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 10 группы

Рублевская Маргарита Владимировна

Преподаватель: асс. Андронова М.В.

2023, Минск

**Вариант№ 13**

1. 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.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Директивы препроцессора в языке С

#include <stdio.h>

#include <stdlib.h>

int main() {

setlocale(LC\_ALL, "");

FILE\* fileA, \* fileB; // Объявляем два указателя на тип FILE

int num; // Объявляем переменную типа int с именем num

int count[1000] = { 0 }; // Максимальное значение числа в fileA

// Открываем файл fileA для чтения

errno\_t A = fopen\_s(&fileA, "fileA.txt", "r");

if (fileA == NULL) {

printf("Не удалось открыть файл fileA.txt\n");

return 1;

}

// Считываем числа из fileA и считаем их количество

while (fscanf\_s(fileA, "%d", &num) != EOF) {

count[num]++;

}

// Закрываем файл fileA

if (fclose(fileA) != 0) {

printf("Ошибка при закрытии файла fileA.txt\n");

return 1;

}

// Открываем файл fileB для записи

errno\_t B = fopen\_s(&fileB, "fileB.txt", "w");

if (fileB == NULL) {

printf("Не удалось открыть файл fileB.txt.n");

return 1;

}

// Повторное открытие файла fileA для чтения

errno\_t A1 = fopen\_s(&fileA, "fileA.txt", "r");

if (fileA == NULL) {

printf("Не удалось повторно открыть файл fileA.txt\n");

return 1;

}

// Записываем в fileB числа, которые встречаются в fileA более двух раз

while (fscanf\_s(fileA, "%d", &num) != EOF) {

if (count[num] > 2) {

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

}

}

// Закрываем файл fileA

if (fclose(fileA) != 0) {

printf("Ошибка при закрытии файла fileA.txt\n");

return 1;

}

// Закрываем файл fileB

if (fclose(fileB) != 0) {

printf("Ошибка при закрытии файла fileB.txt\n");

return 1;

}

// Открываем файл fileB для чтения и выводим его содержимое на экран

errno\_t B1 = fopen\_s(&fileB, "fileB.txt", "r");

if (fileB == NULL) {

printf("Не удалось открыть файл fileB.txt\n");

return 1;

}

printf("Содержимое файла fileB:\n");

while (fscanf\_s(fileB, "%d", &num) != EOF) {

printf("%d\n", num);

}

// Закрываем файл fileB

if (fclose(fileB) != 0) {

printf("Ошибка при закрытии файла fileB.txt\n");

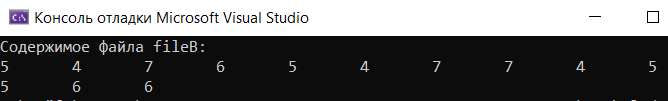
return 1;

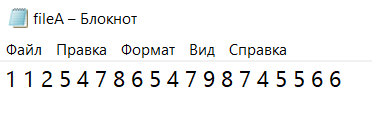
}

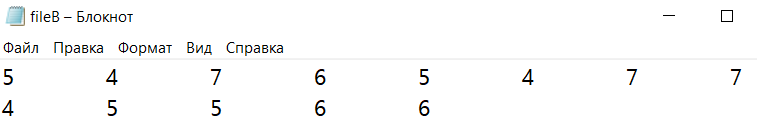
return 0;

}

Консоль VS:







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.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Директивы препроцессора в языке С

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

int main() {

setlocale(LC\_ALL, "");

FILE\* fileF1; // Объявляем указатель на тип FILE

// Открываем файл fileA для чтения

errno\_t F1 = fopen\_s(&fileF1, "F1.txt", "r");

if (F1 != 0 || fileF1 == NULL) {

perror("Не удалось открыть файл F1.txt\n");

return 1;

}

FILE\* fileF2; // Объявляем указатель на тип FILE

// Открываем файл fileB для чтения

errno\_t F2 = fopen\_s(&fileF2, "F2.txt", "w");

if (F2 != 0 || fileF2 == NULL) {

perror("Не удалось открыть файл F2.txt\n");

fclose(fileF1);

return 1;

}

// Считываем строки из файла "F1.txt" по очереди и проверяем, начинается ли каждая строка с цифры

char buffer[1000];

while (fgets(buffer, sizeof(buffer), fileF1)) { // Если строка начинается с цифры, то она записывается в файл "F2.txt"

if (isdigit(buffer[0])) {

fprintf(fileF2, "%s", buffer);

}

}

// Закрываем файлы fileF1 и fileF2

fclose(fileF1);

fclose(fileF2);

// Повторное открытие файлjd fileF1 и fileF2 для чтения

errno\_t F1\_1 = fopen\_s(&fileF1, "F1.txt", "r");

errno\_t F2\_1 = fopen\_s(&fileF2, "F2.txt", "r");

if (F1\_1 != 0 || F2\_1 != 0 || fileF1 == NULL || fileF2 == NULL) {

printf("Ошибка при открытии файлов F1.txt или F2.txt\n");

return 1;

}

// Выводим содержимое файла fileF1 на экран

printf("Содержимое файла F1.txt:\n");

while (fgets(buffer, sizeof(buffer), fileF1)) {

printf("%s", buffer);

}

printf("\n");

// Выводим содержимое файла fileF1 на экран

printf("Содержимое файла F2.txt (только строки начинающиеся с цифры):\n");

while (fgets(buffer, sizeof(buffer), fileF2)) {

printf("%s", buffer);

}

// Закрываем файлы fileF1 и fileF2

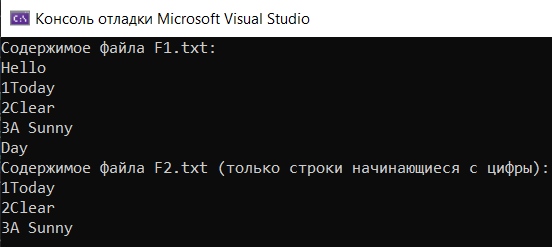
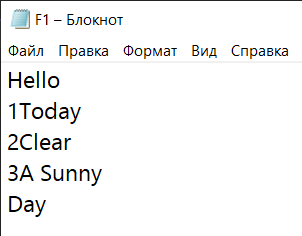
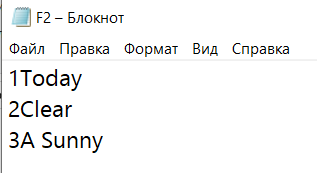
fclose(fileF1);

fclose(fileF2);

return 0;

}

Консоль VS:

****

**Вариант№ 4**

1. Компоненты файла **fileA** – целые отличные от нуля положительные и отрицательные числа. Получить файл **fileB**, состоящий из положительных чисел.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Директивы препроцессора в языке С

#include <stdio.h>

#include <stdlib.h>

int main() {

setlocale(LC\_ALL, "");

FILE\* fileA; // Объявляем указатель на тип FILE

// Открываем файл fileA для чтения

errno\_t A1 = fopen\_s(&fileA, "fileA.txt", "r");

if (fileA == NULL) {

perror("Не удалось открыть файл fileA.txt\n");

return 1;

}

FILE\* fileB; // Объявляем указатель на тип FILE

// Открываем файл fileB для чтения

errno\_t B1 = fopen\_s(&fileB, "fileB.txt", "w");

if (fileB == NULL) {

perror("Не удалось открыть файл fileB.txt\n");

fclose(fileA);

return 1;

}

int num;

while (fscanf\_s(fileA, "%d", &num) == 1) { // Читаем числа из fileA

if (num > 0) { // Проверяем, что число положительное

fprintf(fileB, "%d\t", num); // Записываем положительное число в fileB

}

}

// Закрываем файлы fileА и fileВ

fclose(fileA);

fclose(fileB);

// Повторное открытие файлов fileА и fileВ для чтения

errno\_t A2 = fopen\_s(&fileA, "fileA.txt", "r");

errno\_t B2 = fopen\_s(&fileB, "fileB.txt", "r");

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

printf("Ошибка при открытии файлов fileA.txt или fileB.txt\n");

return 1;

}

int number;

// Выводим содержимое файла fileА на экран

printf("Содержимое файла fileA.txt:\n");

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

printf("%d\t", number);

}

printf("\n");

// Выводим содержимое файла fileВ на экран

printf("Содержимое файла fileB.txt (только положительные числа):\n");

while (fscanf\_s(fileB, "%d", &number) == 1) {

printf("%d\t", number);

}

// Закрываем файлы fileF1 и fileF2

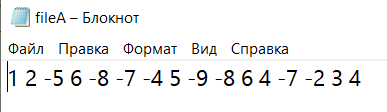
fclose(fileA);

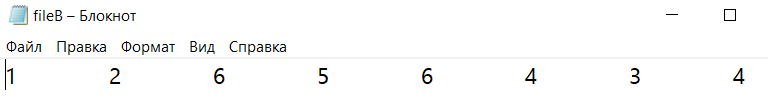
fclose(fileB);

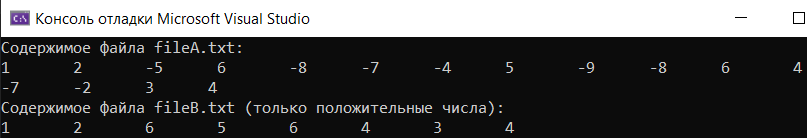
return 0;

}

Консоль VS:







2. Компоненты файла **f** – целые двухзначные числа. Получить файл **g**, образованный из **f** включением только чисел больших некоторого числа, вводимого с клавиатуры.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Директивы препроцессора в языке С

#include <stdio.h>

#include <stdlib.h>

int main() {

setlocale(LC\_ALL, "");

FILE\* fileF; // Объявляем указатель на тип FILE

// Открываем файл fileF для чтения

errno\_t F1 = fopen\_s(&fileF, "f.txt", "r");

if (fileF == NULL) {

perror("Не удалось открыть файл f.txt\n");

return 1;

}

int threshold;

printf("Введите пороговое значение: ");

scanf\_s("%d", &threshold);

FILE\* fileG; // Объявляем указатель на тип FILE

// Открываем файл fileG для чтения

errno\_t G1 = fopen\_s(&fileG, "g.txt", "w");

if (fileG == NULL) {

perror("Не удалось открыть файл g.txt\n");

fclose(fileF);

return 1;

}

int number;

while (fscanf\_s(fileF, "%d", &number) == 1) { // Если значение больше заданного порога threshold, то записываем его в файл fileG через табуляцию

if (number > threshold) {

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

}

}

// Закрываем файлы fileF и fileG

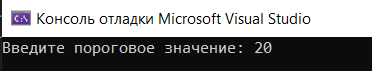
fclose(fileF);

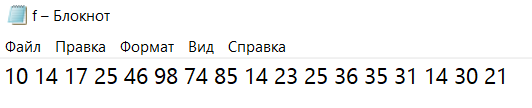
fclose(fileG);

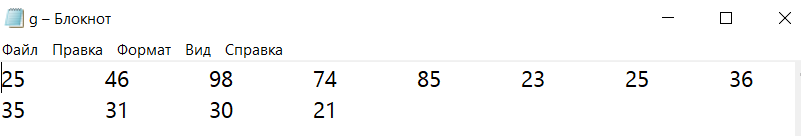
return 0;

}

Консоль VS:

****

****

****

**Вариант№ 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.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Директивы препроцессора в языке С

#include <stdio.h>

#include <stdlib.h>

#include <math.h> // Заголовочный файл позволяет использовать математические функции в программе на языке С

int main() {

setlocale(LC\_ALL, "");

FILE\* filefA; // Объявляем указатель на тип FILE

// Открываем файл fA для чтения

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

if (filefA == NULL) {

perror("Не удалось открыть файл fA.txt\n");

return 1;

}

int size = 0;

float currentValue, closestValue, userValue;

//Вводим число с клавиатуры

printf("Введите целое число: ");

if (scanf\_s("%f", &userValue) != 1) {

printf("Ошибка ввода. Введите целое число.\n");

return 1;

}

int position = 0, closestPosition;

float smallestDifference = INFINITY;

// Ищем ближайшее значение к userValue в файле filefA

while (fscanf\_s(filefA, "%F", &currentValue) == 1) {

if (abs(userValue - currentValue) < smallestDifference) {

closestValue = currentValue;

smallestDifference = abs(userValue - currentValue);

closestPosition = position;

}

position++;

}

// Закрываем файл filefA

fclose(filefA);

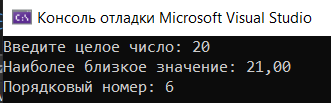
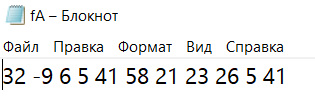
printf("Наиболее близкое значение: %.2f\n", closestValue);

printf("Порядковый номер: %d\n", closestPosition);

return 0;

}

Консоль VS:



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**.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Директивы препроцессора в языке С

#include <stdio.h>

#include <stdlib.h>

int main() {

setlocale(LC\_ALL, "");

FILE\* fileF1; // Объявляем указатель на тип FILE

// Открываем файл fileF1 для чтения

errno\_t F1 = fopen\_s(&fileF1, "F1.txt", "r");

if (fileF1 == NULL) {

perror("Не удалось открыть файл F1.txt\n");

return 1;

}

FILE\* fileF2; // Объявляем указатель на тип FILE

// Открываем файл fileF2 для чтения

errno\_t F2 = fopen\_s(&fileF2, "F2.txt", "w");

if (fileF2 == NULL) {

perror("Не удалось открыть файл F2.txt\n");

fclose(fileF1);

return 1;

}

char buffer[1000];

int isEven = 0;

while (fgets(buffer, sizeof(buffer), fileF1)) {

if (isEven) {

fprintf(fileF2, "%s", buffer);

}

isEven = !isEven;

}

// Закрываем файлы fileF1 и fileF2

fclose(fileF1);

fclose(fileF2);

// Повторное открытие файлов fileF1 и fileF2 для чтения

errno\_t F1\_1 = fopen\_s(&fileF1, "F1.txt", "r");

errno\_t F2\_1 = fopen\_s(&fileF2, "F2.txt", "r");

if (fileF1 == NULL || fileF2 == NULL) {

printf("Ошибка при открытии файлов F1.txt или F2.txt\n");

return 1;

}

// Выводим содержимое файла fileF2 на экран

printf("Содержимое файла F1.txt:\n");

while (fgets(buffer, sizeof(buffer), fileF1)) {

printf("%s", buffer);

}

printf("\n");

// Выводим содержимое файла fileF2 на экран

printf("Содержимое файла F2.txt (чётные строки из F1):\n");

while (fgets(buffer, sizeof(buffer), fileF2)) {

printf("%s", buffer);

}

// Закрываем файлы fileF1 и fileF2

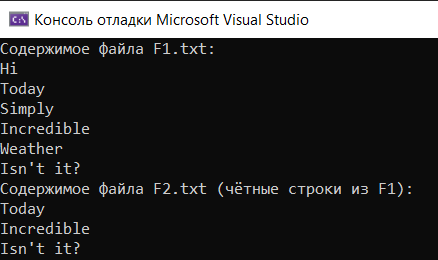
fclose(fileF1);

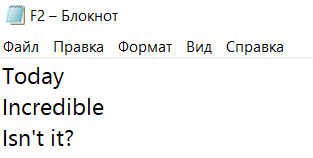
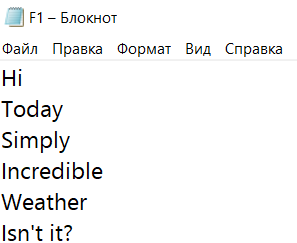
fclose(fileF2);

return 0;

}

Консоль VS:



****

**Вариант№ 11**

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

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Директивы препроцессора в языке С

#include <stdio.h>

#include <stdlib.h>

int main() {

setlocale(LC\_ALL, "");

FILE\* fileA, \* fileB; // Объявляем два указателя на тип FILE

int num;

int count[1000] = { 0 }; // Максимальное значение числа в fileA

// Открываем файл fileA для чтения

errno\_t A = fopen\_s(&fileA, "fileA.txt", "r");

if (fileA == NULL) {

printf("Не удалось открыть файл fileA.txt\n");

return 1;

}

// Считываем числа из fileA и считаем их количество

while (fscanf\_s(fileA, "%d", &num) != EOF) {

count[num]++;

}

// Закрываем файл fileA

if (fclose(fileA) != 0) {

printf("Ошибка при закрытии файла fileA.txt\n");

return 1;

}

// Открываем файл fileB для записи

errno\_t B = fopen\_s(&fileB, "fileB.txt", "w");

if (fileB == NULL) {

printf("Не удалось открыть файл fileB.txt.n");

return 1;

}

// Повторное открытие файла fileA для чтения

errno\_t A1 = fopen\_s(&fileA, "fileA.txt", "r");

if (fileA == NULL) {

printf("Не удалось повторно открыть файл fileA.txt\n");

return 1;

}

// Записываем в fileB числа, которые не повторяются в fileA

while (fscanf\_s(fileA, "%d", &num) != EOF) {

if (count[num] < 2) {

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

}

}

// Закрываем файл fileA

if (fclose(fileA) != 0) {

printf("Ошибка при закрытии файла fileA.txt\n");

return 1;

}

// Закрываем файл fileB

if (fclose(fileB) != 0) {

printf("Ошибка при закрытии файла fileB.txt\n");

return 1;

}

// Открываем файл fileB для чтения и выводим его содержимое на экран

errno\_t B1 = fopen\_s(&fileB, "fileB.txt", "r");

if (fileB == NULL) {

printf("Не удалось открыть файл fileB.txt\n");

return 1;

}

printf("Содержимое файла fileB:\n");

while (fscanf\_s(fileB, "%d", &num) != EOF) {

printf("%d\t", num);

}

// Закрываем файл fileB

if (fclose(fileB) != 0) {

printf("Ошибка при закрытии файла fileB.txt\n");

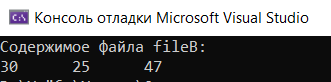
return 1;

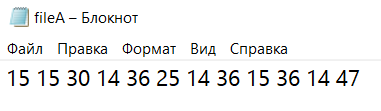
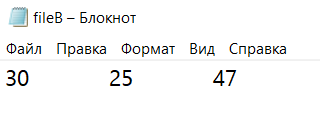
}

return 0;

}

Консоль VS:





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**».

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Директивы препроцессора в языке С

#include <stdio.h>

#include <stdlib.h>

#include <string.h> // Подключаем библиотеку для работы со строками

int main() {

setlocale(LC\_ALL, "");

FILE\* fileF1; // Объявляем указатель на тип FILE

// Открываем файл fileF1 для чтения

errno\_t F1 = fopen\_s(&fileF1, "F1.txt", "r");

if (fileF1 == NULL) {

perror("Не удалось открыть файл F1.txt\n");

return 1;

}

FILE\* fileF2; // Объявляем указатель на тип FILE

// Открываем файл fileF2 для чтения

errno\_t F2 = fopen\_s(&fileF2, "F2.txt", "w");

if (fileF2 == NULL) {

perror("Не удалось открыть файл F2.txt\n");

fclose(fileF1);

return 1;

}

char buffer[1000];

int lineCounter = 1;

while (fgets(buffer, sizeof(buffer), fileF1)) {

// Проверяем, заканчивается ли строка символом 'a' или 'A'

if (buffer[strlen(buffer) - 2] == 'a' || buffer[strlen(buffer) - 2] == 'A') {

fprintf(fileF2, "%s", buffer);

}

}

// Закрываем файлы fileF1 и fileF2

fclose(fileF1);

fclose(fileF2);

// Повторное открытие файлов fileF1 и fileF2 для чтения

errno\_t F1\_1 = fopen\_s(&fileF1, "F1.txt", "r");

errno\_t F2\_1 = fopen\_s(&fileF2, "F2.txt", "r");

char ch;

// Выводим содержимое файла fileF1 на экран

printf("Содержимое файла F1.txt:\n");

ch = fgetc(fileF1);

while (ch != EOF) {

printf("%c", ch);

ch = fgetc(fileF1);

}

printf("\n");

// Выводим содержимое файла fileF2 на экран

printf("Содержимое файла F2.txt (строки, заканчивающиеся на 'a' или 'A'):\n");

ch = fgetc(fileF2);

while (ch != EOF) {

printf("%c", ch);

ch = fgetc(fileF2);

}

// Закрываем файлы fileF1 и fileF2

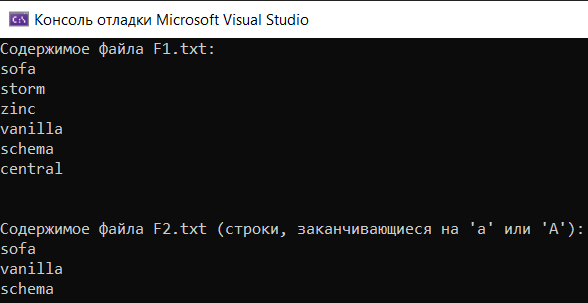
fclose(fileF1);

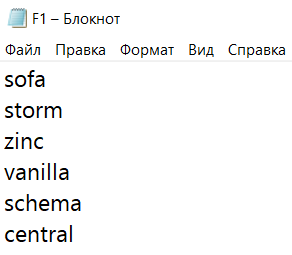
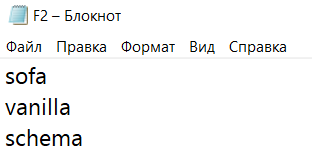
fclose(fileF2);

return 0;

}

Консоль VS:

****

****