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»

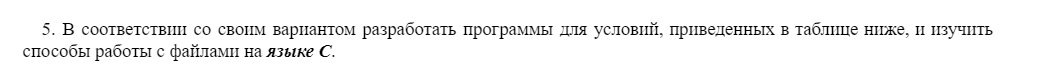
 Выполнила:

Студентка1 курса 6 группы

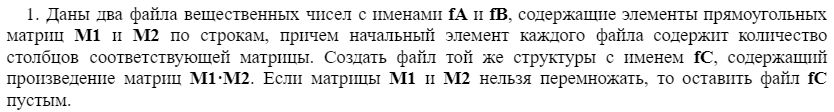
Альшевская Алина Михайловна

Преподаватель: асс. Андронова М.В.

2023, Минск



Вариант 1



#include <iostream>

#include <fstream>

#include <vector>

#include <Windows.h>

using namespace std;

void multiplyMatrices(const char\* fileA, const char\* fileB, const char\* fileC) {

// Открываем файлы для чтения

FILE\* fdA, \* fdB;

if (fopen\_s(&fdA, fileA, "r") != 0 || fopen\_s(&fdB, fileB, "r") != 0) {

cout << "Ошибка открытия файлов." << endl;

return;

}

// Считываем количество столбцов в каждой матрице

int colsA, colsB;

if (fscanf\_s(fdA, "%d", &colsA) == EOF || fscanf\_s(fdB, "%d", &colsB) == EOF) {

cout << "Ошибка чтения количества столбцов." << endl;

fclose(fdA);

fclose(fdB);

return;

}

// Создаем вектор для матрицы A

vector<vector<double>> matrixA;

// Считываем матрицу A из файла

double elementA;

while (fscanf\_s(fdA, "%lf", &elementA) != EOF) {

vector<double> row;

row.push\_back(elementA); // ДОбавляем первый элемент в вектор строки матрицы. поскольку мы не знаем сколько столбцов, то используем такой цикл, который, дойдя до конца файла вернет EOF

for (int i = 1; i < colsA; i++) {

if (fscanf\_s(fdA, "%lf", &elementA) == EOF) {

cout << "Ошибка чтения матрицы A." << endl;

fclose(fdA);

fclose(fdB);

return;

}

row.push\_back(elementA);

}

matrixA.push\_back(row);

}

// Закрываем файл с матрицей A

fclose(fdA);

// Создаем вектор для матрицы B

vector<vector<double>> matrixB;

// Считываем матрицу B из файла

double elementB;

while (fscanf\_s(fdB, "%lf", &elementB) != EOF) {

vector<double> row;

row.push\_back(elementB);

for (int i = 1; i < colsB; i++) {

if (fscanf\_s(fdB, "%lf", &elementB) == EOF) {

cout << "Ошибка чтения матрицы B." << endl;

fclose(fdB);

return;

}

row.push\_back(elementB);

}

matrixB.push\_back(row);

}

// Закрываем файл с матрицей B

fclose(fdB);

// Проверяем, можно ли перемножить матрицы

if (colsA != matrixB.size()) {

cout << "Невозможно перемножить матрицы. Количество столбцов первой матрицы не равно количеству строк второй матрицы." << endl;

return;

}

// Открываем файл для записи

FILE\* fdC;

if (fopen\_s(&fdC, fileC, "w") != 0) {

cout << "Ошибка открытия файла для записи." << endl;

return;

}

// Выполняем умножение матриц. .size() используется для получения количества элементов в векторе

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

for (int j = 0; j < matrixB[0].size(); j++) {

double sum = 0.0;

for (int k = 0; k < matrixB.size(); k++) {

sum += matrixA[i][k] \* matrixB[k][j];

}

fprintf\_s(fdC, "%.2f ", sum);

}

fprintf\_s(fdC, "\n");

}

// Закрываем файл с результатами

fclose(fdC);

cout << "Успешно создан файл " << fileC << " с результатом перемножения матриц." << endl;

}

int main() {

SetConsoleOutputCP(1251);

const char\* fileA = "fA.txt";

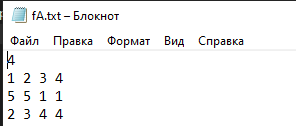
const char\* fileB = "fB.txt";

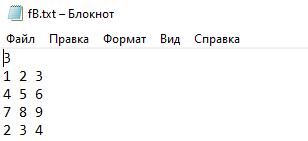
const char\* fileC = "fC.txt";

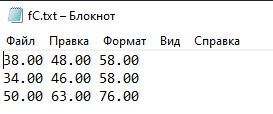
multiplyMatrices(fileA, fileB, fileC);

return 0;

}









#include <iostream>

#include <fstream>

#include <Windows.h>

#include <vector>

using namespace std;

void cute(const char\* fileF, const char\* fileG, const int K)

{

// Открываем файл для чтения

FILE\* filef;

if (fopen\_s(&filef, fileF, "r") != 0) {

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

return;

}

// Создаем вектор для хранения чисел

vector<int> numbers;

int num;

// Считываем числа из файла и сохраняем их в вектор

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

numbers.push\_back(num);

}

// Закрываем файл для чтения

fclose(filef);

// Открываем файл для записи

FILE\* fileg;

if (fopen\_s(&fileg, fileG, "w") != 0) {

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

return;

}

// Записываем в файл числа из вектора, которые кратны числу K

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

if (numbers[i] % K == 0) {

fprintf\_s(fileg, "%d ", numbers[i]);

}

}

// Закрываем файл для записи

fclose(fileg);

}

int main() {

// Устанавливаем кодировку для корректного вывода на консоль (CP1251)

SetConsoleOutputCP(1251);

// Имена файлов и число K

const char\* fileF = "fileF.txt";

const char\* fileG = "fileG.txt";

int k = 43;

// Вызываем функцию cute с заданными параметрами

cute(fileF, fileG, k);

return 0;

}

