**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:

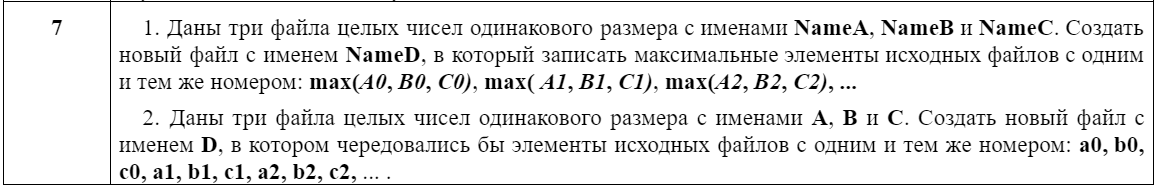
Студентка 1 курса 7 группы

Шинкевич Марина Дмитриевна

Преподаватель: асс. Андронова М.В.

2023, Минск

Дополнительные задания к лабораторной 18



#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main() {

FILE\* File\_A, \* File\_B, \* File\_C, \* File\_D;//Объявляет указатели на файлы File\_A, File\_B, File\_C, File\_D.

//Объявляет массив символов nam\_A и инициализирует его строкой "NameA.txt".Аналогично для файлов B, C и D.

char nam\_A[] = "NameA.txt";

char nam\_B[] = "NameB.txt";

char nam\_C[] = "NameC.txt";

char nam\_D[] = "NameD.txt";

//Открывает файлы File\_A, File\_B, File\_C для чтения. Если хотя бы один из них не открывается успешно, выводит сообщение об ошибке и завершает программу с кодом 1.

if (fopen\_s(&File\_A, nam\_A, "r") != 0 || fopen\_s(&File\_B, nam\_B, "r") != 0 || fopen\_s(&File\_C, nam\_C, "r") != 0) {

printf("Ошибка\n");

return 1;

}

//Открывает файл File\_D для записи. Если файл не открывается успешно, выводит сообщение об ошибке и завершает программу с кодом 1.

if (fopen\_s(&File\_D, nam\_D, "w") != 0) {

printf("Ошибка\n");

return 1;

}

int number\_A, number\_B, number\_C;

while (fscanf(File\_A, "%d", &number\_A) == 1 && fscanf(File\_B, "%d", &number\_B) == 1 && fscanf(File\_C, "%d", &number\_C) == 1) {//Начинается цикл, который будет выполняться до тех пор, пока успешно удается считать по одному числу из каждого из файлов File\_A, File\_B, File\_C.

int max = (number\_A > number\_B) ? number\_A : number\_B;//Определяет максимальное число из number\_A и number\_B.

max = (max > number\_C) ? max : number\_C;//Сравнивает найденное максимальное число с number\_C и определяет окончательное максимальное число.

fprintf(File\_D, "%d\n", max);//Записывает найденное максимальное число в файл File\_D.

}

// Закрывает все открытые файлы.

fclose(File\_A);

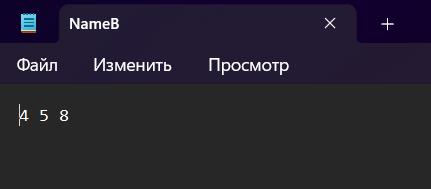
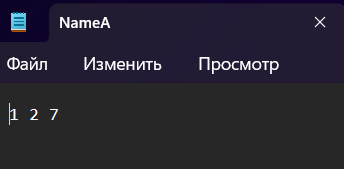
fclose(File\_B);

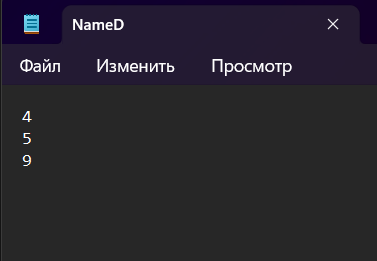
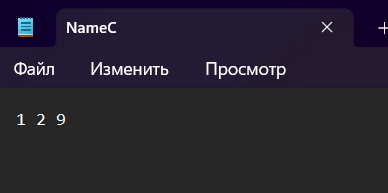
fclose(File\_C);

fclose(File\_D);

return 0;

}





#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main() {

FILE\* File\_A, \* File\_B, \* File\_C, \* File\_D;//Объявляет указатели на файлы File\_A, File\_B, File\_C, File\_D.

//Объявляет массив символов nam\_A и инициализирует его строкой "NameA.txt".Аналогично для файлов B, C и D.

char nam\_A[] = "NameA.txt";

char nam\_B[] = "NameB.txt";

char nam\_C[] = "NameC.txt";

char nam\_D[] = "NameD.txt";

//Открывает файлы File\_A, File\_B, File\_C для чтения. Если хотя бы один из них не открывается успешно, выводит сообщение об ошибке и завершает программу с кодом 1.

if (fopen\_s(&File\_A, nam\_A, "r") != 0 || fopen\_s(&File\_B, nam\_B, "r") != 0 || fopen\_s(&File\_C, nam\_C, "r") != 0) {

printf("Ошибка\n");

return 1;

}

//Открывает файл File\_D для записи. Если файл не открывается успешно, выводит сообщение об ошибке и завершает программу с кодом 1.

if (fopen\_s(&File\_D, nam\_D, "w") != 0) {

printf("Ошибка\n");

return 1;

}

int number\_A, number\_B, number\_C;

//Начинается цикл while, который будет выполняться до тех пор, пока успешно удается считать по одному числу из каждого из файлов File\_A, File\_B, File\_C. Условие в скобках проверяет успешность считывания чисел из каждого файла.

while (fscanf(File\_A, "%d", &number\_A) == 1 && fscanf(File\_B, "%d", &number\_B) == 1 && fscanf(File\_C, "%d", &number\_C) == 1) {

\

fprintf(File\_D, "%d ", number\_A);

fprintf(File\_D, "%d ", number\_B);

fprintf(File\_D, "%d ", number\_C);

}

// Закрывает все открытые файлы.

fclose(File\_A);

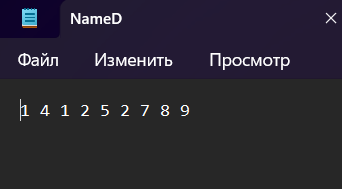
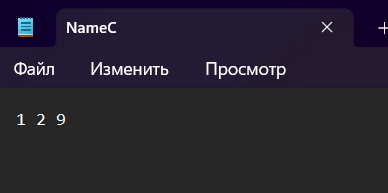
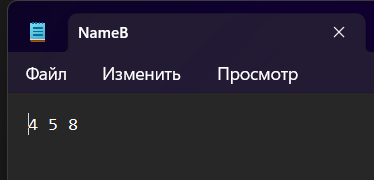
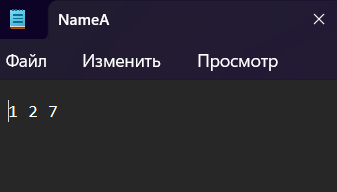
fclose(File\_B);

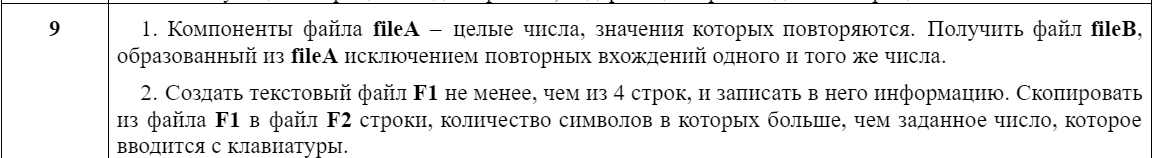
fclose(File\_C);

fclose(File\_D);

return 0;

}





#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main() {

FILE\* FilA, \* FilB;//Объявляет два указателя на файл FilA и FilB.

// Определяют имена файлов, с которыми будет работать программа.

char File\_A[] = "NameA.txt";

char File\_B[] = "NameB.txt";

//Открывают файлы File\_A для чтения и File\_B для записи.

FilA = fopen(File\_A, "r");

FilB = fopen(File\_B, "w");

//Проверяет, были ли успешно открыты оба файла. Если хотя бы один из них не удалось открыть, программа выводит сообщение об ошибке и завершает свою работу с кодом возврата 1.

if (FilA == NULL || FilB == NULL) {

printf("Ошибка\n");

return 1;

}

int num;

// Объявляет массив mass размером 10000 элементов и инициализирует его нулями. Этот массив будет использоваться для отслеживания уникальных чисел, считанных из файла.

int mass[10000] = { 0 };

//Начинается цикл while, который будет выполняться до тех пор, пока успешно удается считать числа из файла FilA. Условие в скобках проверяет успешность считывания числа.

while (fscanf(FilA, "%d", &num) == 1) {

if (mass[num] == 0) {//Проверяет, было ли уже считано число num. Если оно не встречалось ранее, выполняется следующий блок кода.

fprintf(FilB, "%d ", num);

mass[num] = 1;//Устанавливает флаг в массиве exists, указывающий на то, что число num уже было обработано.

}

}

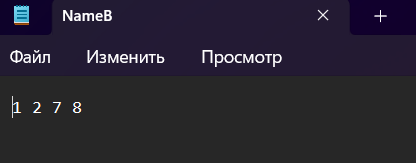
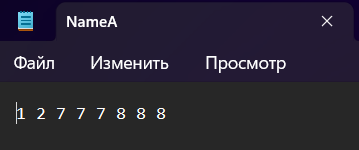
//Закрывают файлы FilA и FilB после завершения работы с ними

fclose(FilA);

fclose(FilB);

return 0;

}



#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main() {

FILE\* FilA, \* FilB;//Объявляет два указателя на файл FilA и FilB.

// Определяют имена файлов, с которыми будет работать программа.

char File\_A[] = "NameA.txt";

char File\_B[] = "NameB.txt";

int X;

printf("X: ");

scanf("%d", &X); //ввод из консоли значения

//Открывают файлы File\_A для чтения и File\_B для записи.

FilA = fopen(File\_A, "r");

FilB = fopen(File\_B, "w");

//Проверяет, были ли успешно открыты оба файла. Если хотя бы один из них не удалось открыть, программа выводит сообщение об ошибке и завершает свою работу с кодом возврата 1.

if (FilA == NULL || FilB == NULL) {

printf("Ошибка\n");

return 1;

}

char line[100];

while (fgets(line, sizeof(line), FilA) != NULL) { //поиск строки большей, чем значение X

if (strlen(line) > X) {

fputs(line, FilB);

}

}

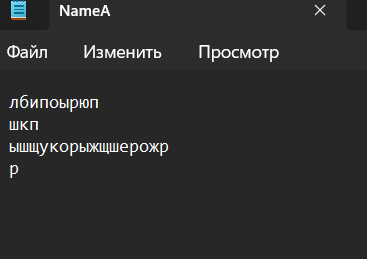
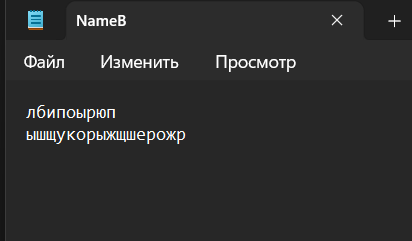
//Закрывают файлы FilA и FilB после завершения работы с ними

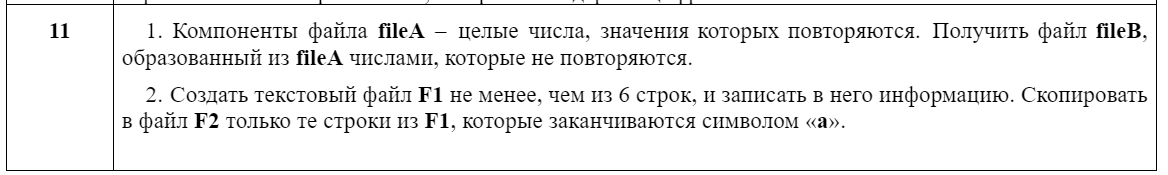
fclose(FilA);

fclose(FilB);

return 0;

}



#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main() {

FILE\* FilA, \* FilB;//Объявляет два указателя на файл FilA и FilB.

// Определяют имена файлов, с которыми будет работать программа.

char File\_A[] = "NameA.txt";

char File\_B[] = "NameB.txt";

//Открывают файлы File\_A для чтения и File\_B для записи.

FilA = fopen(File\_A, "r");

FilB = fopen(File\_B, "w");

//Проверяет, были ли успешно открыты оба файла. Если хотя бы один из них не удалось открыть, программа выводит сообщение об ошибке и завершает свою работу с кодом возврата 1.

if (FilA == NULL || FilB == NULL) {

printf("Ошибка\n");

return 1;

}

int mass[100] = { 0 }; // Объявляет массив mass размером 100 элементов и инициализирует его нулями. Этот массив будет использоваться для подсчета количества вхождений каждого числа из файла FilA.

int num;

while (fscanf(FilA, "%d", &num) == 1) { ////Начинается цикл while, который будет выполняться до тех пор, пока успешно удается считать числа из файла FilA. Условие в скобках проверяет успешность считывания числа.

mass[num]++;// Увеличивает значение элемента массива mass, соответствующего считанному числу num, на 1. Таким образом, в массиве mass подсчитывается количество вхождений каждого числа.

}

fseek(FilA, 0, SEEK\_SET); //Перемещает указатель текущей позиции в файле FilA обратно в начало файла для повторного чтения.

while (fscanf(FilA, "%d", &num) == 1) {//цикл while также считывает числа из файла FilA

if (mass[num] == 1) {//Проверяет, сколько раз число num встречается в файле FilA

fprintf(FilB, "%d ", num);//Записывает значение числа num в файл FilB

mass[num] = -1; //Помечает число num, которое уже было записано в файл FilB, устанавливая значение - 1 в массиве mass.

}

}

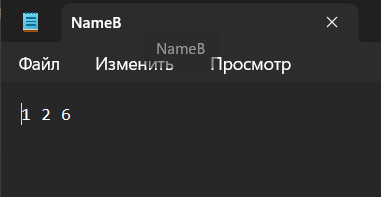
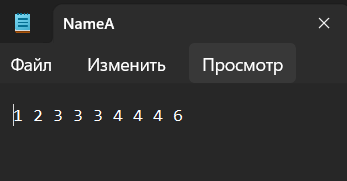
//Закрывают файлы FilA и FilB после завершения работы с ними

fclose(FilA);

fclose(FilB);

return 0;

}



#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main() {

FILE\* FilA, \* FilB;//Объявляет два указателя на файл FilA и FilB.

// Определяют имена файлов, с которыми будет работать программа.

char File\_A[] = "NameA.txt";

char File\_B[] = "NameB.txt";

char line[100];

//Открывают файлы File\_A для чтения и File\_B для записи.

FilA = fopen(File\_A, "r");

FilB = fopen(File\_B, "w");

//Проверяет, были ли успешно открыты оба файла. Если хотя бы один из них не удалось открыть, программа выводит сообщение об ошибке и завершает свою работу с кодом возврата 1.

if (FilA == NULL || FilB == NULL) {

printf("Ошибка\n");

return 1;

}

while (fgets(line, sizeof(line), FilA) != NULL) {// Начинается цикл while, который будет выполняться до тех пор, пока успешно удается считать строку из файла FilA. Функция fgets считывает строку из файла FilA и сохраняет ее в массив line.

if (line[strlen(line) - 2] == 'a') { //Проверяет последний символ строки, считанной из файла FilA.

fputs(line, FilB);//Записывает строку, считанную из файла FilA, в файл FilB.

}

}

//Закрывают файлы FilA и FilB после завершения работы с ними

fclose(FilA);

fclose(FilB);

return 0;

}

