Государственное ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ   
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«Донецкий национальный технический университет»

Д09.03.04-ПОИС.20-20/5847.ЛР

***Кафедра*** искусственного интеллекта  
 и системного анализа

Лабораторная работа №4

по дисциплине "Основы программирования"

на тему: "Работа с файлами"

Проверили:

\_\_\_\_\_\_\_\_\_\_\_\_ асс. А.П. Семёнова

(дата, подпись)

\_\_\_\_\_\_\_\_\_\_\_\_ асс. И.В. Савицкая

(дата, подпись)

Выполнил:

\_\_\_\_\_\_\_\_\_\_\_\_ ст.гр.ПИ-20г М. А. Евсеев

(дата, подпись)

Донецк – 2020

Вариант 9

1. В исходном текстовом файле «In.txt» посчитать количество строк, которые являются правильной записью вещественного числа. В результирующий файл «Out.txt» записать строку вида «В исходном файле \_ строк, удовлетворяющих заданному условию», если количество найденных строк больше нуля или «В исходном файле нет искомых строк» в противном случае. Если исходного файла не существует, то записать в результирующий файл сообщение об ошибке.

# Ход работы

Описание характеристик входного и выходного файла

Исходный файл:

Имя файла - «In.txt»;

Тип элемента – вещественный;

Режим доступа – чтение.

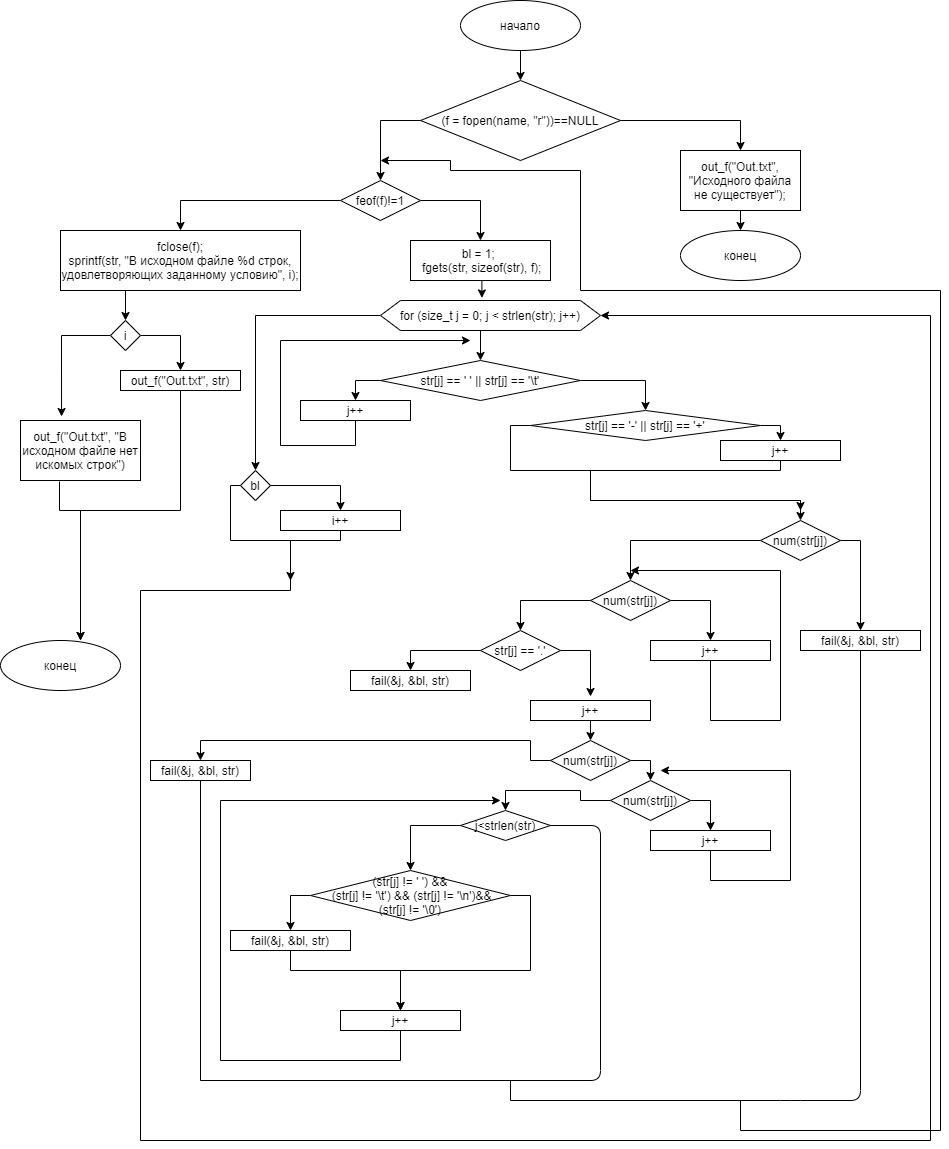
Выходной файл:

Имя файла - «Out.txt»;

Тип элемента – символьный;

Режим доступа – запись.

Алгоритм



# Листинг программы

Код программы, созданной в Visual Studio 2019:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define num(x) ((x>='0')&&(x<='9'))

//Запись строки str в файл name

void out\_f(char\* name, char\* str) {

FILE\* f;

f = fopen(name, "w");

fputs(str, f);

fclose(f);

}

//Завершение цикла и обнуление логической переменной

void fail(int \*j, int \*bl, char \* str){

\*j= strlen(str);

\*bl = 0;

}

int main(void) {

FILE\* f;

char str[255], \*a;

int i = 0 , bl;

char name[] = "In.txt";

if ((f = fopen(name, "r")) == NULL)

{

out\_f("Out.txt", "Исходного файла не существует");

return 0;

}

while(feof(f)!=1) {

bl = 1;

fgets(str, sizeof(str), f); // Чтение строки из файла

for (size\_t j = 0; j < strlen(str); j++) {//каждая буква из строки

while (str[j] == ' '|| str[j] == '\t') j++; //пока отступы

(str[j] == '-' || str[j] == '+') ? (j++) : NULL; //если перед числом будет знак

if (num(str[j])) {//если цифра - проверяем, иначе выходим

while (num(str[j])) j++; //пока цифры

if (str[j] == '.') {//если точка после цифр - проверяем, иначе выходим

j++;

if (num(str[j])) { // если цифра после точки проверяем, иначе выходим

while (num(str[j])) {//пока цифры

j++;

}

while (j<strlen(str)) {

if ((str[j] != ' ') && (str[j] != '\t') && (str[j] != '\n')&& (str[j] != '\0')) {

fail(&j, &bl, str);

}

j++;

}

}

else fail(&j, &bl, str); //если после точки нет цифры -выходим

}

else fail(&j, &bl, str);//если нет точки после цифр -выходим

}

else fail(&j, &bl, str);//если после отступов нет цифры -выходим

}

(bl) ? (i++) : NULL;

}

fclose(f);

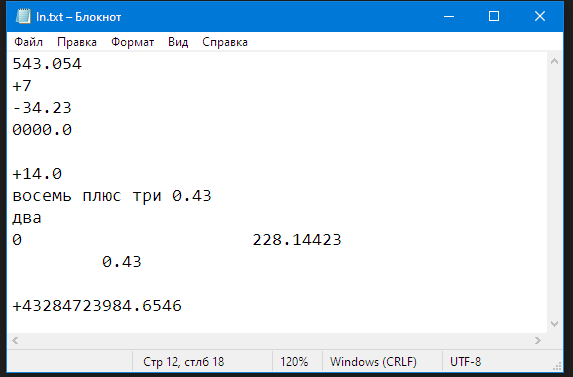
sprintf(str, "В исходном файле %d строк, удовлетворяющих заданному условию", i);

(i) ? (out\_f("Out.txt", str)) : (out\_f("Out.txt", "В исходном файле нет искомых строк"));

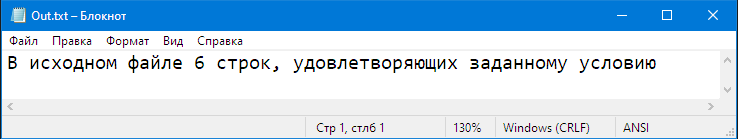
}

# Контрольные примеры

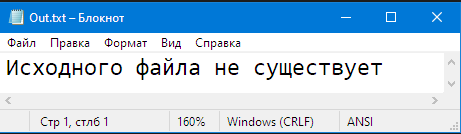
Исходный текстовый файл «In.txt»:



После запуска программы создается текстовый файл «Out.txt»:



Если нет исходного текстового файла не существует, то результирующий файл выглядит следующим образом:



2. В файле записаны числа. Создать новый файл, содержащий длины всех убывающих подпоследовательностей элементов исходного файла.

# Ход работы

Описание характеристик входного и выходного файла

Исходный файл:

Имя файла – задается пользователем;

Тип элемента – вещественный;

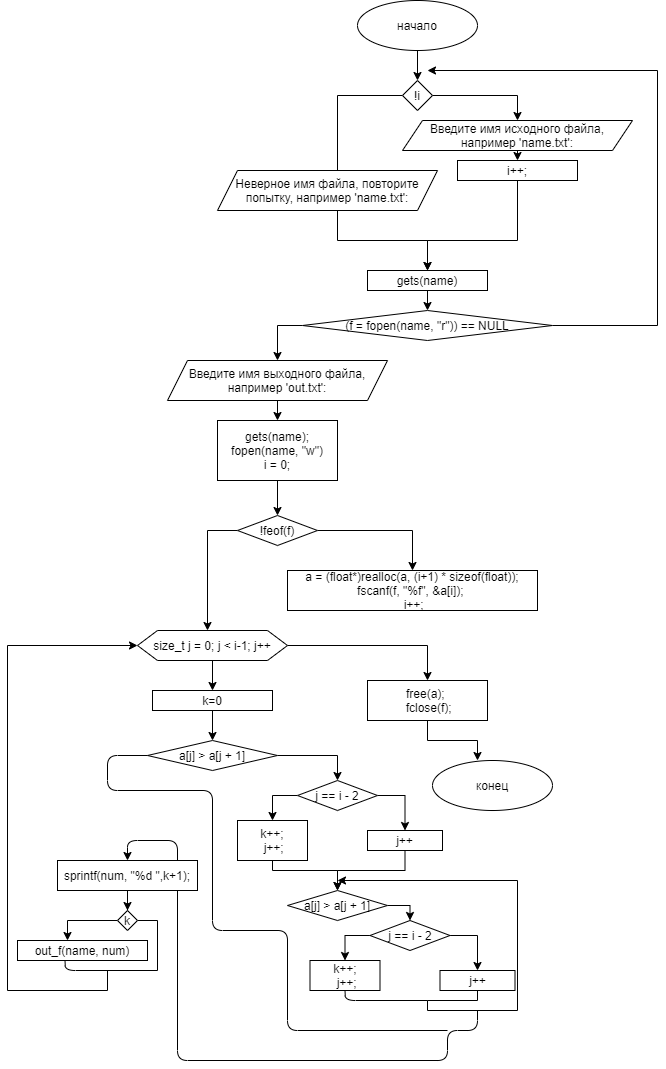
Режим доступа – чтение.

Выходной файл:

Имя файла - задается пользователем;

Тип элемента – целочисленный;

Режим доступа – запись.



# Листинг программы

Код программы, созданной в Visual Studio 2019:

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

#include <Windows.h>

//запись файла

void out\_f(char\* name, char\* str) {

FILE\* f;

f = fopen(name, "a");

fputs(str, f);

fclose(f);

}

void main() {

SetConsoleCP(1251);// установка кодовой страницы win-cp 1251 в поток ввода

SetConsoleOutputCP(1251); // установка кодовой страницы win-cp 1251 в поток вывода

FILE\* f;

char name[32], num[64];

int i = 0, k = 0, m=0;

float \*a = NULL;

do{ //открытие файла

if (!i) {

printf("Введите имя исходного файла, например 'name.txt':\n");

i++;

}

else printf("Неверное имя файла, повторите попытку, например 'name.txt':\n");

gets(name);

} while ((f = fopen(name, "r")) == NULL);

printf("Введите имя выходного файла, например 'out.txt':\n");

gets(name);

fopen(name, "w");//создание нового выходного файла

i = 0;

while (!feof(f)) {

a = (float\*)realloc(a, (i+1) \* sizeof(float));

fscanf(f, "%f", &a[i]);

i++;

}

for (size\_t j = 0; j < i-1; j++) {

k = 0;

if (a[j] > a[j + 1]) {

if (j == i - 2) j++;

else {

k++;

j++;

}

while (a[j] > a[j + 1]) {

if (j == i - 2) j++;

else {

k++;

j++;

}

}

}

sprintf(num, "%d ",k+1);

(k) ? (out\_f(name, num)) : 0;

}

free(a);

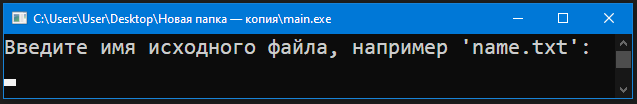
fclose(f);

getchar();

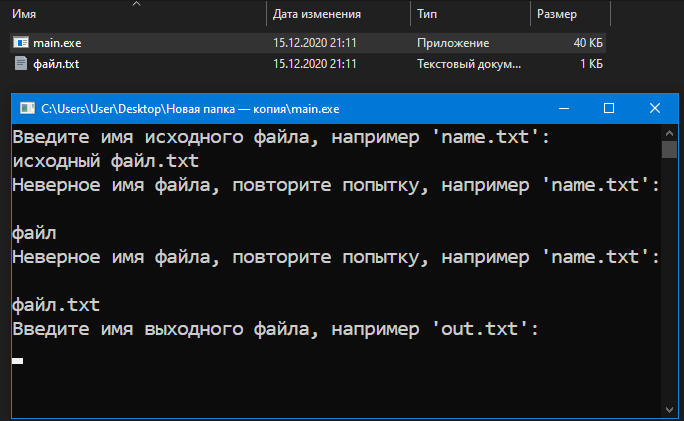
}

# Контрольные примеры

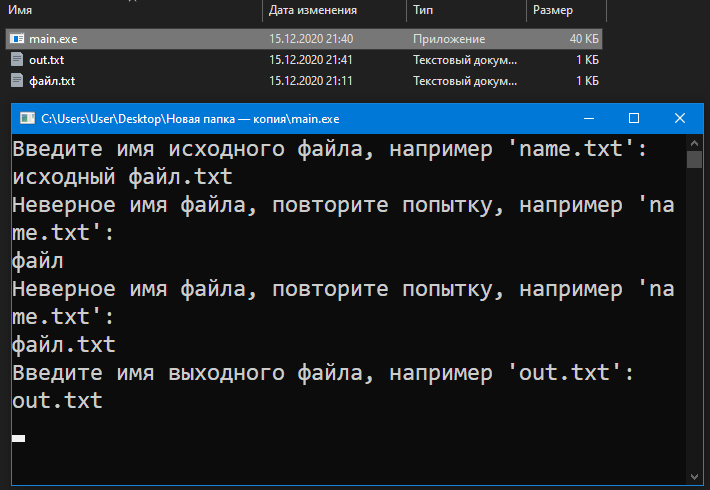
Открыв программу, мы должны ввести имя исходного файла:



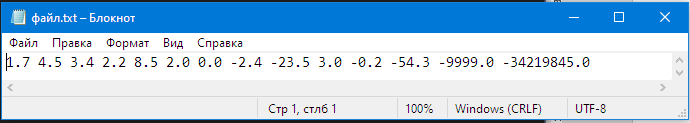
После ввода правильного имя исходного файла, требуется ввести результирующий файл:



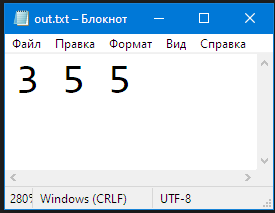
Введя имя выходного файла, программа его создает:



Содержимое файла «файл.txt»:



Содержимое файла «out.txt»:



Вывод: научился работать с файлами, проводить анализ задачи, определять характеристики файлов и их способ обработки, создавать программы и функции на Си, работающие с файлами.