**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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 ЯЗЫКЕ ВЫСОКОГО УРОВНЯ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 9

«Изучение принципов работы с двоичными файлами»

Вариант 4

Выполнили:

студент группы М30-209Б-19: Кузнецов И.И.

студент группы М30-209Б-19: Мухаметгалиев А.Р.

Проверил:

Доцент, к.т.н., Новиков П. В.

Доцент, к.т.н., Ивашенцев И. В.

Москва 2020

**Содержание**

1. **Задание.**
2. **Структурная схема алгоритма программы и подпрограмм.**
3. **Текст программы.**
4. **Результаты работы программы.**
5. **Вывод.**

**Задание**

Вариант 4

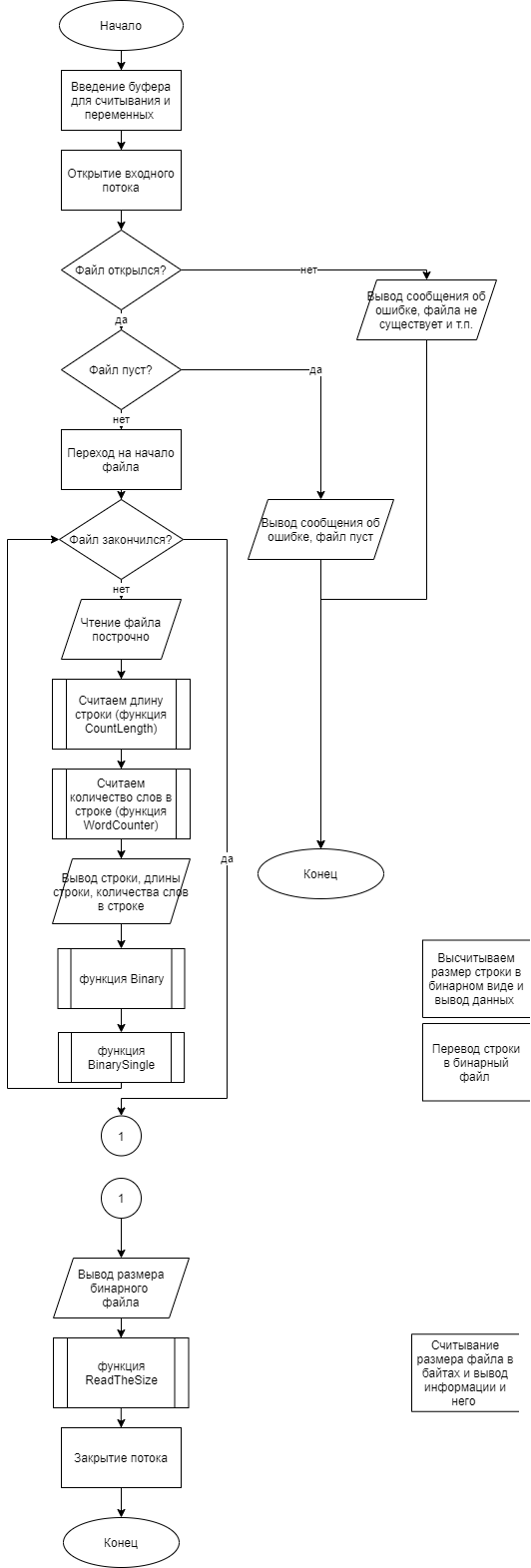
1. Разработать программу и подпрограмму (подпрограммы), работающую с текстовым файлом и выполняющую действия согласно варианту задания:

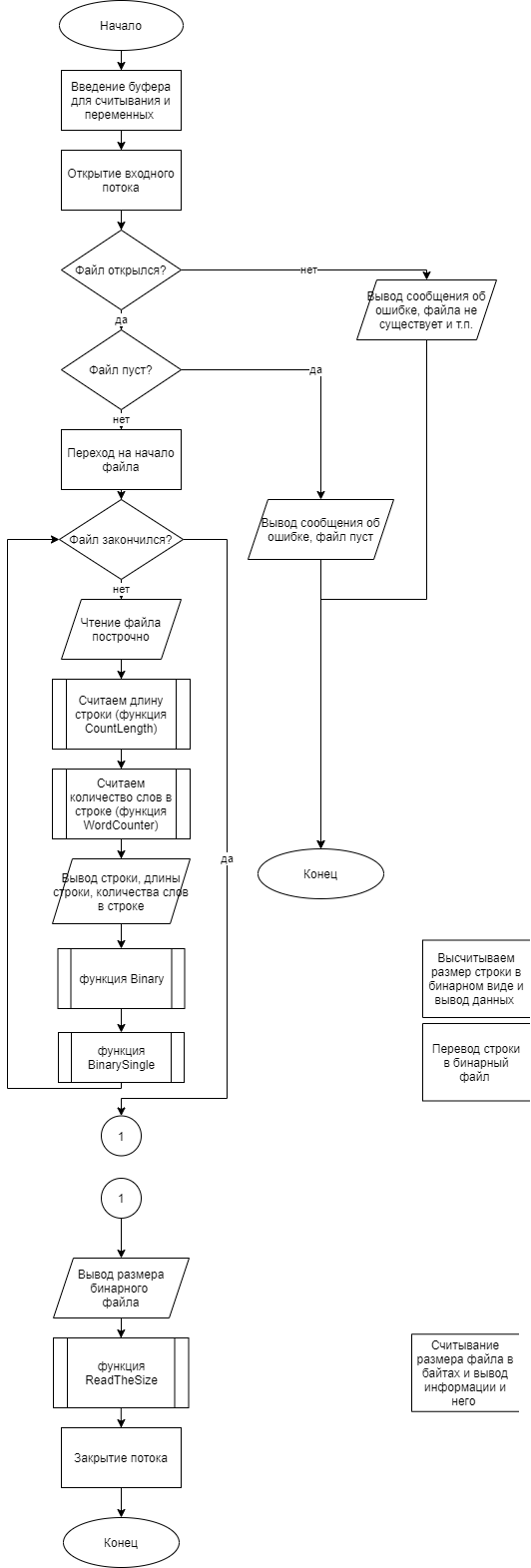
Преобразовать входной текстовый файл в выходной двоичный, содержащий записи следующего вида:

* + данные типа float – длина строки;
  + данные типа int – количество слов в строке;
  + массив типа char – строка входного файла без завершающего символа '\n'.

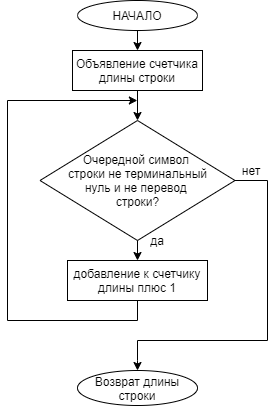
1. Разработать программу, которая выводит на экран размер и содержимое двоичного файла заданной структуры, созданного в соответствии с заданием из п.1.

**Структурная схема алгоритма программы.**

****



**Структурная схема алгоритма подпрограммы CountLenght.**

****

Функция “CountLenght”

Назначение: Подсчет длины строки

Сигнатура функции: float CountLength(const char \*str)

Обращение к функции: CountLenght

Параметры: const char \*str

**Структурная схема алгоритма подпрограммы WordCounter.**

****

Функция “WordCounter”

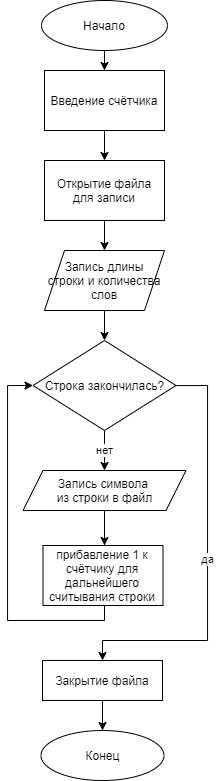
Назначение: Подсчёт количества слов в строке

Сигнатура функции: int WordCounter(const char \*str)

Обращение к функции: WordCounter

Параметры: const char \*str

**Структурная схема алгоритма подпрограммы BinarySingle.**

****

Функция “BinarySingle”

Назначение: Перевод строки в бинарный файл

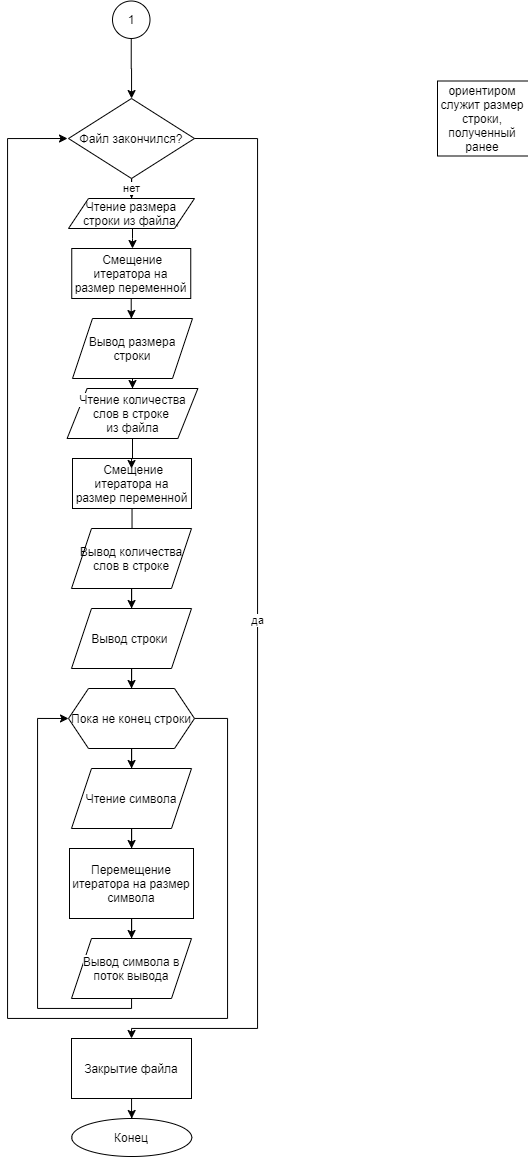
Сигнатура функции: void BinarySingle(const char \*str, float lengthString, int numberWords)

Обращение к функции: BinarySingle

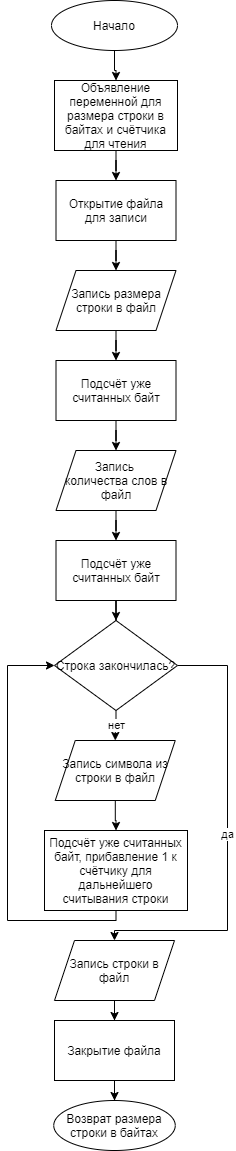
Параметры: const char \*str, float lengthString, int numberWords

**Структурная схема алгоритма подпрограммы** **ReadTheSize.**





**Структурная схема алгоритма подпрограммы Binary.**



Функция “Binary”

Назначение: Высчитывание размера строки в бинарном виде и вывод данных

Сигнатура функции: int Binary(const char \*str, float lengthString, int numberWords)

Обращение к функции: Binary

Параметры: const char \*str, float lengthString, int numberWords

**Текст программы**

#include <stdio.h>

int WordCounter(const char \*str);

float CountLength(const char \*str);

void BinarySingle(const char \*str, float lengthString, int numberWords);

void ReadTheSize();

int Binary(const char \*str, float lengthString, int numberWords);

int main() {

/\*

\* Условие задачи 1:

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'.

\* Условие задачи 2:

Разработать программу, которая выводит на экран размер и содержимое двоичного файла

заданной структуры, созданного в соответствии с заданием из п.1.

\*/

FILE \*in;

char str[256]; //буфферная строка

int sizeOfFile = 0;

in = fopen("../Fin.txt", "rt");

int numberWords; //количество слов в строке

float lengthString; //длина строки

if (in == NULL) { //проверка на наличие файла

perror("ERROR with a reading file ");

return -1;

}

if (getc(in) == EOF) { //проверка на пустоту файла

printf("File is empty!\n");

return -2;

}

rewind(in); //переход на начало файла

while (fgets(str, sizeof(str), in) != NULL) { //считать построчно, пока не конец файла

lengthString = CountLength(str); //считаем длину строки

numberWords = WordCounter(str); //считаем количество слов в строке

printf("The string is: %s", str); //вывод строки

printf("The string's size is: %f\n", lengthString); //вывод длины строки

printf("The number of words: %d\n\n", numberWords); //вывод количества слов в строке

sizeOfFile += Binary(str, lengthString, numberWords); //высчитываем размер строки(бин) и вывод информации

//BinarySingle(str, lengthString, numberWords); //перевод строки в бинарный файл

}

printf("------------------------------------\n");

printf("The size of binary file: %d", sizeOfFile); //вывод размер бинарного файла

//ReadTheSize(); //считаем размер файла в байтах и выводим всю информацию из него

fclose(in);

return 0;

}

/\*

\* Функция для подсчета длины строки

\* @param \*str - входная строка, в которой необходимо найти количество слов

\* return количество слов, в строке

\*/

int WordCounter(const char \*str) {

int counter = 0;

int i = 0;

while (str[i] != '\0') { //проходимся по всей строке

while (str[i] != ' ' && str[i] != '\0') i++; //находим слово идем по нему и

counter++; //добавляем к общему количеству

while (str[i] == ' ') i++; //пропускаем пробелы

}

return counter;

}

float CountLength(const char \*str) { //подсчет длины строки

int i = 0;

float counter = 0;

while (str[i] != '\0' && str[i] != '\n') {

counter++;

i++;

}

return counter;

}

void BinarySingle(const char \*str, float lengthString, int numberWords) {

int i = 0;

FILE \*out;

out = fopen("../Fout.bin", "a+b");

fwrite(&lengthString, sizeof(float), 1, out); //записываем длину строку

fwrite(&numberWords, sizeof(int), 1, out); //записываем количество слов

while (str[i] != '\0' && str[i] != '\n') { //записываем строку

fwrite(&str[i], sizeof(char), 1, out);

i++;

}

fclose(out);

}

int Binary(const char \*str, float lengthString, int numberWords) {

int i = 0;

int sizeOfString = 0;

FILE \*out;

out = fopen("../Fout.bin", "a+b");

fwrite(&lengthString, sizeof(float), 1, out);

printf("The string's size is: %f\n", lengthString);

sizeOfString += sizeof(float);

fwrite(&numberWords, sizeof(int), 1, out);

printf("The number of words: %d\n\n", numberWords);

sizeOfString += sizeof(int);

while (str[i] != '\0' && str[i] != '\n') { //записываем строку

fwrite(&str[i], sizeof(char), 1, out);

sizeOfString += sizeof(char);

i++;

}

printf("The string is: %s", str);

fclose(out);

return sizeOfString;

}

void ReadTheSize() {

FILE \*rebin;

rebin = fopen("../Fout.bin", "rb");

int sizeOfFile; //размер файла

float lengthString; //длина строки

int numberWords; //количество слов в строке

char symbol; //буферный символ

int i = 0; //итератор

fseek(rebin, 0, SEEK\_END); //смещение указателя чтения на конец файла

sizeOfFile = ftell(rebin); //получение размера файла

printf("The size of a file is: %u bytes \n\n", sizeOfFile);

fseek(rebin, 0, SEEK\_SET); //смещение указателя на начало файла

while (i < sizeOfFile) { //читаем до конца файла

fread(&lengthString, sizeof(int), 1, rebin); //чтение размера строки из файла

i += sizeof(float); //смещение итератора на размер переменной

printf("The length of this string is: %f \n", lengthString);

fread(&numberWords, sizeof(int), 1, rebin); //чтение количества слов из файла

i += sizeof(int); //смещение итератора на размер переменной

printf("The number of words is: %u \n", numberWords);

printf("%s", "The string is: ");

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

fread(&symbol, sizeof(char), 1, rebin);

i += sizeof(char);;

putchar(symbol);

}

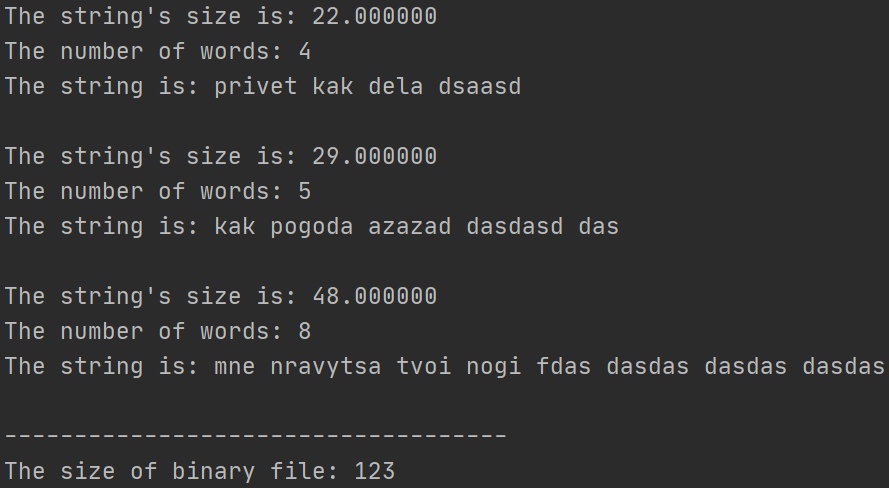
printf("\n\n");

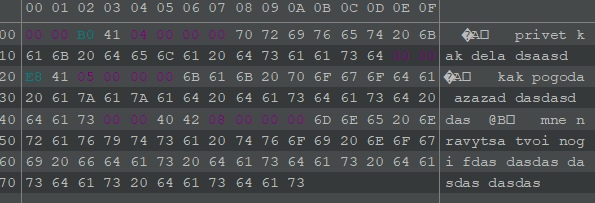
}

fclose(rebin);

}

**Результаты работы программы**





Вывод: Программа работает корректно

**Вывод**

Мы разработали программу и подпрограммы:

1. работающие с текстовым файлом и выполняющую действия согласно варианту задания:

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'.

1. Которая выводит на экран размер и содержимое двоичного файла заданной структуры, созданного в соответствии с заданием из п.1.