**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ**(**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**)» (**МАИ)**

Институт №3   
«Системы управления, информатика и  
электроэнергетика»

Кафедра 304   
Вычислительные машины, системы и сети

**ПРОГРАММИРОВАНИЕ   
НА ЯЗЫКАХ   
ВЫСОКОГО УРОВНЯ**

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №9

Бригада №10

Выполнил:

Студент группы М30-211Б-19:

Брюхов И.А.

Проверил:

Дмитриева Е.А

Москва 2020

**Оглавление**

**Задание** **3**

**Схема алгоритма** **4-6**

**Текст программы** **7-9**

**Тесты** **10-11**

**Вывод** **12**

Задание:

Вариант №4

Разработать программу и подпрограмму (подпрограммы), работающую с текстовым файлом и преобразовать входной текстовый файл в выходной двоичный, содержащий записи следующего вида:

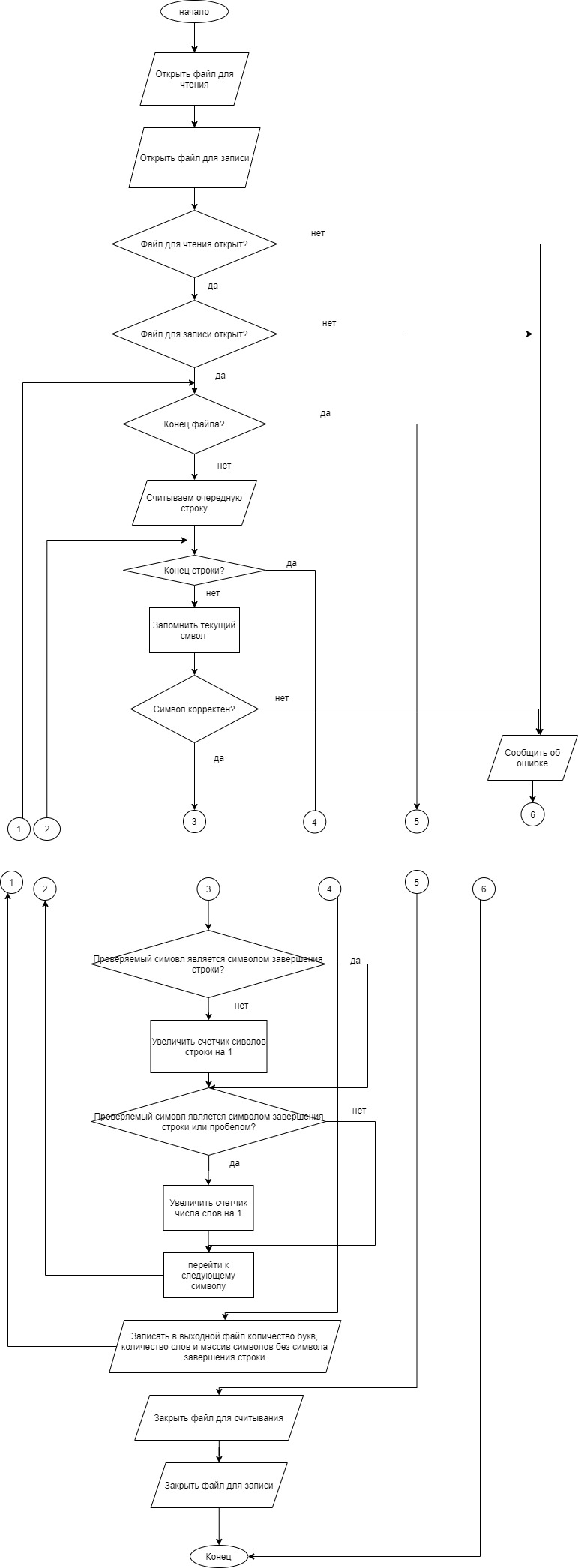
• данные типа float – длина строки;

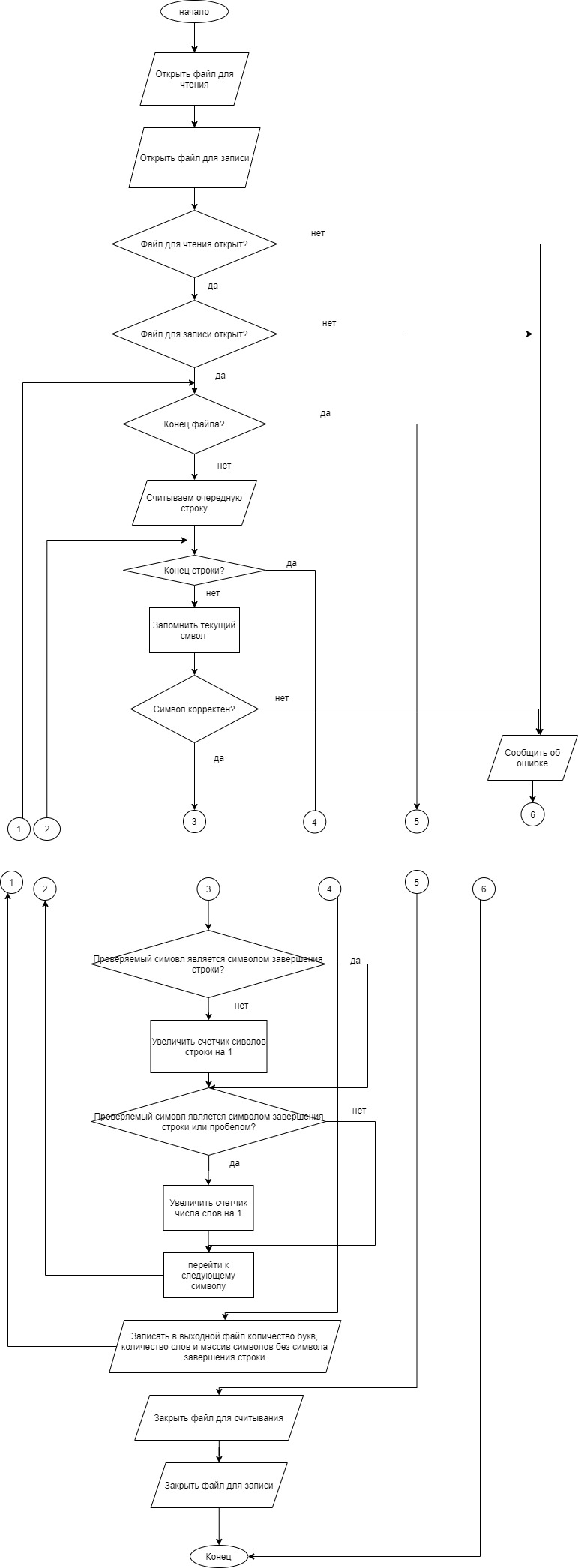
• данные типа int – количество слов в строке;

• массив типа char – строка входного файла без завершающего символа '\n'.

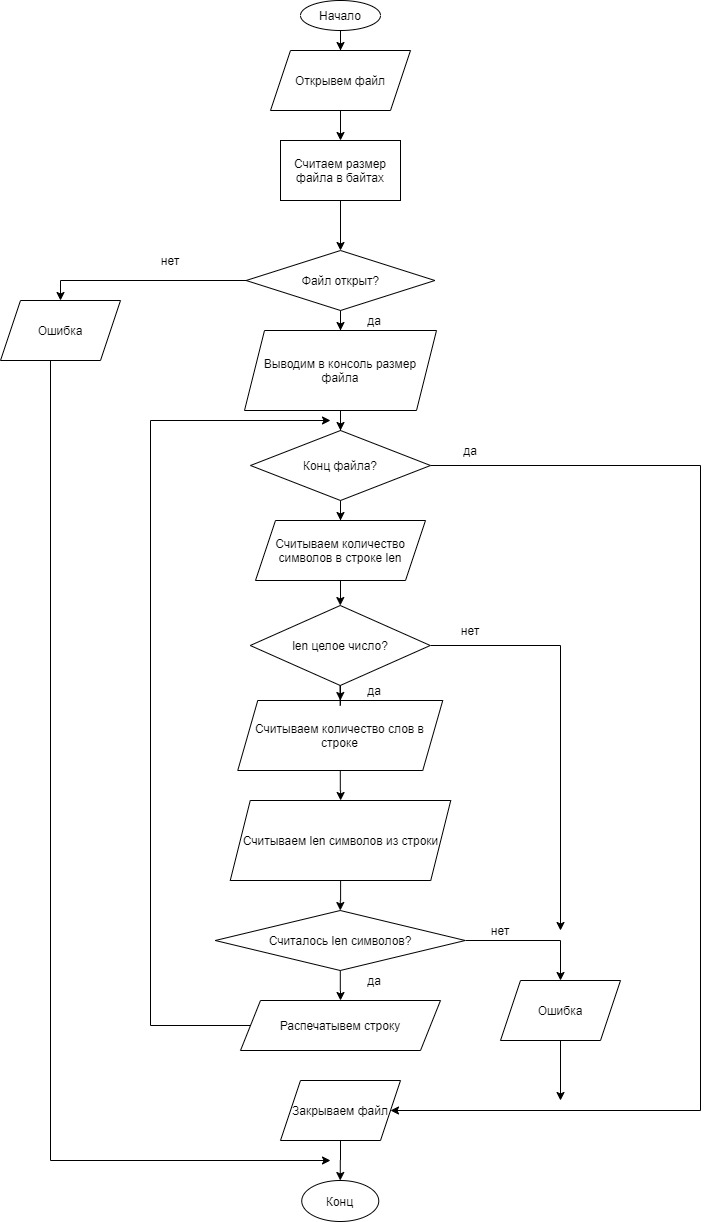
Схема алгоритма:

Часть 1:





Часть 2:



Текст программы:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*Project name: Laba9 \*

\*Project Type: Win32 Console Application \*

\*File Name:Laba9.ccp \*

\*Language: C++ \*

\*Programmer(1): Брюхов Илья; М3О-211Б-19; Вариант №4\*

\*Modified by:05.10.2020 \*

\*Created:16.09.2020 \*

\*Lastrevision: 05.10.2020 \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Подключаемые Библиотеки \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <locale.h>

#include <string.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Глобальные константы \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

const int BUF\_SIZE = 256;

const char\* fname = "test.txt";

const char\* result = "result.bin";

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Основная Программа \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

setlocale(LC\_ALL, "rus");

FILE\* in;

FILE\* out;

if ((in = fopen(fname, "r")) == NULL)

{

printf("Ошибка открытия файла %s\n",fname);

return 1;

}//if

if ((out = fopen(result, "wb")) == NULL)

{

printf("Ошибка открытия файла %s\n", result);

return 1;

}//if

char buf[BUF\_SIZE];

const char delim[] = " .,;:?!\t\n";//Знаки разделители

while (fgets(buf, BUF\_SIZE, in) != NULL) {

float len = 0;//Длина строки

int word\_count = 0;//Счетчик количества слов в строке

char\* ptr;

if((ptr=strchr(buf,'\n'))!=NULL){//Находим указатель на символ конца строки

\*ptr = '\0';//Удаляем знак окончания строки

}

len = strlen(buf);

char\* word;

long int cur = ftell(out);//Запоминаем положение курсора в начале строки

fseek(out, sizeof(int) + sizeof(float), SEEK\_CUR);//Оставляем место для записи длины строки и кол. слов

fwrite(buf, sizeof(char), len, out);//Записываем строку

fseek(out, cur, SEEK\_SET);//Возвращаем курсор в начало строки

for (word = strtok(buf, delim); word != NULL; word = strtok(NULL, delim))//Считаем количество слов

word\_count++;

fwrite(&len, sizeof(float), 1, out);//Записываем длину строки

fwrite(&word\_count, sizeof(int), 1, out);//Записываем кол. слов

fseek(out, 0, SEEK\_END);//Возвращаем курсор в конец файла

}//while

if (!feof(in)) {

fclose(in);

fclose(out);

printf("Ошибка чтения\n");

return 1;

}//if

fclose(in);//Закрываем входной поток из файла

fclose(out);//Закрываем поток записи в файл

PrintFile(result);//Распечатываем полученный двоичный файл

}//int main()

void PrintFile(const char\* file\_name)

{

// [float] [int] [char\*]

//Структура файла: [количество символов][количество слов][строка]

FILE\* in;

if ((in = fopen(file\_name, "rb")) == NULL) {//Открытие файла и проверка ошибок

printf("Ошибка чтения бинарного файла \"%s\"\n", fname);

return;

}//if

fseek(in, 0,SEEK\_END);//Перемещаем курсор в конец файла

long int file\_size = ftell(in);//Узнаем размер файла в байтах

rewind(in);//Возвращаем курсор в начало файла

printf("Размер файла в байтах: %li\n", file\_size);

float len;//Длина строки

int word\_count;//Количество слов

char str[BUF\_SIZE];//Строка

while (fread(&len, sizeof(float), 1, in)==1)//Цикл пока можно считать строку

{

if ((float)((int)len) != len) {//Проверка является ли размер строки целым числом

printf("Неправильная структура файла\n");

fclose(in);

return;

}//if

if (fread(&word\_count, sizeof(int), 1, in) !=1)//Проверка структуры файла

{

printf("Неправильная структура файла\n");

fclose(in);

return;

}//if

if (fread(str, sizeof(char), (int)len, in) != (int)len) {

if (feof(in)) {

printf("Неправильная структура файла\n");

}//if

else {

printf("Ошибка чтения файла\n");

}//else

fclose(in);

return;

}//if

str[(int)len] = '\0'; //Добавляем в конец строки знак конца строки

puts(str); //Печатаем строки

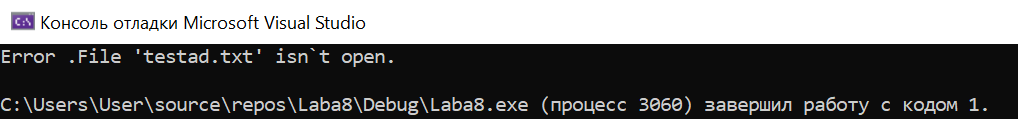
}//while

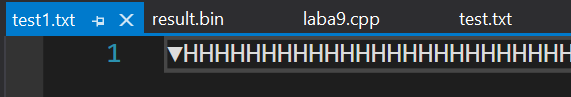
fclose(in);

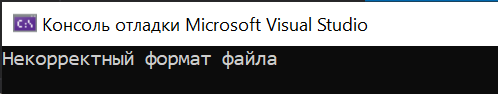
}//PrintFile

Тесты:

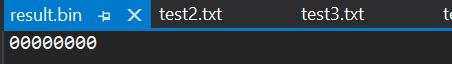
1. Некорректное имя файла
   1. Цель: Проверить работоспособность в случае когда не удастся открыть файл fname = "not\_exist.txt"
   2. Исходные данные:
   3. Ожидаемый результат: "Ошибка открытия файла not\_exist.txt "
   4. Полученный результат:

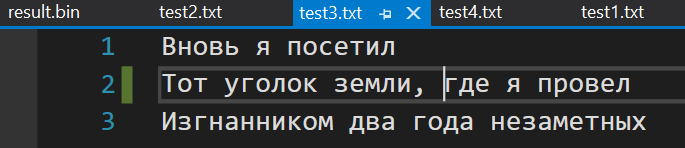


1. Файл содержащий недопустимые символы
   1. Цель: Проверить работоспособность в случае когда файл содержит недопустимый символ из таблицы ASCII под номером 31
   2. Исходные данные: 
   3. Ожидаемый результат: "Некорректный формат файла"
   4. Полученный результат:



1. Пустой входной файл
   1. Цель: Проверить работоспособность в случае когда файл пуст
   2. Исходные данные: пустой входной файл
   3. Ожидаемый результат: Пустой выходной файл
   4. Полученный результат:



1. Корректный файл с русскими словами
   1. Цель: Проверить работоспособность в случае когда файл содержит русские буквы
   2. Исходные данные: 
   3. Ожидаемый результат:

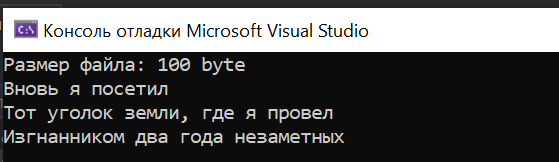
Размер файла: 100 byte

Вновь я посетил

Тот уголок земли, где я провел

Изгнанником два года незаметных

* 1. Полученный результат:



Вывод:

Разработка программы завершена на том основании, что

1. Полученный результат совпал с ожидаемым

2. Считаем набор тестов полным