclose(f2);

return 0;

}



2.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

using namespace std;

int main() {

const int rows = 2; const int cols = 2;

int matrixA[rows][cols]; // Матрица A

int matrixB[rows][cols]; // Матрица B

int matrixC[rows][cols]; // Матрица C для хранения результата

int i, j;

FILE\* out = fopen("file3.txt", "w");

FILE\* fileA = fopen("file1.txt", "r");

FILE\* fileB = fopen("file2.txt", "r");

int ka = fgetc(fileA) - '0'; // Считывает количество столбцов матрицы A

int kb = fgetc(fileB) - '0'; // Считывает количество столбцов матрицы B

// Считывает элементы матрицы A

for (i = 0; i < rows; i++) {

for (j = 0; j < cols; j++) {

fscanf(fileA, "%d", &matrixA[i][j]);

}

}

// Считывает элементы матрицы B

for (i = 0; i < rows; i++) {

for (j = 0; j < cols; j++) {

fscanf(fileB, "%d", &matrixB[i][j]);

}

}

// Выполняет умножение матриц и записывает результат в файл

for (i = 0; i < rows; i++) {

for (j = 0; j < 1; j++) {

matrixC[i][j] = 0;

for (int k = 0; k < cols; k++) {

matrixC[i][j] += matrixA[i][k] \* matrixB[k][j]; // Выполняет умножение и суммирование

}

fprintf(out, "%d ", matrixC[i][j]); // Записывает результат в файл

}

fprintf(out, "\n"); // Переходит на новую строку

}

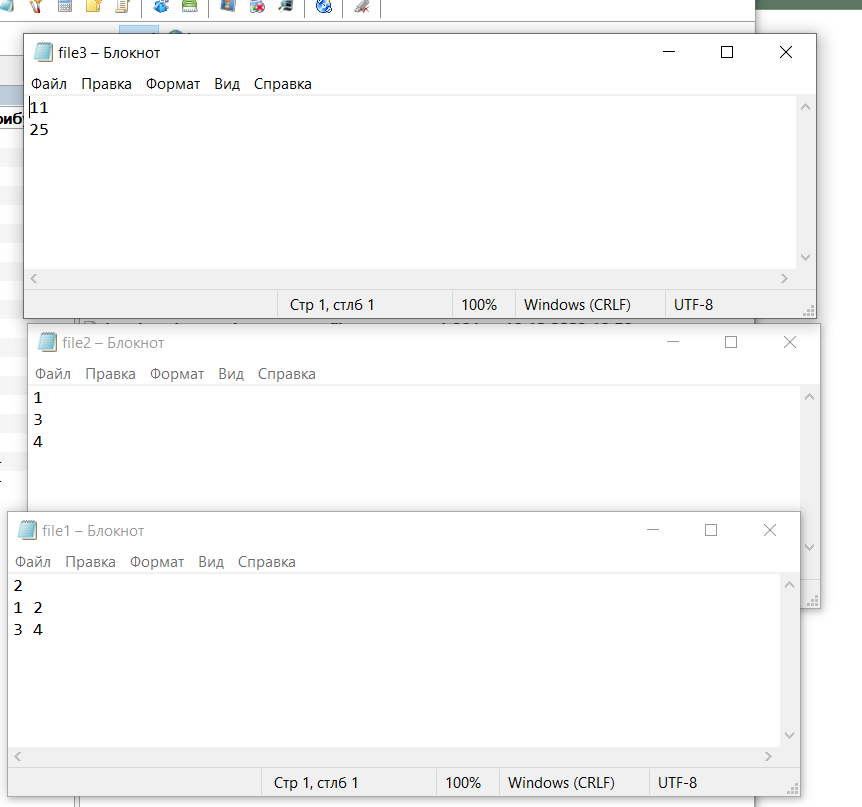
fclose(fileA);

fclose(fileB);

fclose(out);

return 0;

}



Дополнительные варианты

|  |  |
| --- | --- |
| **9** | 1. Компоненты файла **fileA** –  целые числа, значения которых повторяются.  Получить файл **fileB**, образованный из **fileA** исключением повторных вхождений одного и того же числа.  2. Создать текстовый файл **F1** не менее, чем из 4 строк, и записать в него информацию. Скопировать из файла **F1** в файл **F2** строки, количество символов в которых больше, чем заданное число, которое вводится с клавиатуры. |

1.

#include <stdio.h>

#include<iostream>

#include <locale>

using namespace std;

int main()

{

srand(time(0));

setlocale(LC\_ALL, "");

int a;

int array[15];

errno\_t errf1, errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.

FILE\* f, \* f2;//создаём указатели на файлы

errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи

if (errf1 != 0)//предусматриваем возможность ошибки

{

perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.

return EXIT\_FAILURE;

}

for (a = 0; a < 15; a++)

{

fprintf(f, "%d, ", rand() % 10);//генерируем числа в файл

}

printf("Данные записаны в file1.txt\n");

fclose(f);//закрываем файл

errf1 = fopen\_s(&f, "file1.txt", "r");//открываем тот же файл в режиме чтения

if (errf1 != 0)

{

perror("Невозможно создать файл\n");

return EXIT\_FAILURE;

}

errf2 = fopen\_s(&f2, "file2.txt ", "w");//открываем второй файл в режиме записи

if (errf2 != 0)

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

for (a = 0; a < 15; a++)

{

fscanf\_s(f, "%d, ", &array[a]);//считываем числа из первого файла в массив

}

cout << "Содержимое file1.txt" << endl;

for (int i = 0; i < 15; i++) {

cout << array[i] << ", ";

}

int arr\_size = 15;

for (int i = 0; i < arr\_size; i++) {//удаляем повторяющиеся элементы из массива

for (int j = i + 1; j < arr\_size; j++) {

if (array[i] == array[j]) {

for (int k = j; k < arr\_size; k++) {

array[k] = array[k + 1];

}

j--;

arr\_size--;

}

}

}

cout << endl << "Содержимое file2.txt" << endl;

for (int i = 0; i < arr\_size; i++) {

cout << array[i] << ", ";

}

for (a = 0; a < arr\_size; a++)//записываем значения массива во второй файл

{

fprintf(f2, "%d, ", array[a]);

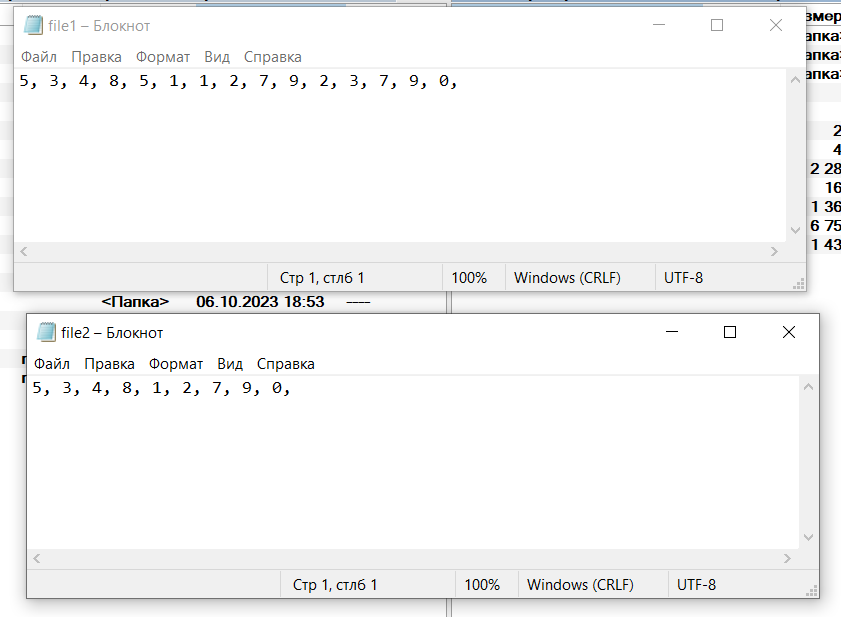
}

fclose(f);

fclose(f2);//закрываем файлы, с которыми работали

return 0;

}



2.

#include <stdio.h>

#include<iostream>

#include <locale>

using namespace std;

int main()

{

srand(time(0));

setlocale(LC\_ALL, "");

int a, lenght;

float array[15];

char buffer[250];

char\* estr;

errno\_t errf1, errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.

FILE\* f, \* f2;//создаём указатели на файлы

errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи

if (errf1 != 0)//предусматриваем возможность ошибки

{

perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.

return EXIT\_FAILURE;

}

printf("Введите текст:\n");

for (a = 0; a < 4; a++)

{

gets\_s(buffer);//передаем введённую строку в буфер

fputs(buffer, f); fprintf(f, "\n");//печатаем буфер в файл

}

printf("Данные записаны в file1.txt\n");

printf("Введите нобходимую длину для переноса:\n");

cin >> lenght;

fclose(f);//закрываем файл

errf2 = fopen\_s(&f2, "file2.txt", "w"); //Открытие для чтения

if (errf2 != NULL)

//Проверка открытия файла

{

printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;

}

errf1 = fopen\_s(&f, "file1.txt", "r"); //Открытие для чтения

if (errf1 != NULL)

//Проверка открытия файла

{

printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;

}

else

printf("\nЗапсианы в file2.txt строки:\n");

for (int i = 0; i < 4; i++) {

fgets(buffer, sizeof(buffer), f); //Чтение строки файла

if (lenght < strlen(buffer) - 1) {//если строка проходит критерий длины, то записываем ее во второй файл

fputs(buffer, f2);

puts(buffer);

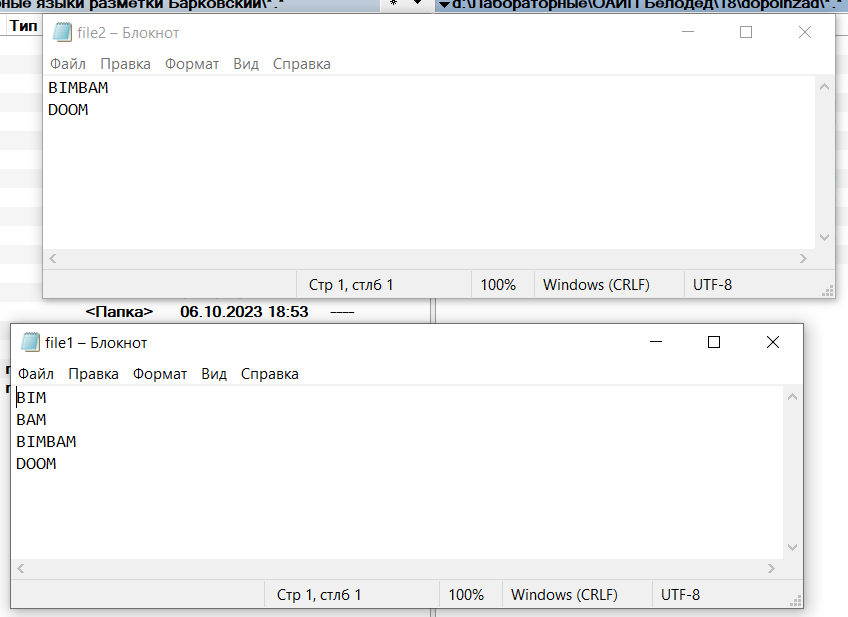
}

}

fclose(f);

fclose(f2);//закрываем файлы, с которыми работали

}



|  |  |
| --- | --- |
| **6** | 1. Компоненты файла **fA** – вещественные числа (положительные и отрицательные). Определить и вывести на экран порядковый номер того из них, которое наиболее близко к введенному пользователем целому числу.  2. Создать текстовый файл **F1** не менее, чем из 6 строк, и записать в него информацию. Скопировать в файл **F2** только четные строки из **F1**. |

1.

#include <stdio.h>

#include<iostream>

#include <locale>

using namespace std;

float randfloat() {//функция для генерации вещественных чисел со значением после запятой

return((float)(rand()) / ((float)(rand()) / 10)) - 10;

}

int main()

{

srand(time(0));

setlocale(LC\_ALL, "");

int a, number, ind = 0;

float array[15];

errno\_t errf1, errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.

FILE\* f, \* f2;//создаём указатели на файлы

errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи

if (errf1 != 0)//предусматриваем возможность ошибки

{

perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.

return EXIT\_FAILURE;

}

for (a = 0; a < 15; a++)

{

fprintf(f, "%f, ", randfloat());//генерируем числа в файл

}

printf("Данные записаны в file1.txt\n\n");

fclose(f);//закрываем файл

cout << "Введите целое число:";

cin >> number;

errf1 = fopen\_s(&f, "file1.txt", "r");//открываем тот же файл в режиме чтения

if (errf1 != 0)

{

perror("Невозможно создать файл\n");

return EXIT\_FAILURE;

}

errf2 = fopen\_s(&f2, "file2.txt ", "w");//открываем второй файл в режиме записи

if (errf2 != 0)

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

for (a = 0; a < 15; a++)

{

fscanf\_s(f, "%f, ", &array[a]);//считываем числа из первого файла в массив

}

cout << endl << "Содержимое file1.txt" << endl;

for (int i = 0; i < 15; i++) {

cout << array[i] << ", ";

if (abs(number - array[i]) < abs(number - array[ind])) { ind = i; }//ищем ближайшее по значению к введённому числу из сгенерированных

}

int arr\_size = 15;

cout << endl << "Содержимое file2.txt" << endl;

cout << array[ind] << ", ";

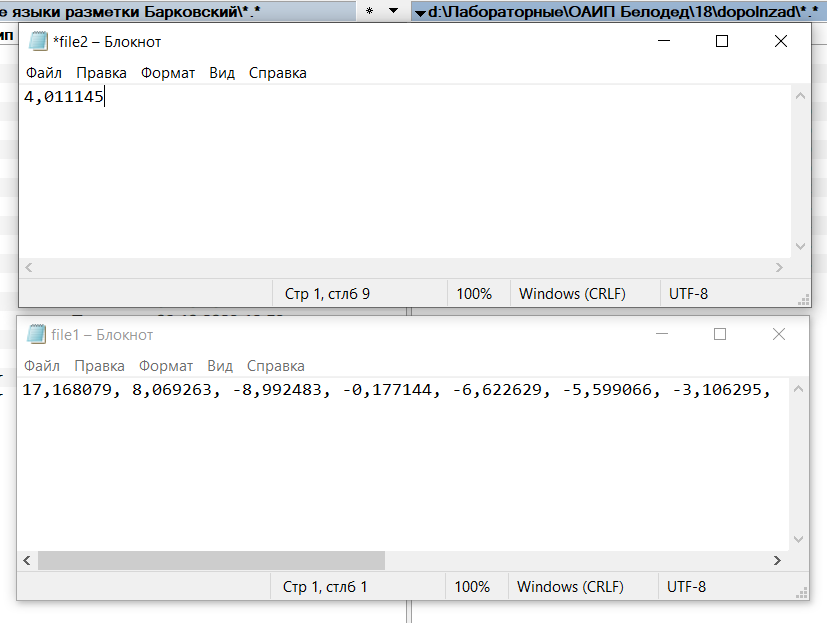
fprintf(f2, "%f, ", array[ind]);//печатаем нужное число в файл

fclose(f);

fclose(f2);//закрываем файлы, с которыми работали

return 0;

}



2.

#include <stdio.h>

#include<iostream>

#include <locale>

using namespace std;

int main()

{

srand(time(0));

setlocale(LC\_ALL, "");

int a, lenght;

float array[15];

char buffer[250];

char\* estr;

errno\_t errf1, errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.

FILE\* f, \* f2;//создаём указатели на файлы

errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи

if (errf1 != 0)//предусматриваем возможность ошибки

{

perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.

return EXIT\_FAILURE;

}

printf("Введите текст:\n");

for (a = 0; a < 6; a++)

{

gets\_s(buffer);//передаем введённую строку в буфер

fputs(buffer, f); fprintf(f, "\n");//печатаем буфер в файл

}

printf("Данные записаны в file1.txt\n");

fclose(f);//закрываем файл

errf2 = fopen\_s(&f2, "file2.txt", "w"); //Открытие для чтения

if (errf2 != NULL)

//Проверка открытия файла

{

printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;

}

errf1 = fopen\_s(&f, "file1.txt", "r"); //Открытие для чтения

if (errf1 != NULL)

//Проверка открытия файла

{

printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;

}

else

printf("\nЗапиcаны в file2.txt строки:\n");

for (int i = 0; i < 6; i++) {

fgets(buffer, sizeof(buffer), f); //Чтение строки файла

if (i % 2 != 0) {//если строка является четной, то записываем ее во второй файл

fputs(buffer, f2);

puts(buffer);

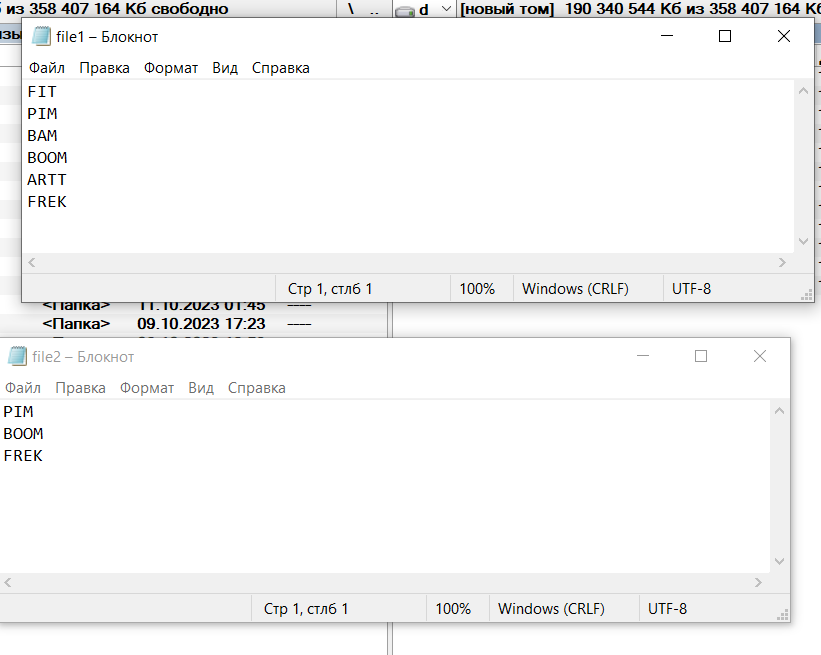
}

}

fclose(f);

fclose(f2);//закрываем файлы, с которыми работали

}



|  |  |
| --- | --- |
| **11** | 1. Компоненты файла **fileA** –  целые числа, значения которых повторяются.  Получить файл **fileB**, образованный из **fileA** числами, которые не повторяются.  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**». |

1.

#include <stdio.h>

#include<iostream>

#include <locale>

using namespace std;

int main()

{

srand(time(0));

setlocale(LC\_ALL, "");

int a;

int array[15];

errno\_t errf1, errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.

FILE\* f, \* f2;//создаём указатели на файлы

errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи

if (errf1 != 0)//предусматриваем возможность ошибки

{

perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.

return EXIT\_FAILURE;

}

for (a = 0; a < 15; a++)

{

fprintf(f, "%d, ", rand() % 10);//генерируем числа в файл

}

printf("Данные записаны в file1.txt\n");

fclose(f);//закрываем файл

errf1 = fopen\_s(&f, "file1.txt", "r");//открываем тот же файл в режиме чтения

if (errf1 != 0)

{

perror("Невозможно создать файл\n");

return EXIT\_FAILURE;

}

errf2 = fopen\_s(&f2, "file2.txt ", "w");//открываем второй файл в режиме записи

if (errf2 != 0)

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

for (a = 0; a < 15; a++)

{

fscanf\_s(f, "%d, ", &array[a]);//считываем числа из первого файла в массив

}

cout << "Содержимое file1.txt" << endl;

for (int i = 0; i < 15; i++) {

cout << array[i] << ", ";

}

int arr\_size = 15;

for (int i = 0; i < arr\_size; i++) {//удаляем повторяющиеся элементы из массива

for (int j = i + 1; j < arr\_size; j++) {

if (array[i] == array[j]) {

for (int k = j; k < arr\_size; k++) {

array[k] = array[k + 1];

}

j--;

arr\_size--;

}

}

}

cout << endl << "Содержимое file2.txt" << endl;

for (int i = 0; i < arr\_size; i++) {

cout << array[i] << ", ";

}

for (a = 0; a < arr\_size; a++)//записываем значения массива во второй файл

{

fprintf(f2, "%d, ", array[a]);

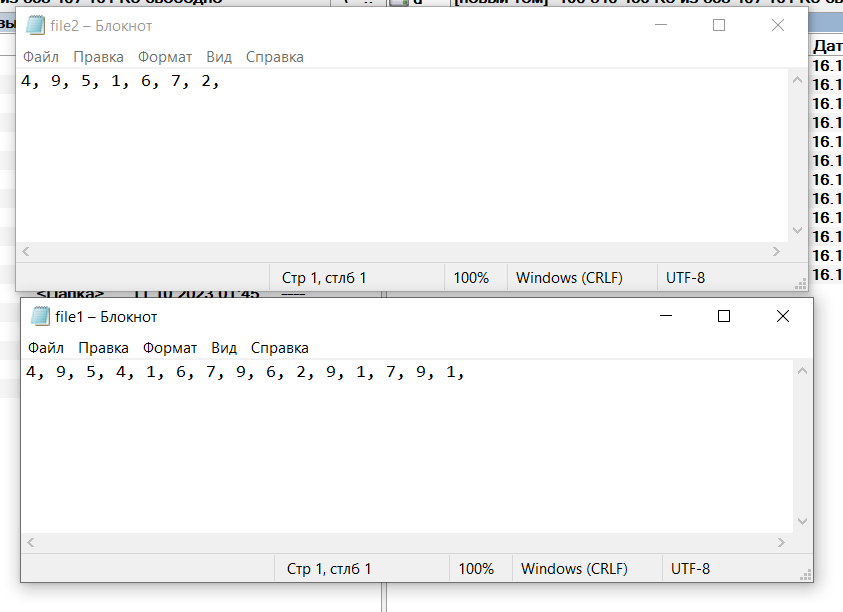
}

fclose(f);

fclose(f2);//закрываем файлы, с которыми работали

return 0;

}



2.

#include <stdio.h>

#include<iostream>

#include <locale>

using namespace std;

int main()

{

srand(time(0));

setlocale(LC\_ALL, "");

int a, lenght;

float array[15];

char buffer[250];

char\* estr;

errno\_t errf1, errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.

FILE\* f, \* f2;//создаём указатели на файлы

errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи

if (errf1 != 0)//предусматриваем возможность ошибки

{

perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.

return EXIT\_FAILURE;

}

printf("Введите текст:\n");

for (a = 0; a < 6; a++)

{

gets\_s(buffer);//передаем введённую строку в буфер

fputs(buffer, f); fprintf(f, "\n");//печатаем буфер в файл

}

printf("Данные записаны в file1.txt\n");

fclose(f);//закрываем файл

errf2 = fopen\_s(&f2, "file2.txt", "w"); //Открытие для чтения

if (errf2 != NULL)

//Проверка открытия файла

{

printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;

}

errf1 = fopen\_s(&f, "file1.txt", "r"); //Открытие для чтения

if (errf1 != NULL)

//Проверка открытия файла

{

printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;

}

else

printf("\nЗапсианы в file2.txt строки:\n");

for (int i = 0; i < 6; i++) {

fgets(buffer, sizeof(buffer), f); //Чтение строки файла

if (buffer[strlen(buffer) - 2] == 'a') {//если строка кончается символом "a", то записываем ее во второй файл

fputs(buffer, f2);

puts(buffer);

}

}

fclose(f);

fclose(f2);//закрываем файлы, с которыми работали

}

