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:

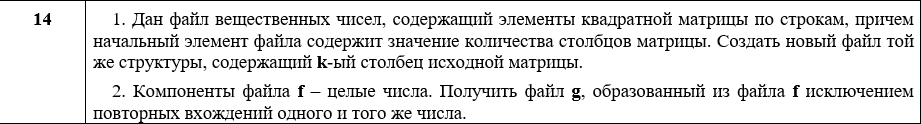
Студент 1 курса 10 группы

Сегренёв Кирилл Сергеевич

Преподаватель: асс. Андронова М.В.

2023, Минск

**ВАРИАНТ 14**



**Задание 1**

**Код:**

#include <stdio.h>

#include <iostream>

int main() {

setlocale(LC\_ALL, "ru");

FILE\* inputFile, \* outputFile;

int n, i, j, k;

printf("Введите k-столбец: ");

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

if (fopen\_s(&inputFile, "D:\\labs\\ОАП\\18lab\\ex1.txt", "r") != 0) { // Открываем файл, если файл не открылся(вернул не 0), то вывести на экран ошибку

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

fscanf\_s(inputFile, "%d", &n); // Т.к первый элемент в файле у нас количество столбцов квадратной матрицы, то считываем его и применяем в своих циклах

float\*\* arr = new float\* [n]; // Создаем динамический массив для записи значений из файла

int number = n;

for (int i = 0; i < number; i++)

arr[i] = new float[number];

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

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

fscanf\_s(inputFile, "%f", &arr[i][j]); // С помощью цикла и функции записываем в массив из файла значения матрицы

}

}

fclose(inputFile); // Закрываем файл, их которого брали значения

if (fopen\_s(&outputFile, "D:\\labs\\ОАП\\18lab\\ex2.txt", "w") != 0) { // Открываем файл в который будем записывать наш столбец

perror("Не удалось открыть файл для записи");

exit(EXIT\_FAILURE);

}

fprintf(outputFile, "%d\n", 1); // Записываем в первую строчку 1, т.к по структуре у нас останется 1 столбик

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

fprintf(outputFile, "%.2f\n", arr[i][k - 1]); // С помощью цикла записываем в новый файл значения матрицы

}

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

delete[] arr[i];

}

delete[] arr;

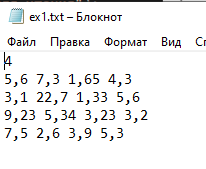
fclose(outputFile); // Закрываем файл, в который записали наши значения

printf("Столбец успешно извлечен!"); // После всех действий выводим сообщение в консоль

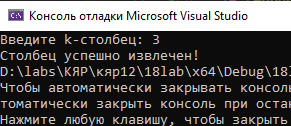
return 0;

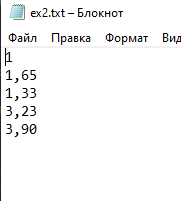
}

**Input:**

****

**Output:**

****

****

**Задание 2**

**Код:**

#include <iostream>

#include <stdio.h>

#include <algorithm>

int main() {

setlocale(LC\_ALL, "ru");

int size = 0;

FILE\* inputFile, \* outputFile;

long fsize;

if (fopen\_s(&inputFile, "D:\\labs\\ОАП\\18lab\\intNumbers.txt", "r") != 0) { // Открываем файл для чтения и смотрим, что он вернет(если 0, то он открылся)

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

fseek(inputFile, 0L, SEEK\_END); // Считаем размер файла для динамического массива

fsize = ftell(inputFile);

fseek(inputFile, 0L, SEEK\_SET);

int\* arr = new int[fsize];

while (fscanf\_s(inputFile, "%d", &arr[size]) == 1) { // Чтобы не заходить за пределы памяти считаем, сколько целых элементов будет в нашей последовательности

size++;

}

fclose(inputFile);

fopen\_s(&outputFile, "D:\\labs\\ОАП\\18lab\\Numbers.txt", "w"); // Открываем второй файл для записи

std::sort(arr, arr + size); // Сортируем массив для дальнейшего сравнения

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

if (arr[i] != arr[i - 1]) {

fprintf(outputFile, "%d ", arr[i]); // Если символы не совпадают, то записать их в новый файл

}

}

fclose(outputFile);

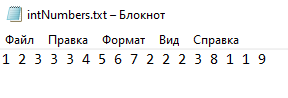
printf("Повторяющиеся числа удалены!\n"); // После всех действий выводим сообщение в консоль

delete[] arr;

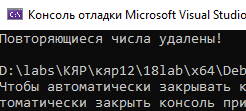
return 0;

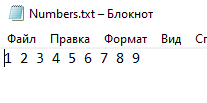
}

**Input:**



**Output:**

****

****

**ДОПОЛНИТЕЛЬНЫЕ ЗАДАНИЯ**

**ВАРИАНТ 4**

**** ****

**Задание 1**

**Код:**

#include <iostream>

#include <stdio.h>

//4

int main() {

FILE\* fileA, \* fileB;

int number;

// Открываем два файла(один для чтения, другой для записи)

if (fopen\_s(&fileA, "D:\\labs\\ОАП\\18lab\\fileA.txt", "r") != 0) {

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

if (fopen\_s(&fileB, "D:\\labs\\ОАП\\18lab\\fileB.txt", "w") != 0) {

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

for (int i = 0; !feof(fileA); i++) { // С помощью функции feof организовываем цикл. Данный цикл продолжается до того момента, пока не встретит значение EOF(конец данных)

fscanf\_s(fileA, "%d", &number); // Считываем каждую цифру в fileA

if (number > 0) // И если эта цифра положительная, то записать ее в новыый файл

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

}

// Закрываем оба файла

fclose(fileB);

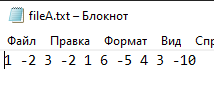
fclose(fileA);

printf("В новом файле остались только положителбные элементы!\n"); // После всех действий выводим сообщение в консоль

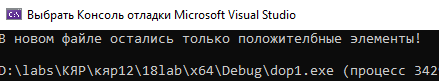
return 0;

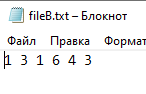
}

**Input:**



**Output:**





**Задание 2**

**Код:**

#include <iostream>

#include <stdio.h>

int main() {

setlocale(LC\_ALL, "ru");

FILE\* f, \* g;

int number, k;

printf("Введите число: ");

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

// Открываем два файла, из одного будем брать значения, в другой записывать

if (fopen\_s(&f, "D:\\labs\\ОАП\\18lab\\f.txt", "r") != 0) {

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

if (fopen\_s(&g, "D:\\labs\\ОАП\\18lab\\g.txt", "w") != 0) {

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

for (int i = 0; !feof(f); i++) {

fscanf\_s(f, "%d", &number); // Сначала из первого файла считываем число

if (number > k) // Если оно больше введеного числа, то записать во второй файл

fprintf(g, "%d ", number);

}

// Закрываем файлы

fclose(f);

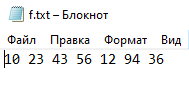
fclose(g);

printf("Числа меньше введенного извлечены!\n"); // После всех действий выводим сообщение в консоль

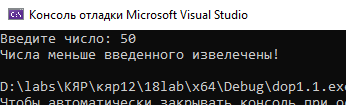
return 0;

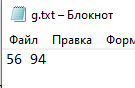
}

**Input:**

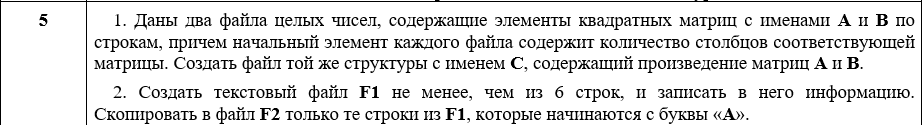


**Output:**





**ВАРИАНТ 5**



**Задание 1**

**Код:**

#include <iostream>

#include <stdio.h>

//5

int main() {

setlocale(LC\_ALL, "ru");

FILE\* A, \* B, \* C;

int n, j;

//Открываем два файла для чтения

if (fopen\_s(&A, "D:\\labs\\ОАП\\18lab\\A.txt", "r") != 0) {

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

if (fopen\_s(&B, "D:\\labs\\ОАП\\18lab\\B.txt", "r") != 0) {

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

fscanf\_s(A, "%d", &n); // Считываем первое целое число(количество столбцов квадратной матрицы)

int\*\* arr1 = new int\* [n];

for (int i = 0; i < n; i++) // Т.к у нас квадратная матрица, используем значение n как для столбцов, так и для строк

arr1[i] = new int[n];

fscanf\_s(B, "%d", &n); // Считываем первое целое число(количество столбцов квадратной матрицы)

int\*\* arr2 = new int\* [n];

for (int i = 0; i < n; i++)

arr2[i] = new int[n];

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

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

fscanf\_s(A, "%d", &arr1[i][j]); // Записываем в первый массив(относится к файлу A.txt) каждый элемент матрицы

fscanf\_s(B, "%d", &arr2[i][j]); // Записываем во второй массив(относится к файлу B.txt) каждый элемент матрицы

}

}

//Закрываем файлы

fclose(A);

fclose(B);

if (fopen\_s(&C, "D:\\labs\\ОАП\\18lab\\С.txt", "w") != 0) { // Открываем файл для записи

perror("Не удалось открыть файл для записи");

exit(EXIT\_FAILURE);

}

fprintf(C, "%d\n", n); // Записываем число колонок матрицы в первую строку

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

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

fprintf(C, "%d ", arr1[i][j] \* arr2[i][j]); // Записываем произведение каждого элемента предыдущих матриц

}

fprintf(C, "\n");

}

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

delete[] arr1[i];

delete[] arr2[i];

}

delete[] arr1;

delete[] arr2;

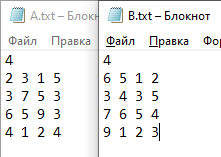
fclose(C); // Закрываем 3 файл

printf("Матрица изменена!\n"); // После всех действий выводим сообщение в консоль

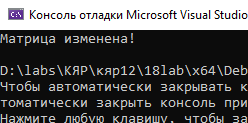
return 0;

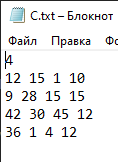
}

**Input:**

****

**Output:**

****

****

**Задание 2**

**Код:**

#include <iostream>

#include <stdio.h>

int main() {

setlocale(LC\_ALL, "ru");

FILE\* F1, \* F2;

char c[300];// Массив используем для записи предложений

if (fopen\_s(&F1, "D:\\labs\\ОАП\\18lab\\dop2.2\\F1.txt", "r") != 0) { // Открываем F1 для чтения

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

if (fopen\_s(&F2, "D:\\labs\\ОАП\\18lab\\dop2.2\\F2.txt", "w") != 0) {// Открываем F2 для записи

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

while(!feof(F1)) // Выполнять цикл, пока данные в файле F1 не закончатся

{

fgets(c, 300, F1); // Из файла F1 берем строки с помощью массива

if(c[0] == 'A') // Если первый элемент массива, следовательно и строки, равен символу A, то записать данную строку в файл F2

fprintf(F2, c);

}

// Закрываем файлы

fclose(F2);

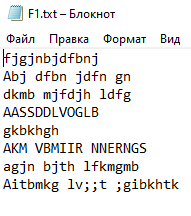
fclose(F1);

printf("Строки успешно извлечены!"); // После всех действий выводим сообщение в консоль

return 0;

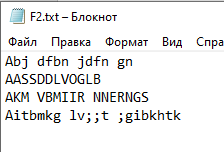
}

**Input:**

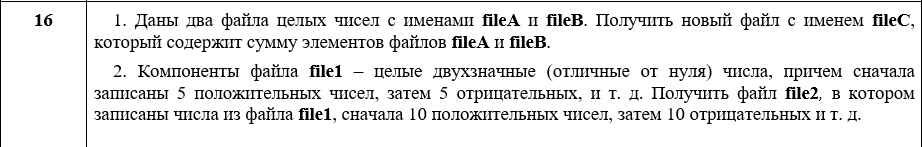


**Output:**

****

****

**ВАРИАНТ 16**



**Задание 1**

**Код:**

#include <iostream>

#include <stdio.h>

int main() {

setlocale(LC\_ALL, "ru");

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

int numberA, numberB;

// Открываем два файла(два для чтения, третий для записи)

if (fopen\_s(&fileA, "D:\\labs\\ОАП\\18lab\\dop3\\fileA.txt", "r") != 0) {

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

if (fopen\_s(&fileB, "D:\\labs\\ОАП\\18lab\\dop3\\fileB.txt", "r") != 0) {

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

if (fopen\_s(&fileC, "D:\\labs\\ОАП\\18lab\\dop3\\fileC.txt", "w") != 0) {

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

while (!feof(fileA)) { // Цикл продолжается, пока данные не закончаться

// Добавляем пробелы и при считывании данных(для того, чтобы последнее число не дублировалось)

fscanf\_s(fileA, "%d ", &numberA); // Считываем первую цифру из fileA и записываем значение в переменную numberA

fscanf\_s(fileB, "%d ", &numberB); // Считываем первую цифру из fileB и записываем значение в переменную numberB

fprintf(fileC, "%d ", numberA + numberB); // Записываем сумму данных переменных в новый файл

}

// Закрываем файлы

fclose(fileA);

fclose(fileB);

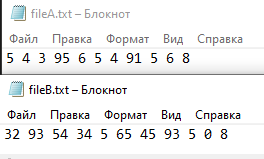
fclose(fileC);

printf("Сумма элементов из fileA и fileB записана в файле C!\n");

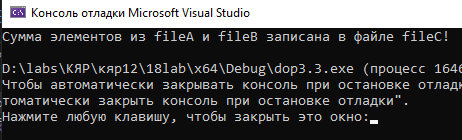
return 0;

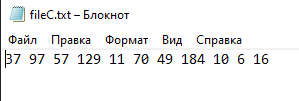
}

**Input:**



**Output:**





**Задание 2**

**Код:**

#include <iostream>

#include <stdio.h>

int main() {

setlocale(LC\_ALL, "ru");

FILE\* file1, \* file2;

int number, scan = 0, i = 0, g = 0;

long fsize = 0L; // Переменная адреса, которую мы будем изменять для каждого блока чисел(10 положительных 10 отрицательных)

// Открываем два файла(один для чтения, другой для записи)

if (fopen\_s(&file1, "D:\\labs\\ОАП\\18lab\\dop3.2\\file1.txt", "r") != 0) {

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

if (fopen\_s(&file2, "D:\\labs\\ОАП\\18lab\\dop3.2\\file2.txt", "w") != 0) {

perror("Не удалось открыть файл для чтения");

exit(EXIT\_FAILURE);

}

fseek(file1, 0L, SEEK\_END);

long fsizeFile = ftell(file1); // Считываем длину файла в байтах для реализации цикла

fseek(file1, 0L, SEEK\_SET);

for (int i = 0; fsize < fsizeFile; i++) {

while (g < 20) { // Т.к в первых 20 элементах у нас 10 положительных и 10 отрицательных, проходим по 20 числам и выбираем только положительные

fscanf\_s(file1, "%d ", &number);

if (number > 0) {

fprintf(file2, "%d ", number);

}

g++;

}

g = 0; // Обнуляем иттератор

fseek(file1, fsize, SEEK\_SET); // Возвращаем позицию в файле на ту, с которой проходила проверка положительных элементов

while (g < 20) { // Теперь проверяем 20 чисел и выбираем из них отрицательные

fscanf\_s(file1, "%d ", &number);

if (number < 0) {

fprintf(file2, "%d ", number);

}

g++;

}

g = 0;

fsize += ftell(file1); // Вычисляем место, где закончил проверку прошлый цикл

fseek(file1, fsize, SEEK\_SET); // Ставим туда указатель

}

fclose(file1);

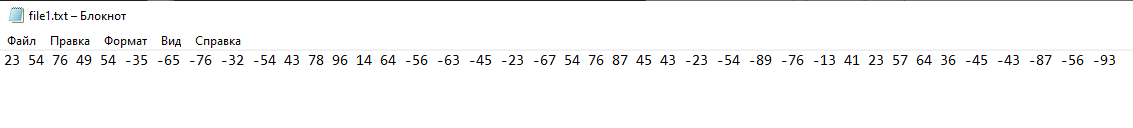
fclose(file2);

printf("Числа успежно переставлены!\n");

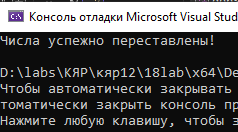
return 0;

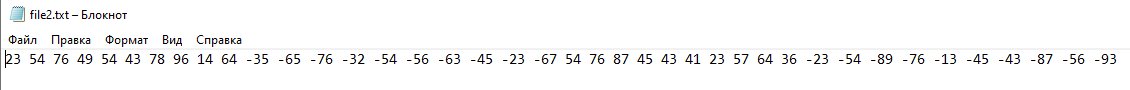
}

**Input:**

****

**Output:**

****

****