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**»

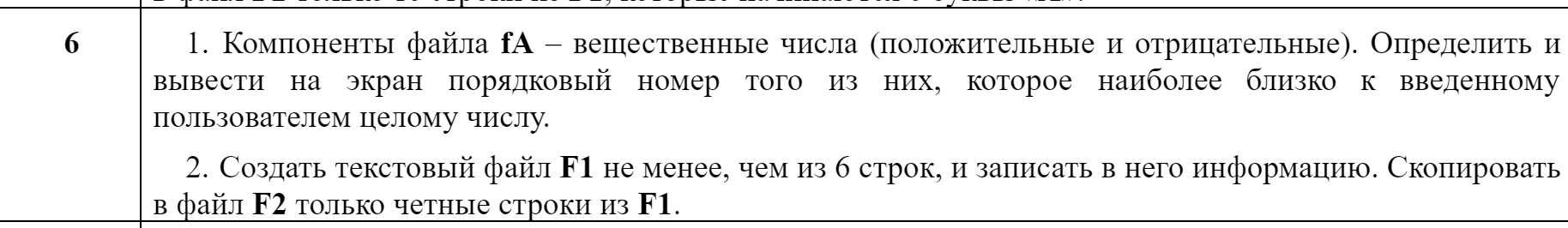
Выполнила:

Студент(ка) 1 курса 7 группы

Гриценко Анна Александровна

Преподаватель: асс. Андронова М.В.

2023, Минск



#include <iostream>

#include <stdio.h>

#include <stdlib.h>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

FILE\* fileA;

double targetNumber, currentNumber;

int lineNumber = 0, closestLineNumber = 0;

double closestDistance = -1;// Используем отрицательное значение для инициализации

// ввод числа пользователем

cout << "Введите целое число: ";

cin >> targetNumber;

//чтение файла с исходными данными

fopen\_s(&fileA, "C:\\Users\\admin\\Documents\\файлы\\fA.txt", "r");

if (fileA == NULL) {

perror("Ошибка открытия файла fA.txt");

return 1;

}

// анализ вещественных чисел из файла

while (fscanf\_s(fileA, "%lf", &currentNumber) != EOF) {

lineNumber++;

// Определение близости текущего числа к введенному

double distance = abs(targetNumber - currentNumber);

// Если это ближайшее число, обновляем переменные

if (closestDistance == -1 || distance < closestDistance) {

closestDistance = distance;

closestLineNumber = lineNumber;

}

}

// Вывод а

// резулятата работы программы на экран

cout << "Cамое близкое число к " << targetNumber << " это " << currentNumber

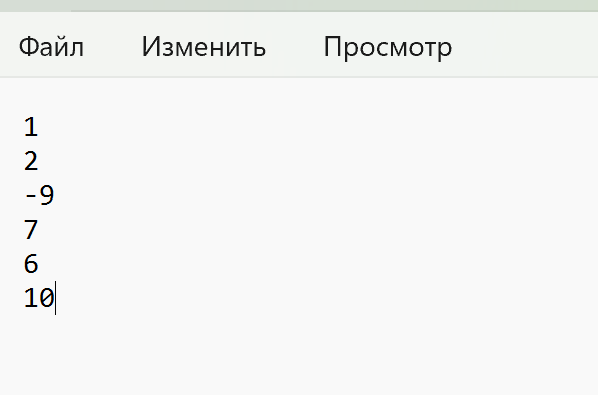
<< " его позиция " << closestLineNumber << " в файле" << endl;

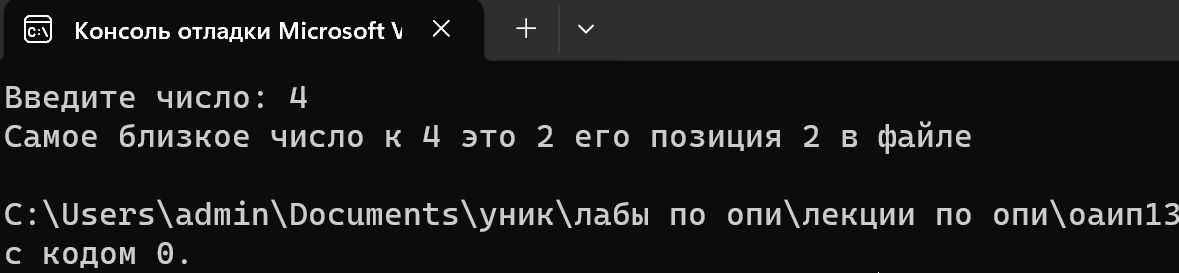
// Закрытие файла

fclose(fileA);

return 0;

}





**Задание 2**

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

FILE\* filefA, \* filefB;

char line[256]; //Указываем фиксированное кол-во символов для строки

int lineNumber = 0;

// Создание и запись в файл fA.txt

fopen\_s(&filefA, "C:\\Users\\admin\\Documents\\файлы\\fA.txt", "w");

if (filefA == NULL) {

perror("Ошибка открытия файла fA.txt");

return 1;

}

fprintf(filefA, "Первая строка\n");

fprintf(filefA, "Вторая строка\n");

fprintf(filefA, "Третья строка\n");

fprintf(filefA, "Четвертая строка\n");

fprintf(filefA, "Пятая строка\n");

fprintf(filefA, "Шестая строка\n");

// Закрываем файл fA.txt

fclose(filefA);

// Копирование четных строк в fB.txt

fopen\_s(&filefA, "fA.txt", "r");

fopen\_s(&filefB, "fB.txt", "w");

if (filefA == NULL || filefB == NULL) {

perror("Ошибка открытия файлов fA.txt или fB.txt");

return 1;

}

// Чтение и копирование из файла четных строк

while (fgets(line, sizeof(line), filefA) != NULL) {

lineNumber++;

if (lineNumber % 2 == 0) {

fprintf(filefB, "%s", line);

}

}

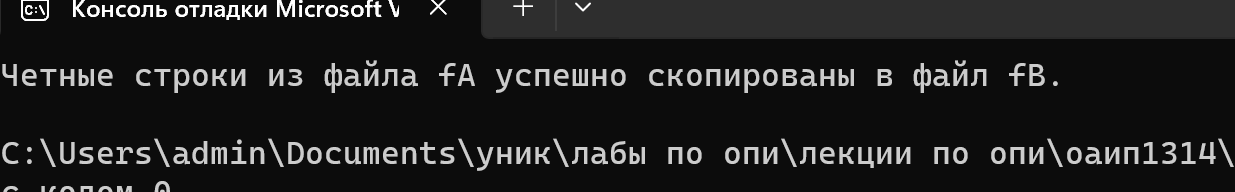
// Закрытие файлов

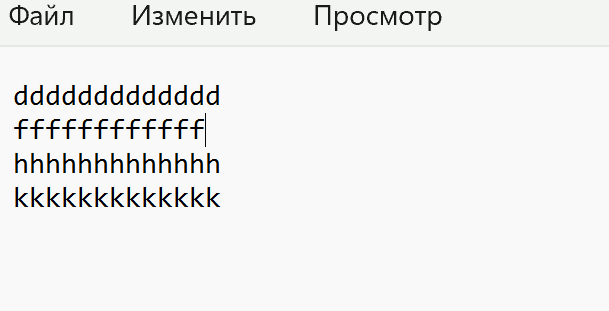
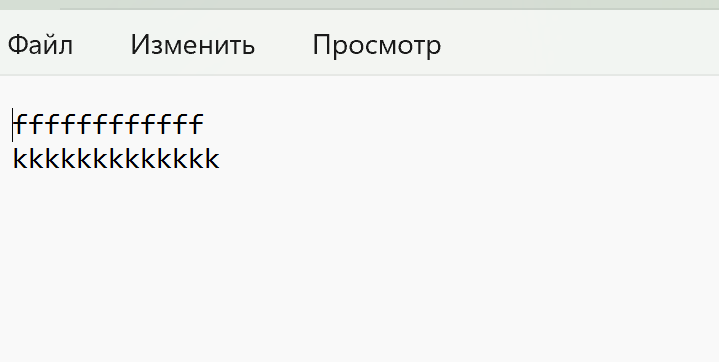
fclose(filefA);

fclose(filefB);

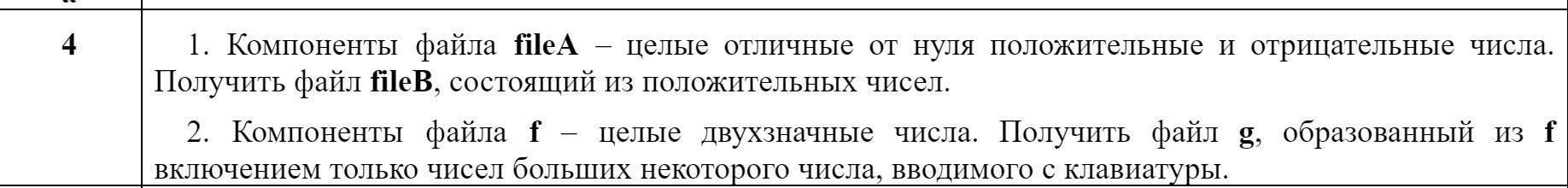
return 0;

}



**** ****

**Дополнтьельные варианты**

****

**Задание 1**

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

FILE\* fileA, \* fileB;

int number;

// Открываем файл fileA для чтения

fileA = fopen("fileA.txt", "r");

if (fileA == NULL) {

perror("Ошибка открытия файла fileA");

return 1;

}

// Открываем файл fileB для записи

fileB = fopen("fileB.txt", "w");

if (fileB == NULL) {

perror("Ошибка открытия файла fileB");

fclose(fileA);

return 1;

}

// Читаем числа из файла fileA и записываем положительные числа в файл fileB

while (fscanf(fileA, "%d", &number) == 1) {

if (number > 0) {

fprintf(fileB, "%d\n", number);

}

}

// Закрываем файлы

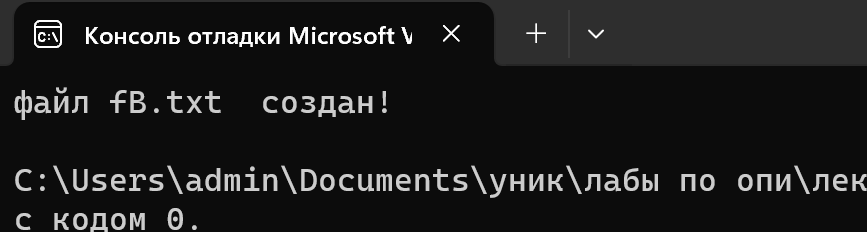
fclose(fileA);

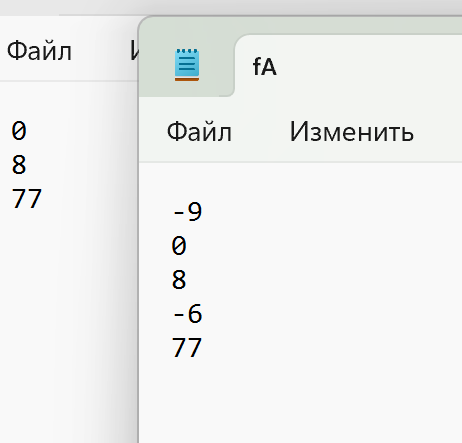
fclose(fileB);

printf("Файл fB.txt созданы\n");

return 0;

}

****

****

**Задание 2**

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

FILE\* fileF, \* fileG;

int threshold, number;

// Вводим пороговое значение с клавиатуры

printf("Введите пороговое значение: ");

scanf("%d", &threshold);

// Открываем файл f для чтения

fileF = fopen("f.txt", "r");

if (fileF == NULL) {

perror("Ошибка открытия файла f");

return 1;

}

// Открываем файл g для записи

fileG = fopen("g.txt", "w");

if (fileG == NULL) {

perror("Ошибка открытия файла g");

fclose(fileF);

return 1;

}

// Читаем числа из файла f и записываем числа большие порогового значения в файл g

while (fscanf(fileF, "%d", &number) == 1) {

if (number > threshold) {

fprintf(fileG, "%d\n", number);

}

}

// Закрываем файлы

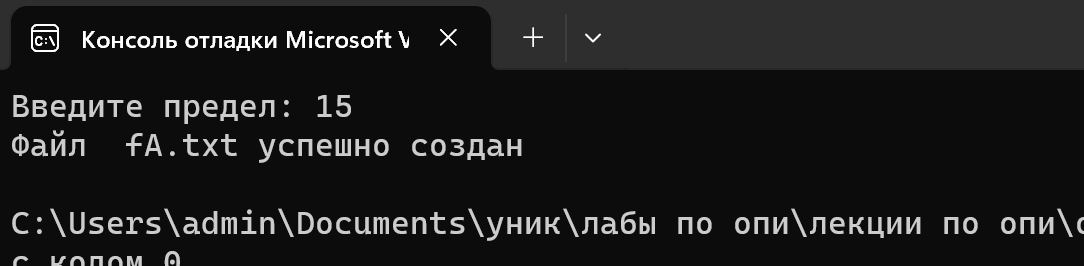
fclose(fileF);

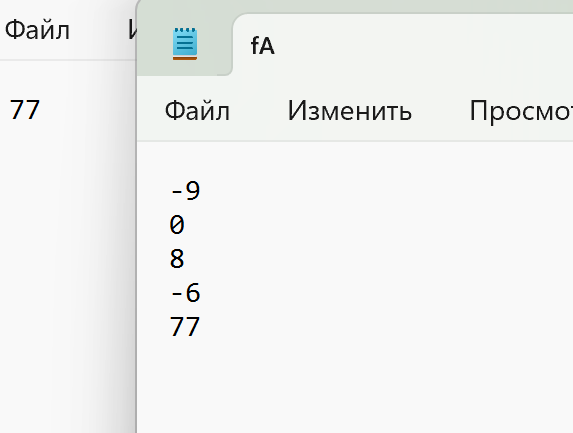
fclose(fileG);

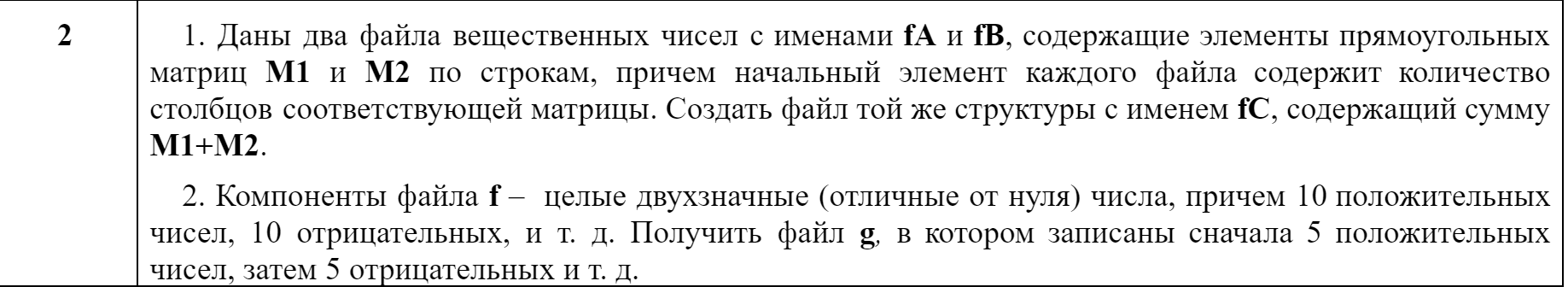
printf("Файл fA.txt успешно создан\n");

return 0;

}

****

****

****

**Задание 1**

#include <stdio.h>

using namespace std;

int main() {

//объявление указателей на файлы

FILE\* fileA, \* fileB, \* fileC;

//объявление переменных для хранения информации

int columnsA, columnsB;

// Открываем файлы для чтения

fopen\_s(&fileA, "fA.txt", "r");

fopen\_s(&fileB, "fB.txt", "r");

//Создание файла

fopen\_s(&fileC, "fC.txt", "w");

// Проверка на открытие файлов

if (fileA == NULL || fileB == NULL || fileC == NULL) {

perror("Ошибка открытия файлов");

return 1;

}

// просмотр количества столбцов из файлов

fscanf\_s(fileA, "%d", &columnsA);

fscanf\_s(fileB, "%d", &columnsB);

// Проверка на размерности матриц

if (columnsA != columnsB) {

printf("Ошибка: количество столбцов матриц не совпадает\n");

return 1;

}

// Запись количества столбцов в файл-результат

fprintf(fileC, "%d\n", columnsA);

// Сложение матриц и запись результата в файл-результат

double elementA, elementB;

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

fscanf\_s(fileA, "%lf", &elementA);

fscanf\_s(fileB, "%lf", &elementB);

fprintf(fileC, "%.2f ", elementA + elementB);

}

cout << "Матрицы сложены, результат записан в файл fC.txt." << endl;

// Закрытие файлов

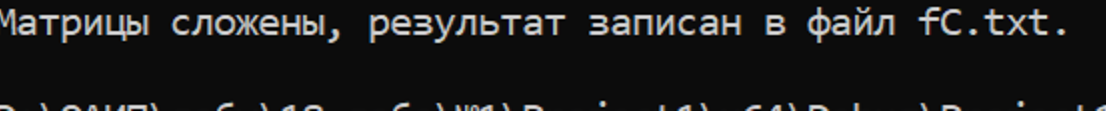
fclose(fileA);

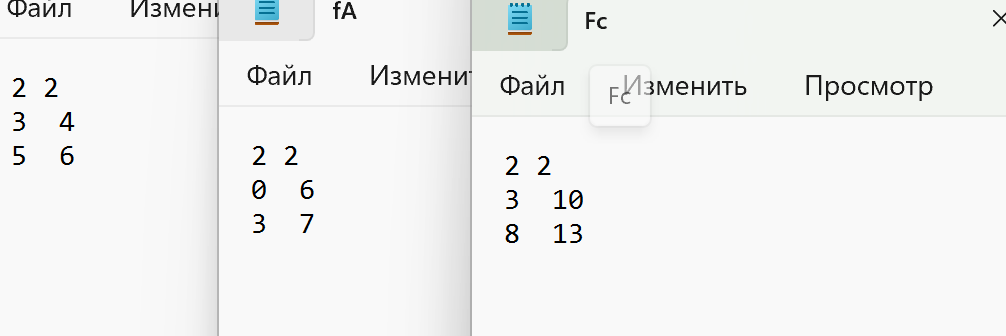
fclose(fileB);

fclose(fileC);

return 0;

}

****

****

**Задание 2**

#include <stdio.h>

int main() {

FILE\* fileF, \* fileG;

int numbers[20]; // Массив для хранения чисел из файла f.txt

// Открываем файлов

fopen\_s(&fileF, "f.txt", "r");

fopen\_s(&fileG, "g.txt", "w");

// Проверка на открытие файлов

if (fileF == NULL || fileG == NULL) {

perror("Ошибка открытия файлов");

return 1;

}

// Чтение чисел из файла f.txt

for (int i = 0; i < 20; i++) {

fscanf\_s(fileF, "%d", &numbers[i]);

}

// Запись чисел в файл g.txt

for (int i = 0; i < 5; i++) {

fprintf(fileG, "%d\n", numbers[i]);

fprintf(fileG, "%d\n", numbers[i + 10]); // Сдвиг на 10 для отрицательных чисел

}

std::cout << "Файл g.txt создан" << std::endl;

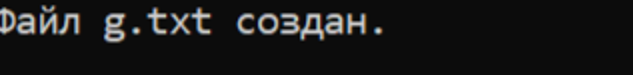
// Закрытие файлов

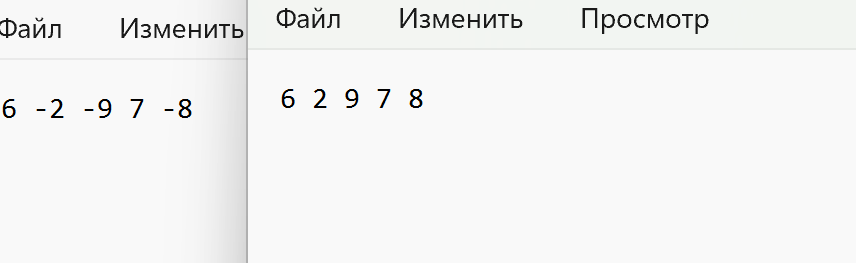
fclose(fileF);

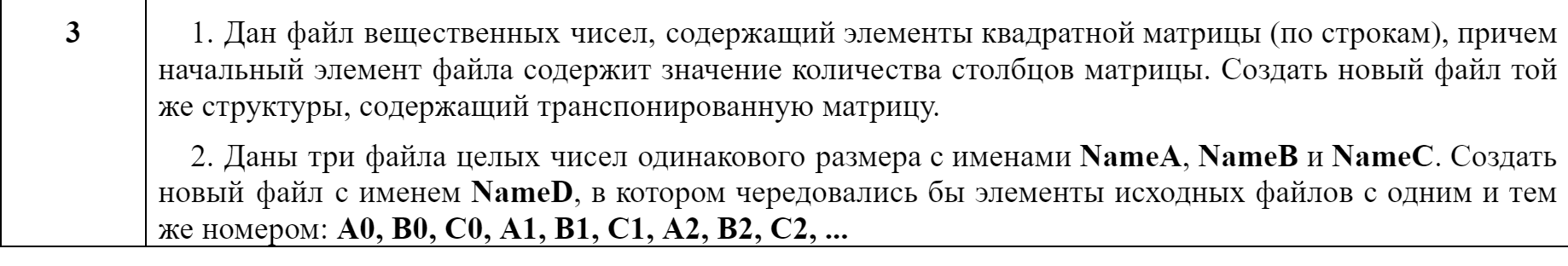
fclose(fileG);

return 0;

}







**Задание 1**

#include <stdio.h>

using namespace std;

int main() {

FILE\* inputFile, \* outputFile;

int cols;

// Открываем файла

inputFile = fopen("input.txt", "r");

if (inputFile == NULL) {

perror("Ошибка открытия входного файла");

return 1;

}

// просмотр количества столбцов из файла

fscanf\_s(inputFile, "%d", &cols);

//запись результата в файл

outputFile = fopen("output.txt", "w");

if (outputFile == NULL) {

perror("Ошибка открытия выходного файла");

fclose(inputFile);

return 1;

}

// Запись количества столбцов в новый файл

fprintf(outputFile, "%d\n", cols);

//динамическое выделение памяти под матрицу

double\*\* matrix = (doube\*\*)malloc(cols \* sizeof(double\*));

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

matrix[i] = (double\*)malloc(cols \* sizeof(double));

}

// Чтение и транспонирование матрицы

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

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

fscanf\_s(inputFile, "%lf", &matrix[i][j]);

}

}

// Запись транспонированной матрицы в новый файл

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

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

fprintf(outputFile, "%.2f ", matrix[j][i]);

}

fprintf(outputFile, "\n");

}

// Освобождение памяти

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

free(matrix[i]);

}

free(matrix);

cout << "Транспонированная матрица записана в файл " << outputFile << endl;

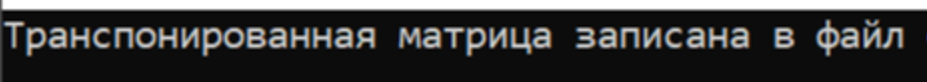
// Закрытие файлов

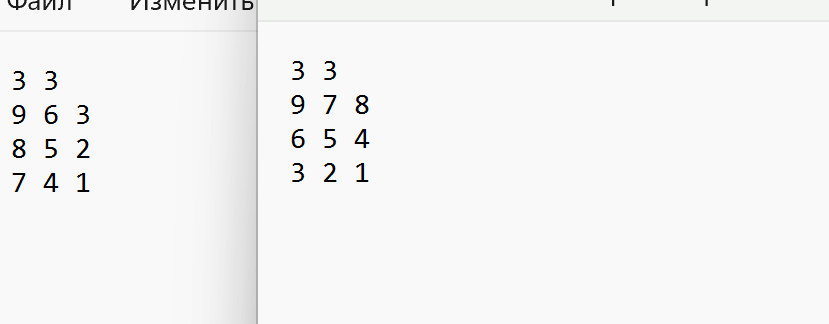
fclose(inputFile);

fclose(outputFile);

return 0;

}

****

****

**Задание 2**

#include <stdio.h>

using namespace std;

int main() {

FILE\* fileA, \* fileB, \* fileC, \* fileD;

int size; // Размер файлов

// Открытие файлов

fopen\_s(&fileA, "NameA.txt", "r");

fopen\_s(&fileB, "NameB.txt", "r");

fopen\_s(&fileC, "NameC.txt", "r");

fopen\_s(&fileD, "NameD.txt", "w");

// Проверка на открытие файлов

if (fileA == NULL || fileB == NULL || fileC == NULL || fileD == NULL) {

perror("Ошибка открытия файлов");

return 1;

}

// просмотр размера файлов

fscanf\_s(fileA, "%d", &size);

// Запись размера в новый файл

fprintf(fileD, "%d\n", size);

// Чередование элементов и запись в новый файл

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

int elementA, elementB, elementC;

fscanf\_s(fileA, "%d", &elementA);

fscanf\_s(fileB, "%d", &elementB);

fscanf\_s(fileC, "%d", &elementC);

fprintf(fileD, "%d %d %d ", elementA, elementB, elementC);

}

scout << "Файл " << nameD << " успешно создан" << endl;

// Закрытие файлов

fclose(fileA);

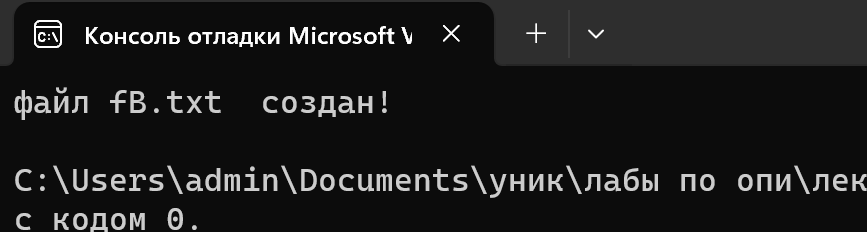
fclose(fileB);

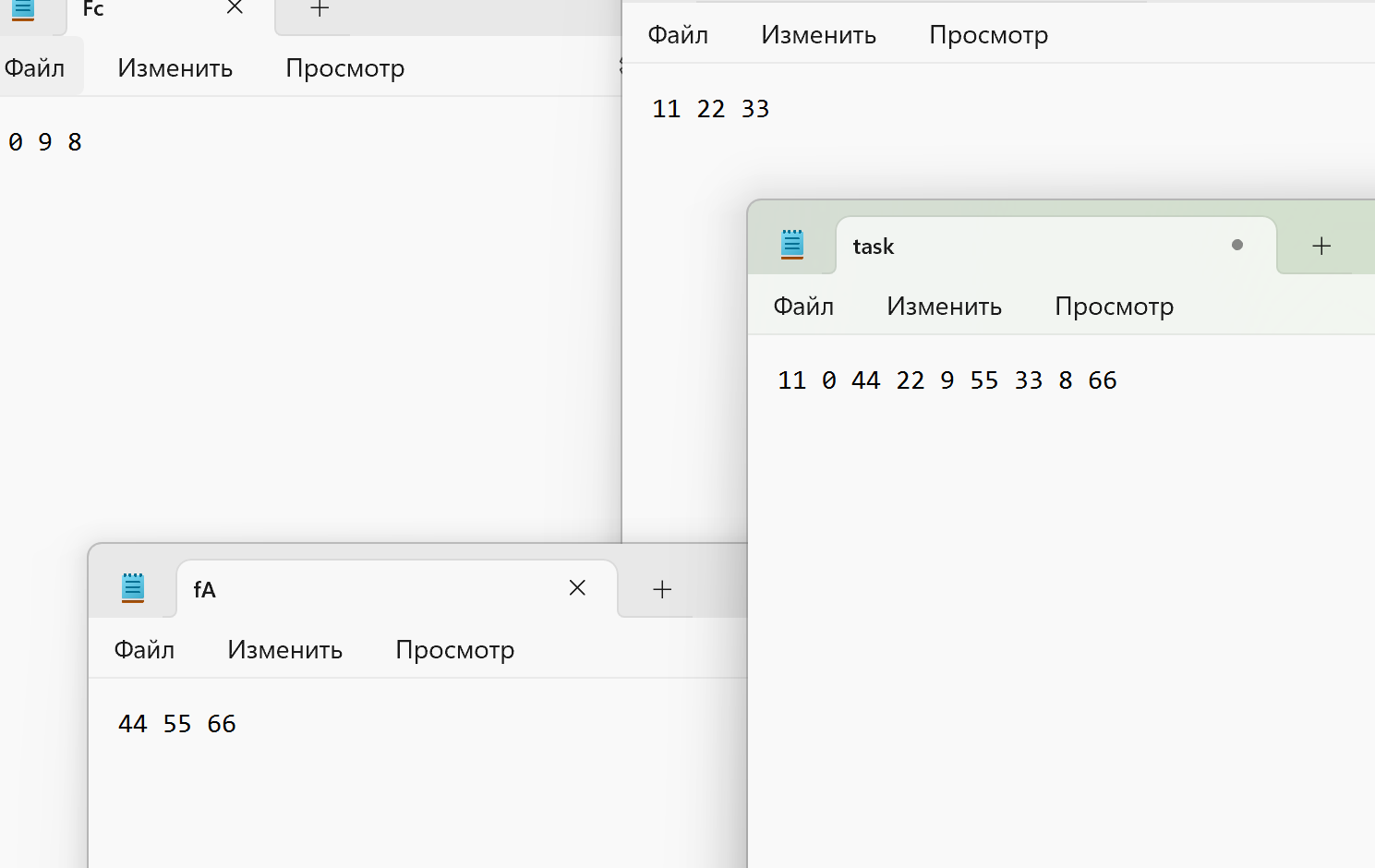
fclose(fileC);

fclose(fileD);

return 0;

}

****

****