Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №9**

Дисциплина: основы алгоритмизации и программирования

Тема: Строковый ввод-вывод

Выполнил работу

студент группы ИВТ-20-1б

Михайлов М. А.

Проверила

Доцент кафедры ИТАС

к.т.н Полякова О. А.

Пермь, 2020

**Цель работы**

Работа с текстовыми файлами, ввод-вывод текстовой информации и ее хранение на внешних носителях.

**Постановка задачи**

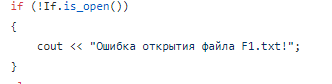
(Вариант 19)

1. Скопировать из файла F1 в файл F2 все строки, в которых нет одинаковых слов.
2. Определить количество согласных букв в последней строке F2.

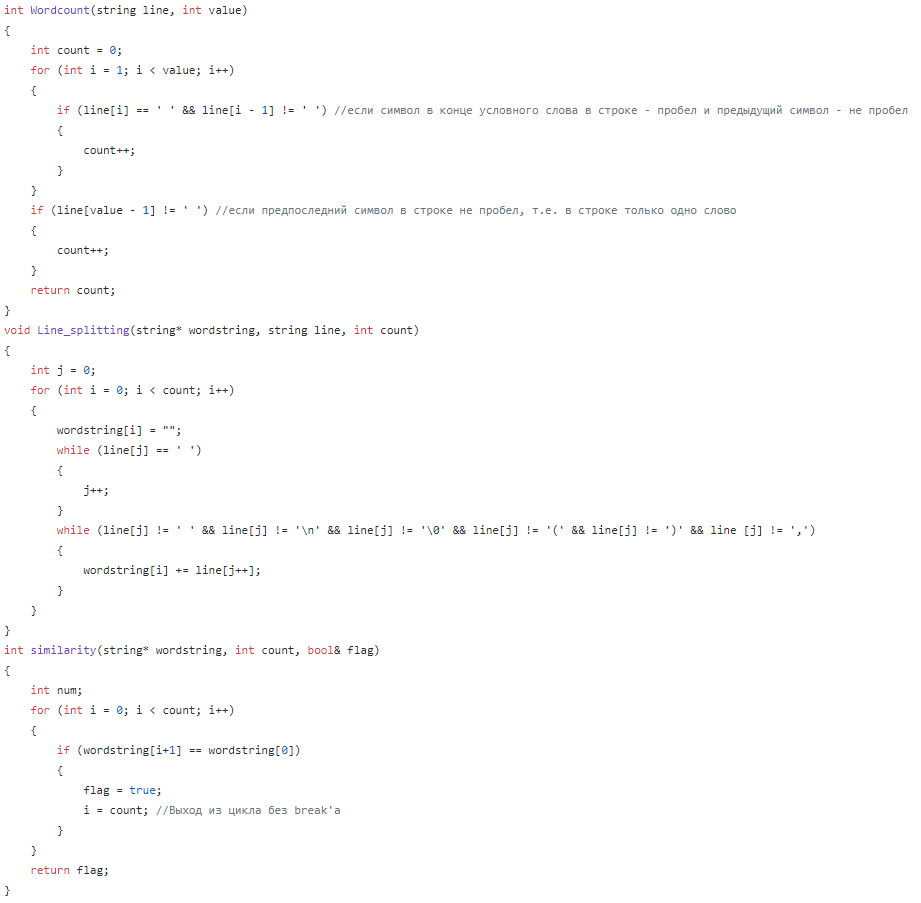
**Анализ задачи**

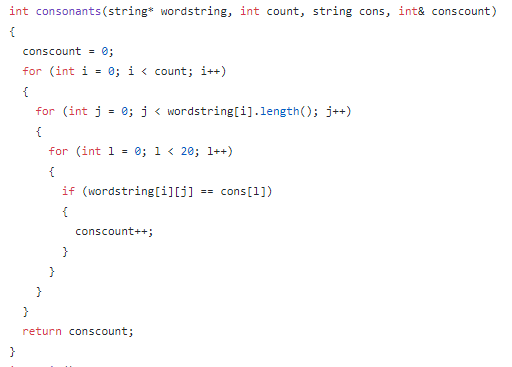
**1.** Для решения задачи необходимо:

**1.1.**  Проверить, отсутствует ли файл ввода. В случае его отсутствия сообщить об ошибке.

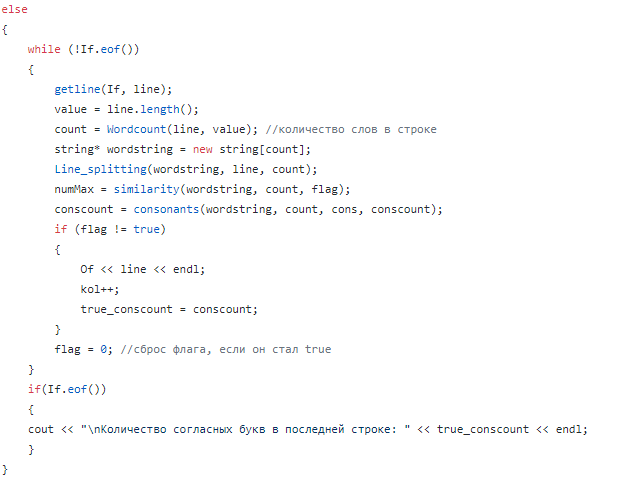
****

**1.2.** Разработать функцию для проверки строки на наличие слов-дубликатов. В строке из value символов подсчитываются слова с использованием функции Wordcount. Затем используется функция Line\_splitting, которая разделяет массив строки на слова и заполняет массив слов, и строка проверяется на наличие одинаковых слов путём сравнивания слов с первым словом

****

**1.3.** Разработать функцию consonants, чтобы подсчитать кол-во слов в последней строке, которая была скопирована в файл F2. 

**1.4.** Реализовать применение этих функций для строк файла F1.



**2.** В ходе работы были использованы следующие типы данных:

**2.1.** Для получения данных из файла F1 используется класс ifstream:



**2.2.** Для вывода данных в файл F2 используется класс ofstream:



**2.3.** Для функции Wordcount используются следующие аргументы:

1. Тип string – Входная строка, в которой будет подсчитано кол-во слов.

2. Тип int - Количество символов во входной строке.

Сама функция имеет тип int для возвращения кол-ва слов в строке.



**2.4.** Для функции Line\_splitting используются следующие аргументы:

1. Указатель на string – массив слов из строки.

2. Тип string – строка, которая будет разбита на слова.

3. Тип int – количество слов в строке.

Сама функция имеет тип void, т. к. в ходе её работы нет необходимости возвращать значение для этой функции.



**2.5.** Для функции similarity используются следующие аргументы:

1. Указатель на string – массив слов из строки.

2. Тип int – количество слов в строке.

3. Тип string – строка со всеми согласными буквами.

4. Ссылка на int – количество согласных букв в строке.

Сама функция имеет тип int для возвращения кол-ва согласных букв в строке.



**3.** Для решения задачи данные были представлены в следующем виде:

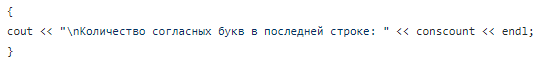
**3.1.** Для обработки строк из файла данные записаны в виде массива из слов (массив из string).

**4.** Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.** Ввод данных из файла F1 реализован с помощью функции getline.



**4.2.** Вывод данных на консоль реализован с помощью оператора cout.

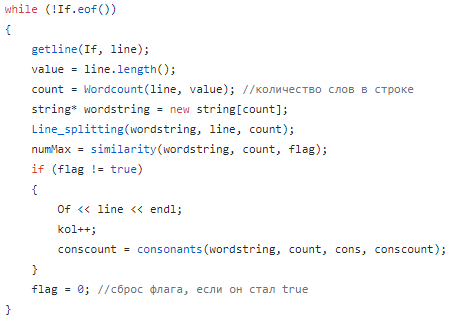


**4.3.**  Вывод данных в файл реализован с помощью оператора вывода в файл.



**5.** Поставленные задачи будут решены следующими действиями:

**5.1.** Для работы с информацией из файла построчно, все операции со строкой проводятся внутри цикла while. После каждого полного прохождения тела цикла производится считывание следующей строки. Повторение цикла продолжается, пока не достигнут конец файла.



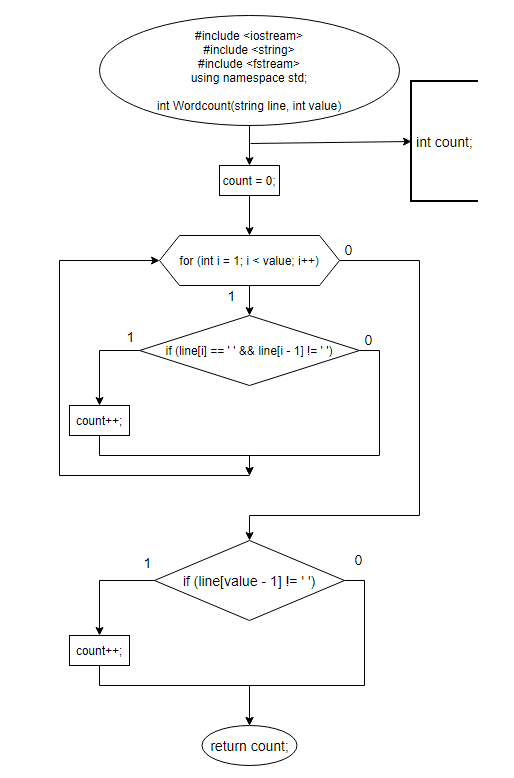
**5.2.** Поиск подходящих для копирования строк и копирование этих строк из файла F1 в F2 реализованы с помощью функций Line\_splitting и similarity.

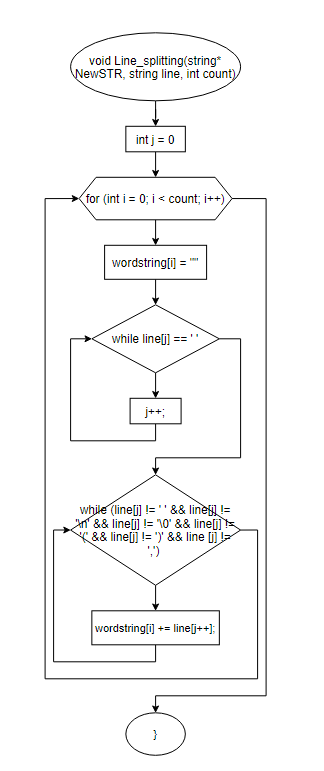


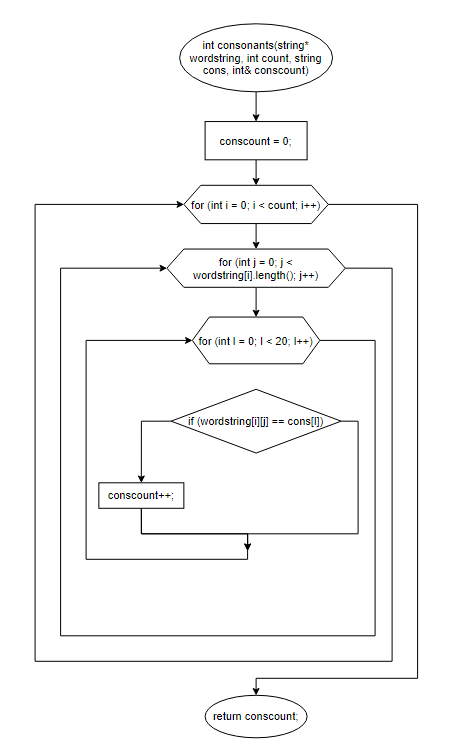
**5.3.** Подсчёт количества согласных букв в последней скопированной строке реализовано с использованием функции consonants.

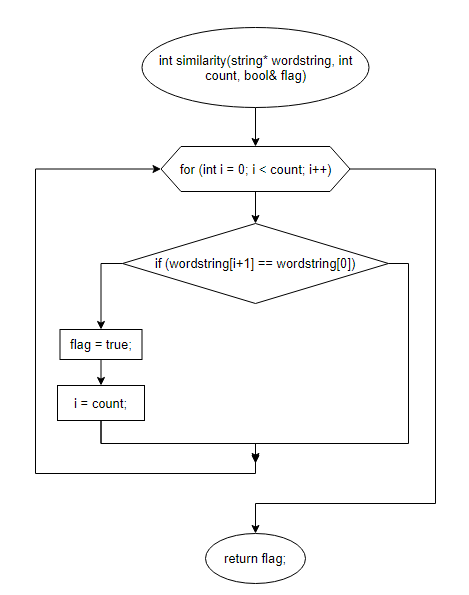


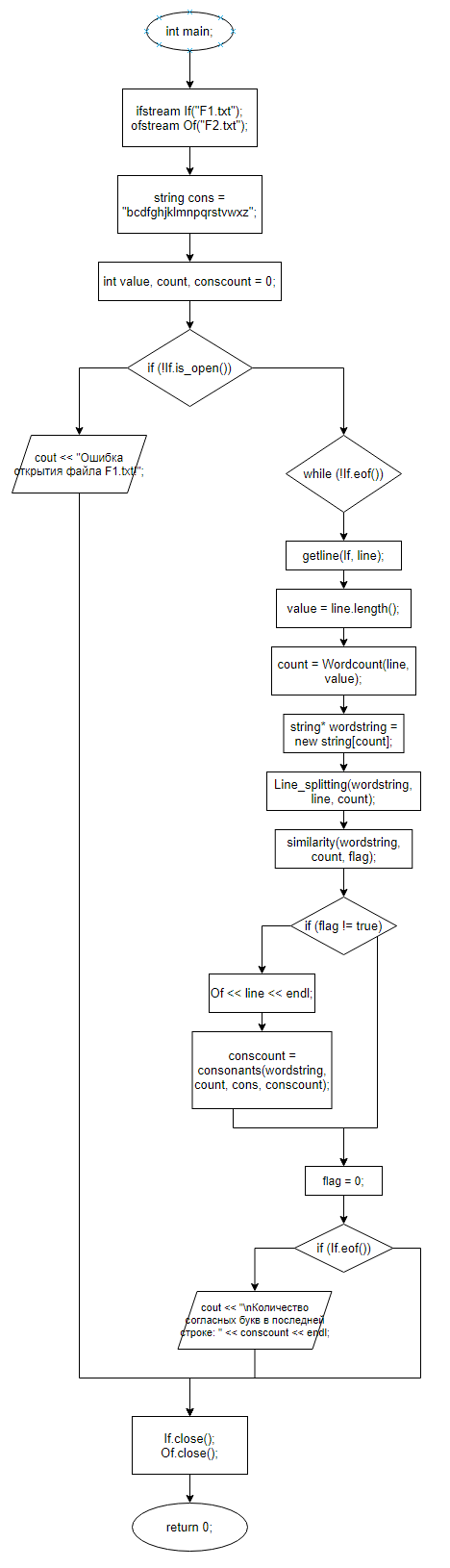
**Блок-схема**

****

****

****

****

****

**Код программы**

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

int Wordcount(string line, int value)

{

int count = 0;

for (int i = 1; i < value; i++)

{

if (line[i] == ' ' && line[i - 1] != ' ') //если символ в конце условного слова в строке - пробел и предыдущий символ - не пробел

{

count++;

}

}

if (line[value - 1] != ' ') //если предпоследний символ в строке не пробел, т.е. в строке только одно слово

{

count++;

}

return count;

}

void Line\_splitting(string\* wordstring, string line, int count)

{

int j = 0;

for (int i = 0; i < count; i++)

{

wordstring[i] = "";

while (line[j] == ' ')

{

j++;

}

while (line[j] != ' ' && line[j] != '\n' && line[j] != '\0' && line[j] != '(' && line[j] != ')' && line [j] != ',')

{

wordstring[i] += line[j++];

}

}

}

int similarity(string\* wordstring, int count, bool& flag)

{

for (int i = 0; i < count; i++)

{

if (wordstring[i+1] == wordstring[0])

{

flag = true;

i = count; //Выход из цикла без break'а

}

}

return flag;

}

int consonants(string\* wordstring, int count, string cons, int& conscount)

{

conscount = 0;

for (int i = 0; i < count; i++)

{

for (int j = 0; j < wordstring[i].length(); j++)

{

for (int l = 0; l < 20; l++)

{

if (wordstring[i][j] == cons[l])

{

conscount++;

}

}

}

}

return conscount;

}

int main()

{

setlocale(LC\_ALL, "ru");

ifstream If("F1.txt");

ofstream Of("F2.txt");

string line;

string cons = "bcdfghjklmnpqrstvwxz";

int value, count, conscount = 0;

bool flag;

if (!If.is\_open())

{

cout << "Ошибка открытия файла F1.txt!";

}

else

{

while (!If.eof())

{

getline(If, line);

value = line.length();

count = Wordcount(line, value); //количество слов в строке

string\* wordstring = new string[count];

Line\_splitting(wordstring, line, count);

similarity(wordstring, count, flag);

if (flag != true)

{

Of << line << endl;

conscount = consonants(wordstring, count, cons, conscount);

}

flag = 0; //сброс флага, если он стал true

}

if(If.eof())

{

cout << "\nКоличество согласных букв в последней строке: " << conscount << endl;

}

}

If.close();

Of.close();

return 0;

}