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 C»

Выполнила:

Студент(ка) 1 курса 7 группы

Подшиваленко Диана Игоревна

Преподаватель: асс. Андронова М.В.

2023, Минск

|  |  |
| --- | --- |
| **11** | 1. Компоненты файла **fileA** –  целые числа, значения которых повторяются.  Получить файл **fileB**, образованный из **fileA** числами, которые не повторяются.  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**». |

#include <stdio.h>

#include <cstdlib>

#include <iostream>

#include <Windows.h>

int main() {

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

FILE\* f, \* g;

int a[1000];

bool t = false;

errno\_t err, err2;

err = fopen\_s(&f, "fileA.txt", "r"); // открываем первый файл с именем "fileA.txt" для чтения

err2 = fopen\_s(&g, "fileB.txt", "w"); // открываем второй файл с именем "fileB.txt" для записи

if (err != 0) // делаем проверку открытия первого файла

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

if (err2 != 0) // делаем проверку открытия второго файла

{

perror("Невозможно создать файл\n");

return EXIT\_FAILURE;

}

int i = 0;

while (!feof(f)) // пока не конец файла

{

fscanf\_s(f, "%d", &a[i]); // считываем числа и записываем в массив чисел

i++;

}

for (int j = 0; j < i - 1; j++) {

for (int h = j + 1; h < i; h++) {

if (a[j] == a[h]) {

t = true;

for (int k = h; k < i; k++) {

a[k] = a[k + 1];

}

i--;

h--;

}

}

if (t) {

for (int k = j; k < i; k++) {

a[k] = a[k + 1];

}

i--;

j--;

}

t = false;

}

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

fprintf(g, "%d ", a[j]); // записываем неповторяющиеся числа во второй файла

}

fclose(f); // закрываем первый файл

fclose(g); // закрываем второй файл

}

#include <stdio.h>

#include <conio.h>

#include <cstdlib>

#include <iostream>

int main() {

setlocale(LC\_CTYPE, "Russian");

FILE\* f, \* g;

char a[500];

int n;

errno\_t err, err2;

err2 = fopen\_s(&g, "F2.txt", "w"); // открываем второй файл с именем "fileB.txt" для записи

if (err2 != 0) // делаем проверку открытия второго файла

{

perror("Невозможно создать файл\n");

return EXIT\_FAILURE;

}

err = fopen\_s(&f, "F1.txt", "r"); // открываем первый файл для чтения

if (err != 0) // делаем проверку открытия первого файла

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

while (fgets(a, 500, f)) // пока считывание происходит нормально и мы не дошли до конца

{

if (a[strlen(a) - 1] == '\n') {

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

fprintf(g, "%s", a);

}

}

else if (a[strlen(a) - 1] == 'a') {

fprintf(g, "%s", a);

}

}

fclose(f); // закрываем первый файл

fclose(g); // закрываем второй файл

return 0;

}

|  |  |
| --- | --- |
| **1** | 1. Даны два файла вещественных чисел с именами **fA** и **fB**, содержащие элементы прямоугольных матриц **M1** и **M2** по строкам, причем начальный элемент каждого файла содержит количество столбцов соответствующей матрицы. Создать файл той же структуры с именем **fC**, содержащий произведение матриц **M1·M2**. Если матрицы **M1** и **M2** нельзя перемножать, то оставить файл **fC** пустым.  2. Компоненты файла **f** – целые двухзначные числа (положительные и отрицательные). Получить файл **g**, образованный из **f** включением только чисел кратных **K**. |

#include <stdio.h>

#include <cstdlib>

#include <iostream>

int main() {

setlocale(LC\_CTYPE, "Russian");

FILE \*f, \*g, \*d;

errno\_t err, err2, err3;

err = fopen\_s(&f, "fA.txt", "r"); // открываем первый файл с именем "fA.txt" для чтения

err2 = fopen\_s(&g, "fB.txt", "r"); // открываем второй файл с именем "fB.txt" для чтения

err3 = fopen\_s(&d, "fC.txt", "w"); // открываем второй файл с именем "fC.txt" для записи

if (err != 0) // делаем проверку открытия первого файла

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

if (err2 != 0) // делаем проверку открытия второго файла

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

if (err3 != 0) // делаем проверку открытия третьего файла

{

perror("Невозможно создать файл\n");

return EXIT\_FAILURE;

}

int n = 1, m, n1 = 1, m1; // создаём переменные для размера первой матрицы и второй матрицы

int \*\*a = (int\*\*) malloc(1 \* sizeof(int\*)); // выделяем память под массив массивов

fscanf\_s(f, "%d", &m); // считываем количество столбцов первой матрицы

a[0] = (int\*) malloc(m \* sizeof(int)); // выделяем память под массив

int k = 0;

int i = 0;

while (fscanf\_s(f, "%d", &a[i][k]) != EOF) { // пока не достигли конца файла читаем число и записываем в матрицу а

k++;

if (k == m) {

i++;

n++;

a = (int\*\*)realloc(a, n \* sizeof(int\*)); // расширяем массив массивов

a[n - 1] = (int\*)malloc(m \* sizeof(int)); // выделяем память под новый массив

k = 0;

}

}

n--;

int\*\* b = (int\*\*)malloc(1 \* sizeof(int\*)); // выделяем память под массив массивов

fscanf\_s(g, "%d", &m1); // считываем количество столбцов во второй матрице

b[0] = (int\*)malloc(m1 \* sizeof(int)); // выделяем память под новый массив

k = 0;

i = 0;

while (fscanf\_s(g, "%d", &b[i][k]) != EOF) { // пока не достигли конца файла читаем число и записываем в матрицу b

k++;

if (k == m1) {

i++;

n1++;

b = (int\*\*)realloc(b, n1 \* sizeof(int\*)); // расширяем массив массивов

b[n1 - 1] = (int\*)malloc(m1 \* sizeof(int)); // выделяем память под новый массив

k = 0;

}

}

n1--;

fclose(f); // закрываем первый файл

fclose(g); // закрываем второй файл

int s;

if (m == n1) { // если количество столбцов первой матрицы = количеству строк второй матрицы, то их можно перемножать

for (int i = 0; i < n; i++) { // поэлементно находим элементы новой матрицы порядка(n x m1)

for (int k = 0; k < m1; k++) {

s = 0;

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

s = s + (a[i][j] \* b[j][k]);

}

fprintf(d, "%d ", s); // записываем полученный элемент перемножения в файл

}

fprintf(d, "\n"); // делаем переход на новую строку

}

}

else { // выводим сообщение, что перемножение матриц невозможно

printf("Матрицы нельзя перемножать ");

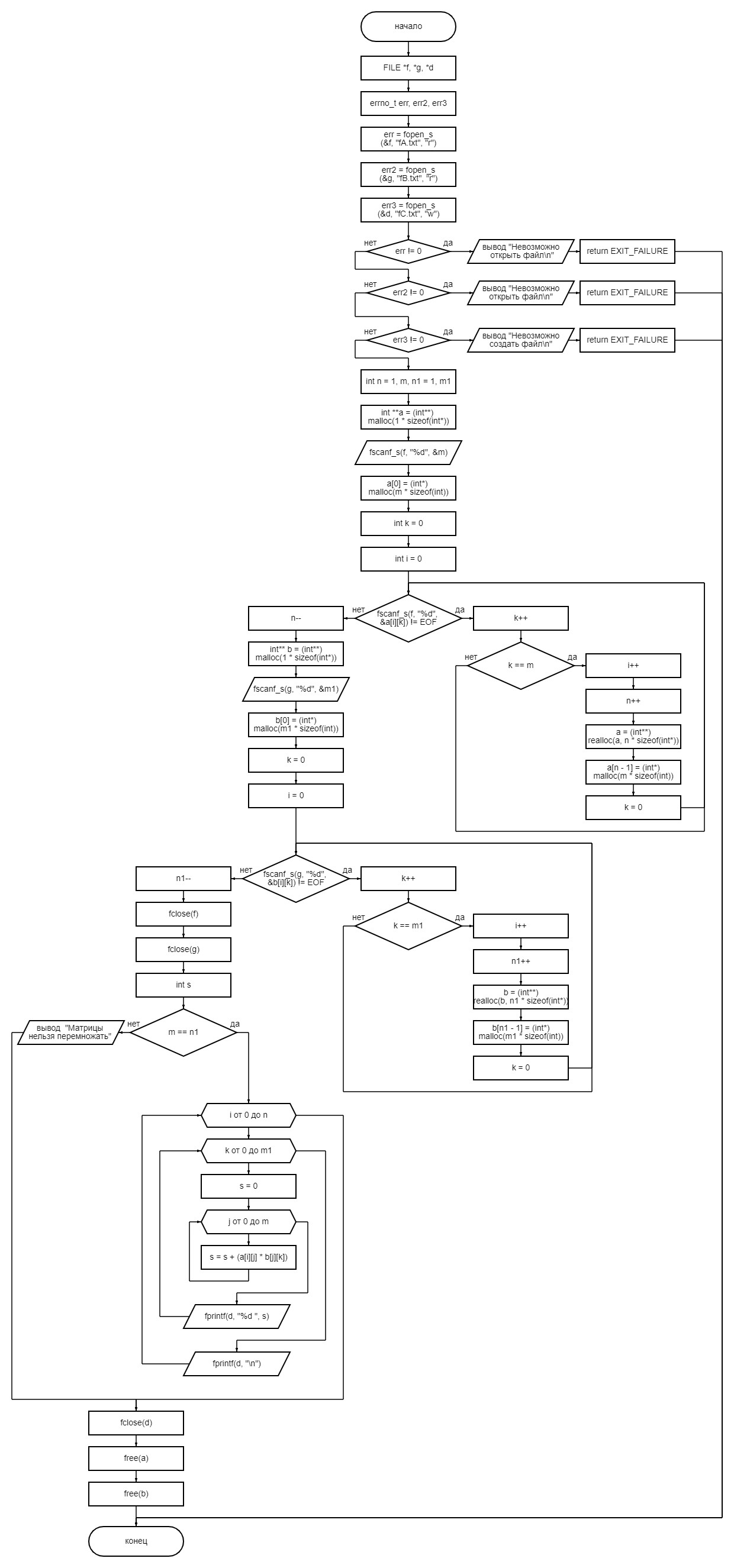
}

fclose(d); // закрываем третий файл

free(a); // освобождаем память

free(b); // освобождаем память

}



#include <stdio.h>

#include <cstdlib>

#include <iostream>

int main() {

setlocale(LC\_CTYPE, "Russian");

FILE \*f, \*g;

errno\_t err, err2;

err = fopen\_s(&f, "f.txt", "r"); // открываем первый файл с именем "f.txt" для чтения

err2 = fopen\_s(&g, "g.txt", "w"); // открываем второй файл с именем "g.txt" для записи

if (err != 0) // делаем проверку открытия первого файла

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

if (err2 != 0) // делаем проверку открытия второго файла

{

perror("Невозможно создать файл\n");

return EXIT\_FAILURE;

}

printf("Введите число, для которого будем искать кратные числа ");

int k;

scanf\_s("%d", &k); // считываем число, которому будем искать кратные

int a;

while (fscanf\_s(f,"%d", &a) != EOF) { // пока не достигли конца файла считываем числа

if (a % k == 0) { // если кратно, то

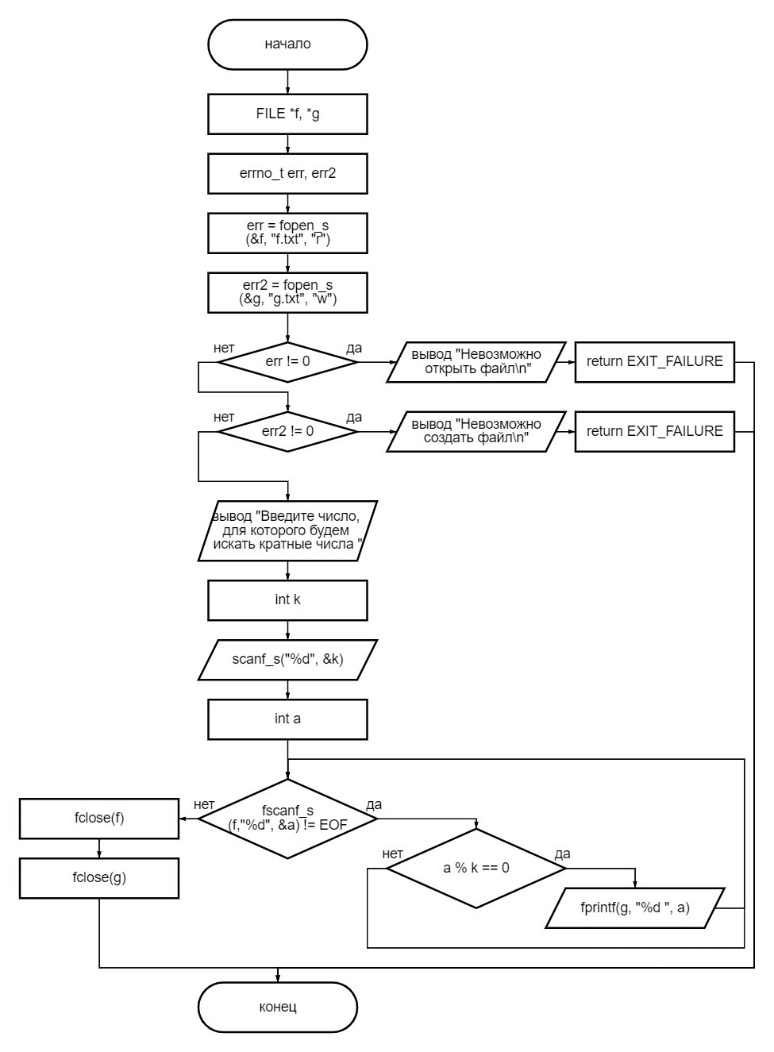
fprintf(g, "%d ", a); // записываем в файл

}

}

fclose(f); // закрываем файл

fclose(g); // закрываем файл

}

|  |  |
| --- | --- |
| **3** | 1. Дан файл вещественных чисел, содержащий элементы квадратной матрицы (по строкам), причем начальный элемент файла содержит значение количества столбцов матрицы. Создать новый файл той же структуры, содержащий транспонированную матрицу.  2. Даны три файла целых чисел одинакового размера с именами **NameA**, **NameB** и **NameC**. Создать новый файл с именем **NameD**, в котором чередовались бы элементы исходных файлов с одним и тем же номером: **A0, B0, C0, A1, B1, C1, A2, B2, C2, ...** |

#include <stdio.h>

#include <cstdlib>

#include <iostream>

int main() {

setlocale(LC\_CTYPE, "Russian");

FILE \*f, \*g;

errno\_t err, err2;

err = fopen\_s(&f, "f.txt", "r"); // открываем первый файл с именем "f.txt" для чтения

err2 = fopen\_s(&g, "g.txt", "w"); // открываем второй файл с именем "g.txt" для записи

if (err != 0) // делаем проверку открытия первого файла

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

if (err2 != 0) // делаем проверку открытия второго файла

{

perror("Невозможно создать файл\n");

return EXIT\_FAILURE;

}

int k;

fscanf\_s(f, "%d", &k); // считываем размер матрицы

float\*\* a = (float\*\*)malloc(k \* sizeof(float\*)); // выделяем память под массив массивов

a[0] = (float\*)malloc(k \* sizeof(float)); // выделяем память под новый массив

int j = 0;

int i = 0;

while (fscanf\_s(f, "%f ", &a[i][j]) != EOF) { // пока не достигли конца файла считываем числа

j++;

if (j == k) {

i++;

a[i] = (float\*)malloc(k \* sizeof(float)); // выделяем память под новый массив

j = 0;

}

}

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

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

fprintf(g, "%.2f ", a[j][i]); // записываем числа в файл g

}

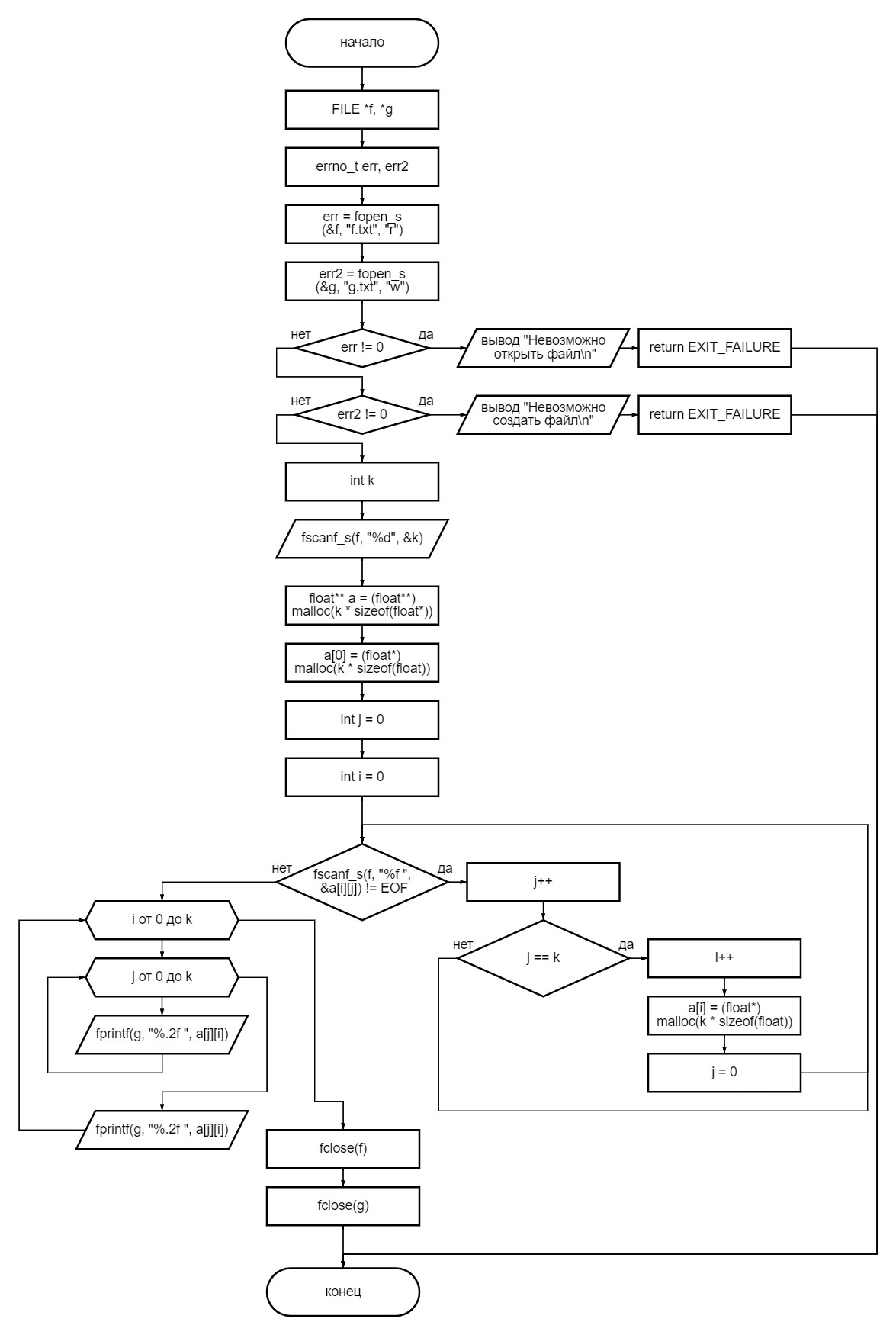
fprintf(g, "\n"); // переходим на новую строку

}

fclose(f); // закрываем файл

fclose(g); // закрываем файл

}



#include <stdio.h>

#include <cstdlib>

#include <iostream>

int main() {

setlocale(LC\_CTYPE, "Russian");

FILE \*f, \*g, \*d, \*t;

errno\_t err, err2, err3, err4;

err = fopen\_s(&f, "NameA.txt", "r"); // открываем первый файл с именем "NameA.txt" для чтения

err2 = fopen\_s(&g, "NameB.txt", "r"); // открываем второй файл с именем "NameB.txt" для чтения

err3 = fopen\_s(&d, "NameC.txt", "r"); // открываем второй файл с именем "NameC.txt" для чтения

err4 = fopen\_s(&t, "NameD.txt", "w"); // открываем второй файл с именем "NameD.txt" для записи

if (err != 0) // делаем проверку открытия первого файла

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

if (err2 != 0) // делаем проверку открытия второго файла

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

if (err3 != 0) // делаем проверку открытия третьего файла

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

if (err4 != 0) // делаем проверку открытия четвёртого файла

{

perror("Невозможно создать файл\n");

return EXIT\_FAILURE;

}

int n;

while (fscanf\_s(f, "%d", &n) != EOF) { // пока не конец файлов(мы можем делать проверку по 1, так как их размеры равны по условию)

fprintf(t, "%d ", n); // записываем число из 1 файла

fscanf\_s(g, "%d", &n); // считываем число из 2 файла

fprintf(t, "%d ", n);// записываем число из 2 файла

fscanf\_s(d, "%d", &n); // считываем число из 3 файла

fprintf(t, "%d ", n);// записываем число из 3 файла

}

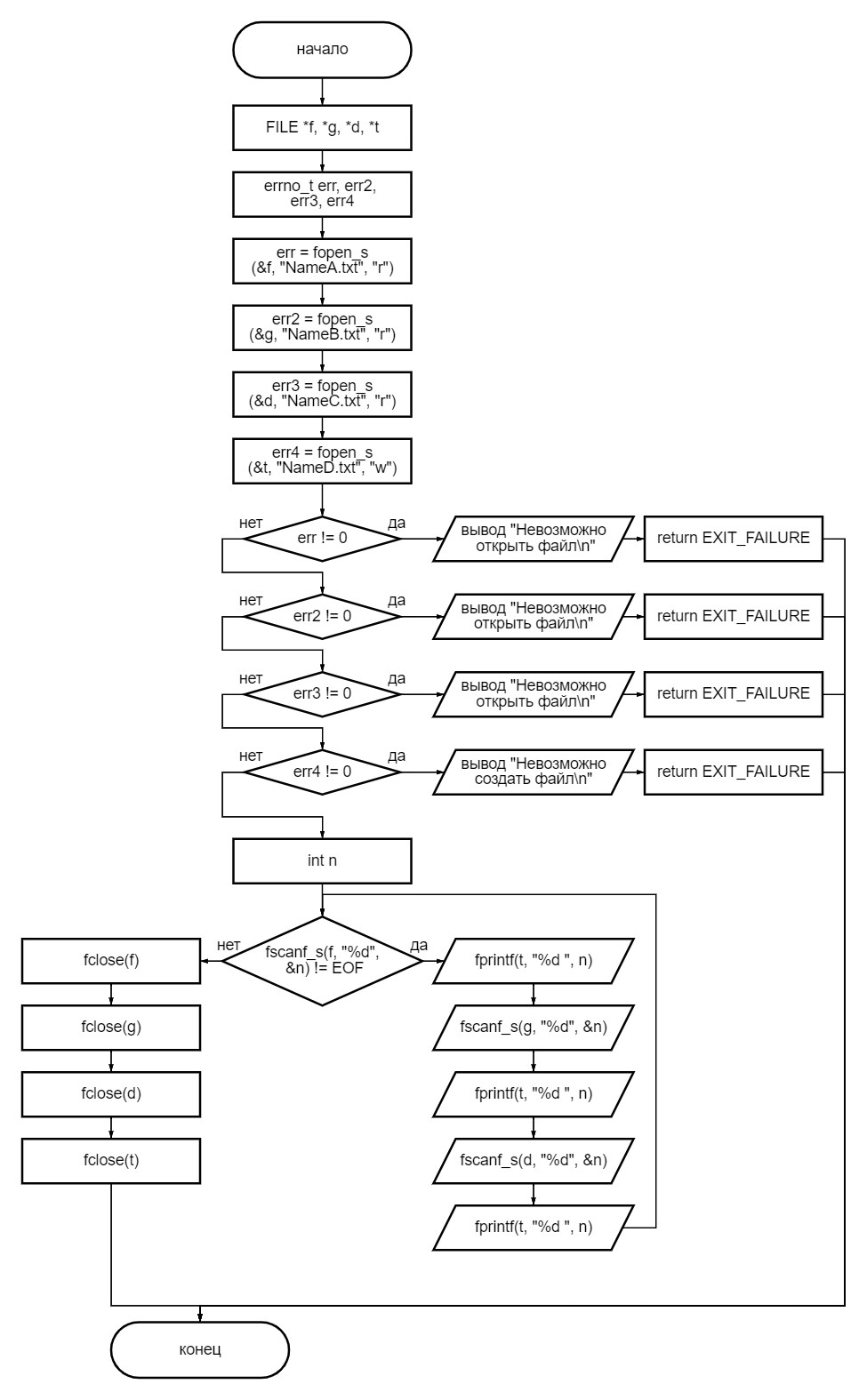
fclose(f); // закрываем файл

fclose(g); // закрываем файл

fclose(d); // закрываем файл

fclose(t); // закрываем файл

}



|  |  |
| --- | --- |
| **4** | 1. Компоненты файла **fileA** – целые отличные от нуля положительные и отрицательные числа. Получить файл **fileB**, состоящий из положительных чисел.  2. Компоненты файла **f** – целые двухзначные числа. Получить файл **g**, образованный из **f** включением только чисел больших некоторого числа, вводимого с клавиатуры. |

#include <stdio.h>

#include <cstdlib>

#include <iostream>

int main() {

setlocale(LC\_CTYPE, "Russian");

FILE \*f, \*g;

errno\_t err, err2;

err = fopen\_s(&f, "fileA.txt", "r"); // открываем первый файл с именем "fileA.txt" для чтения

err2 = fopen\_s(&g, "fileB.txt", "w"); // открываем второй файл с именем "fileB.txt" для записи

if (err != 0) // делаем проверку открытия первого файла

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

if (err2 != 0) // делаем проверку открытия второго файла

{

perror("Невозможно создать файл\n");

return EXIT\_FAILURE;

}

int n;

while (fscanf\_s(f, "%d", &n) != EOF) { // пока не конец файла, считываем число

if (n > 0) { // проверяем, чтобы оно было положительным

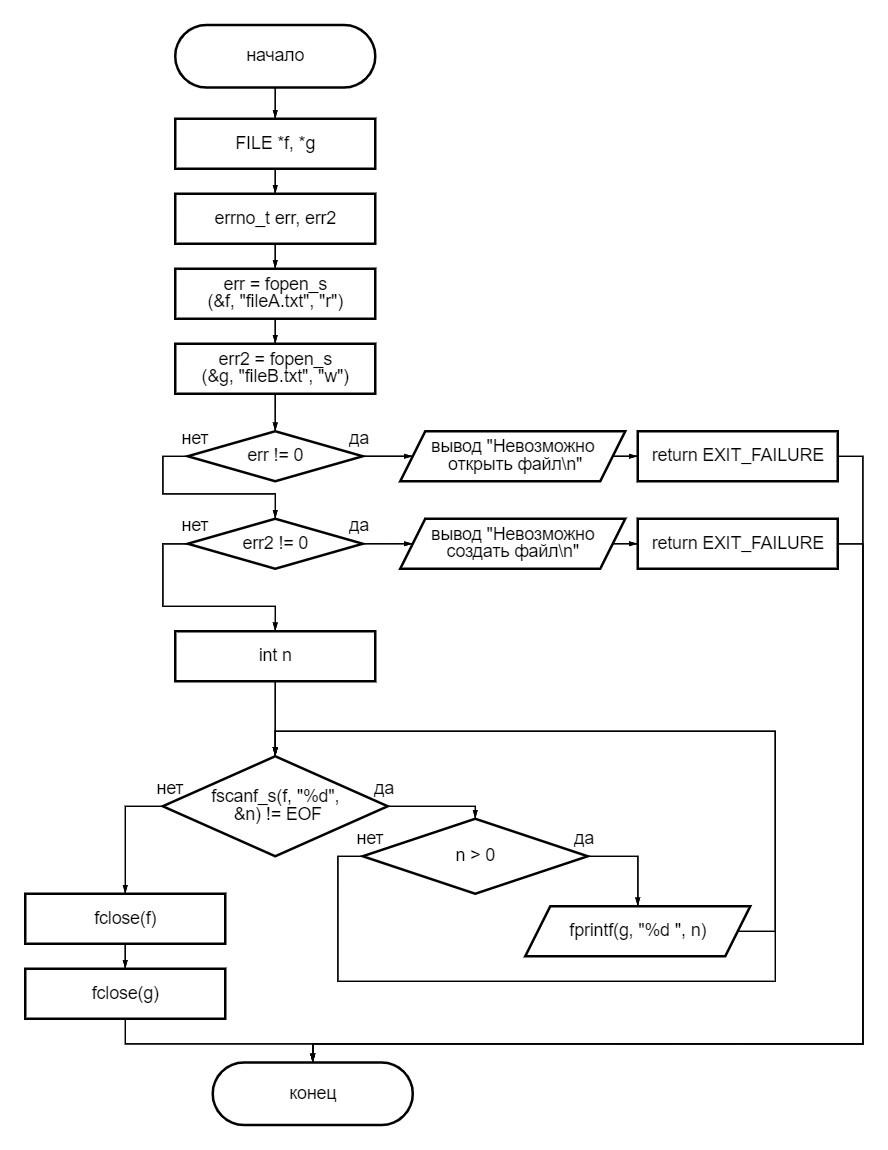
fprintf(g, "%d ", n); // записываем число во второй файл

}

}

fclose(f); // закрываем файл

fclose(g); // закрываем файл

}

#include <stdio.h>

#include <cstdlib>

#include <iostream>

int main() {

setlocale(LC\_CTYPE, "Russian");

FILE \*f, \*g;

errno\_t err, err2;

err = fopen\_s(&f, "f.txt", "r"); // открываем первый файл с именем "f.txt" для чтения

err2 = fopen\_s(&g, "g.txt", "w"); // открываем второй файл с именем "g.txt" для записи

if (err != 0) // делаем проверку открытия первого файла

{

perror("Невозможно открыть файл\n");

return EXIT\_FAILURE;

}

if (err2 != 0) // делаем проверку открытия второго файла

{

perror("Невозможно создать файл\n");

return EXIT\_FAILURE;

}

int n;

int k;

printf("Введите число, больше которого будем искать элементы ");

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

while (fscanf\_s(f, "%d", &n) != EOF) { // пока не конец файла, считываем число

if (n > k) { // проверяем, чтобы оно было больше введённого

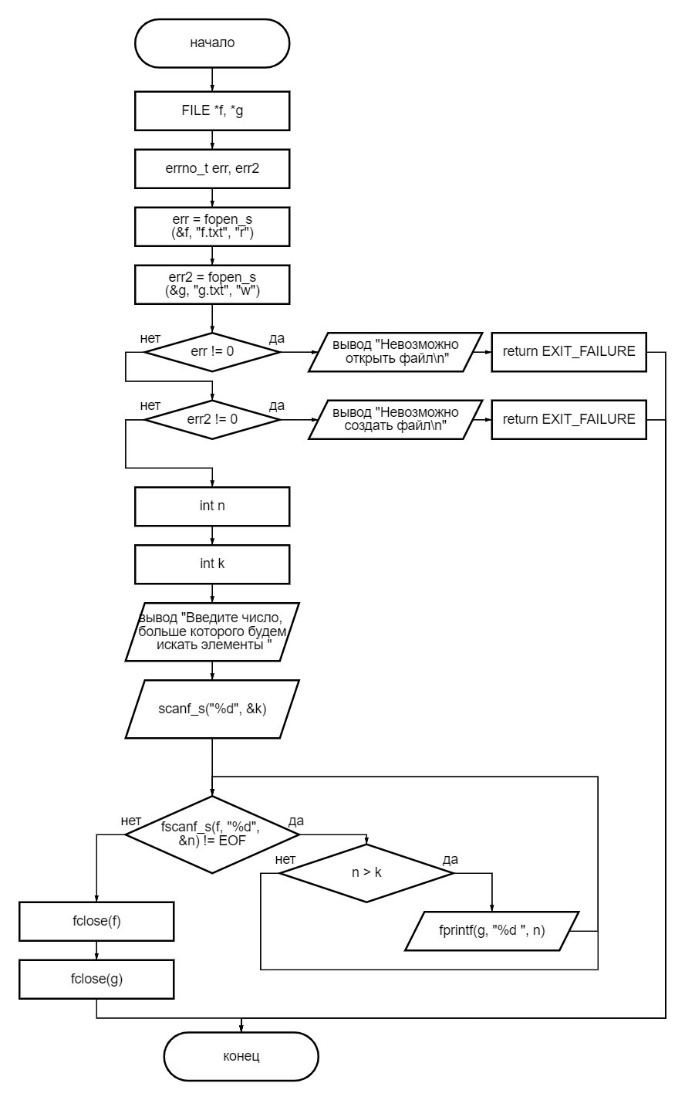
fprintf(g, "%d ", n); // записываем число во второй файл

}

}

fclose(f); // закрываем файл

fclose(g); // закрываем файл

}