Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе № 9**

Дисциплина: «Информатика»

Тема: Структурное программирование. Файловые потоки

Вариант 21

Выполнил:

Студент группы РИС-20-1б

Ознобихин Елисей Андреевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

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

**Постановка задачи**

(Вариант 21)

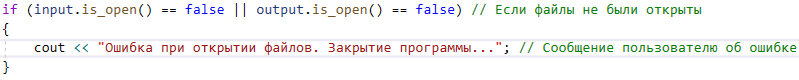
1) Скопировать из файла F1 в файл F2 все строки, в которых есть слова, совпадающие с первым словом.

2) Определить количество согласных букв в последней строке файла F2.

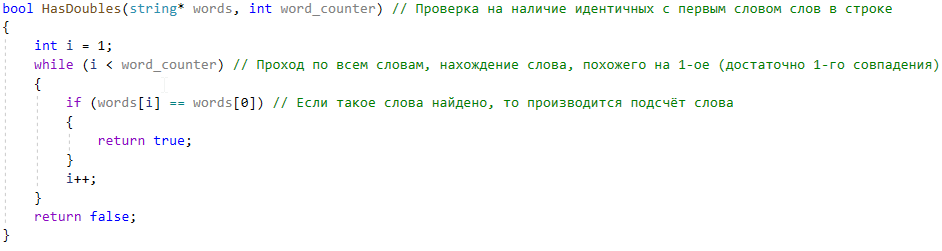
**Анализ задачи**

**1.** Для решения задачи необходимо:

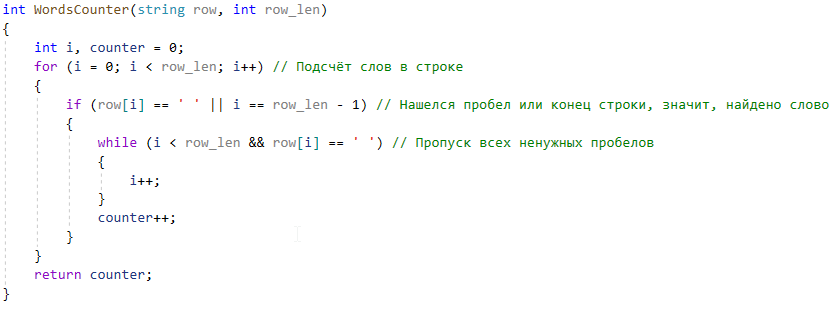
**1.1.** Проверить, существуют ли файлы ввода и вывода и в случае их отсутствия сообщить об ошибке.



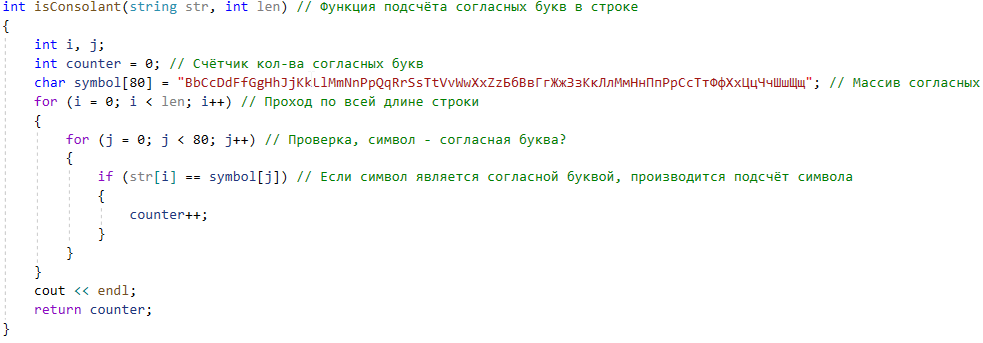
**1.2.** Разработать булевую функцию проверки строки на наличие слов, идентичных первому слову в считываемой из файла ввода строке. Проверка строки ведется попарным сравниванием каждого слова в строке с первым.



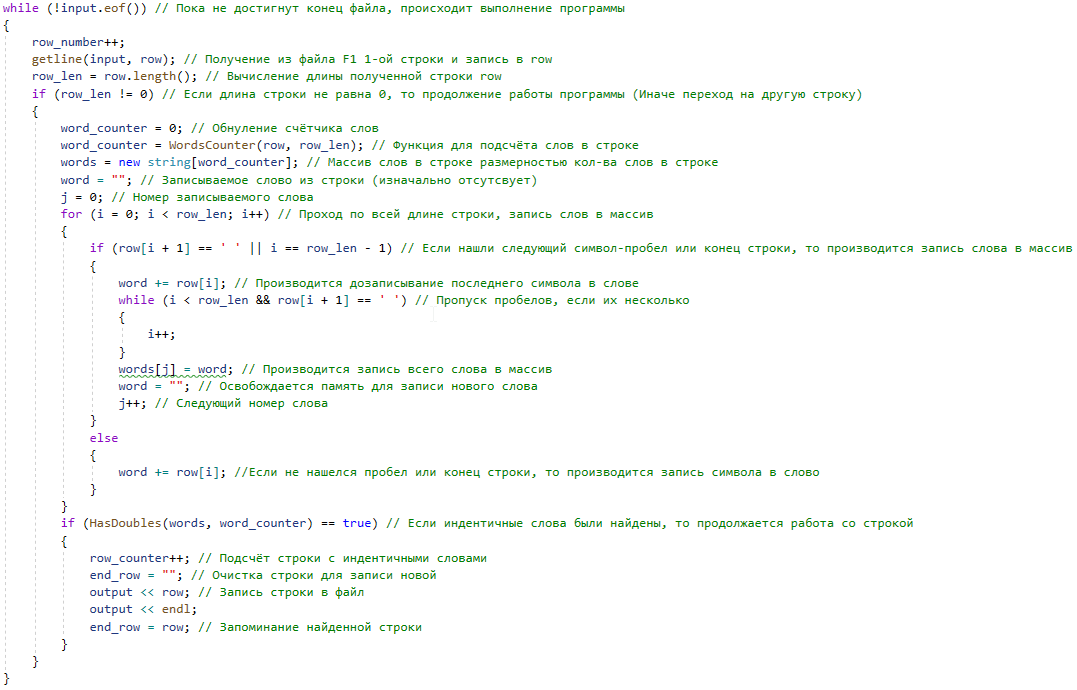
**1.3.** Разработать функцию WordsCounter, чтобы найти и подсчитать количество слов в считываемой из файла ввода строке.



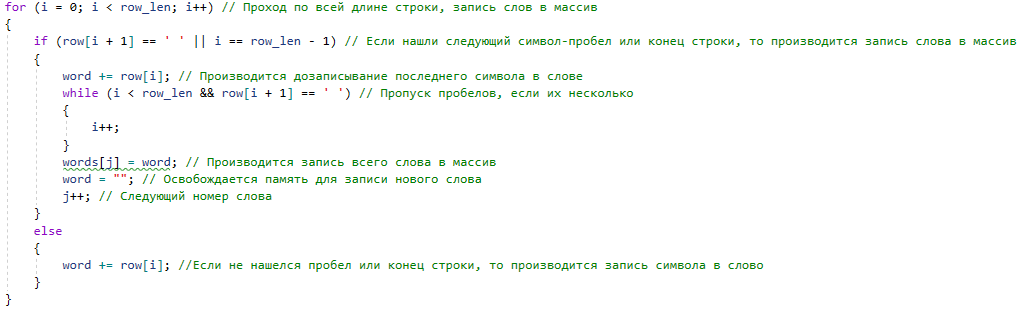
**1.4.** Разработать функцию для проверки каждого символа в строке и подсчёта количества найденных согласных букв в строке.



**1.5.** Реализовать применение этих функций для каждой строки файла ввода F1.txt.



**1.6.** Реализовать получение слов из каждой строки файла ввода.



**1.7.** Реализовать поиск и запись последней строки из файла F2.txt.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_KyDznNWvMP.png

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

**2.1.** Класс ifstream для получения данных из файла:

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_eyiw21MtVs.png

**2.2.** Класс ofstream для записи данных в файл:

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_lklgmwJ5CG.png

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

1. Тип string: рассматриваемая строка, полученная из файла ввода.

2. Тип int: длина этой строки.

Сама функция имеет тип int для подсчёта количества слов в строке.

**C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_tw4HRBtk4h.png**

**2.4.** Для функции HasDoubles используются следующие аргументы:

1. Указатель на string: массив из string, который хранит записанные слова из рассматриваемой строки.

2. Тип int: кол-во слов в рассматриваемой строке.

Сама функция имеет тип bool для передачи информации о том, было ли найдено хотя бы одно слово, идентичное первому слову в строке.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_oaRrtWJC3A.png

**2.5.** Для функции isConsolant используются следующие типы данных:

1. Тип string: рассматриваемая строка.

2. Тип int: длина этой строки.

Сама функция имеет тип int для возвращения количества согласных букв в строке.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_ajYf5MmlQW.png

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

**3.1.** Для обработки строк из файла данные были 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.txt реализован с помощью функции getline.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_ckuFj4oARA.png

**4.2.** Запись данных в файл реализован с помощью оператора вывода в файл.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_SsB1yWZB5g.png

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

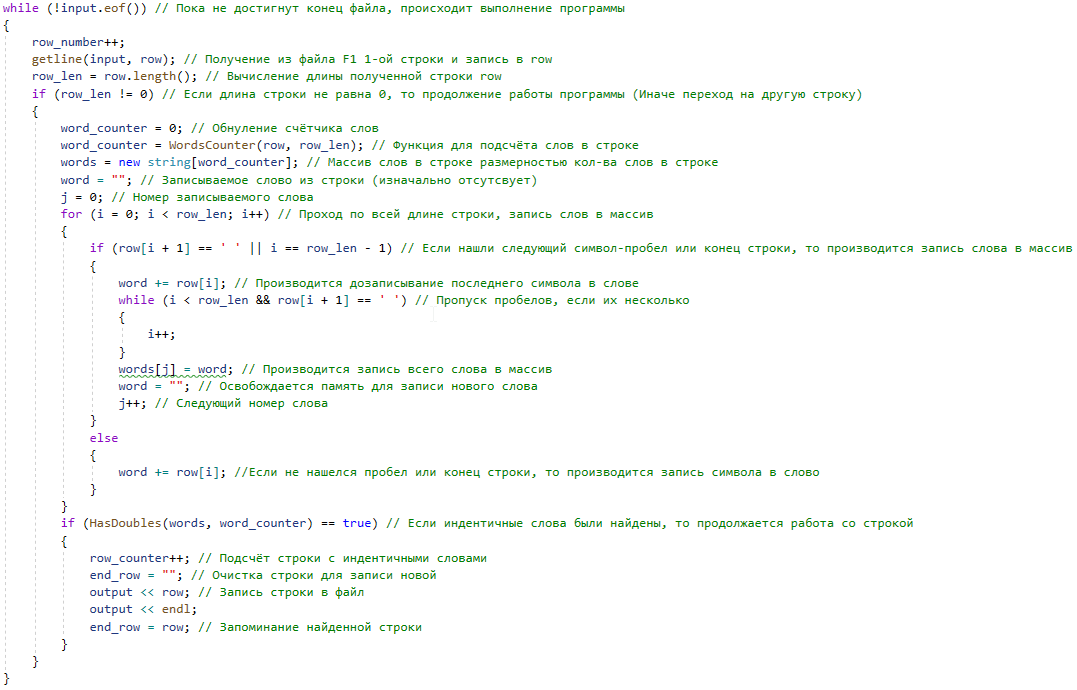
C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_bZeqNxtVYX.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_GBHH35AoXe.png



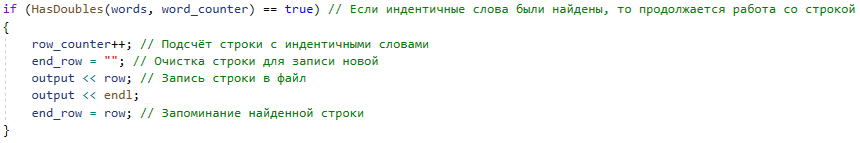
**5.** Поставленные задачи решены следующими действиями:

**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.

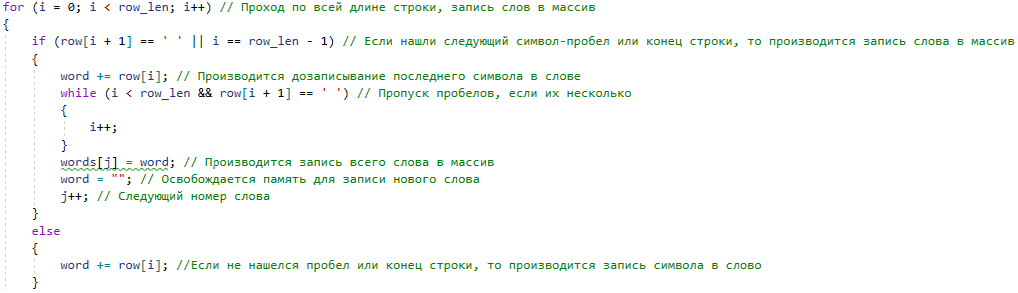


**5.2.** Поиск и копирование строк из файла F1 в F2 выполнены с помощью функций HasDoubles и WordCounter и реализации записи найденных в строке слов через условный оператор.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_t4V2QasPHD.png



**5.3.** Нахождение нужной строки по критерию поиска реализовано с помощью строки ,массива для записи слов и цикла for.

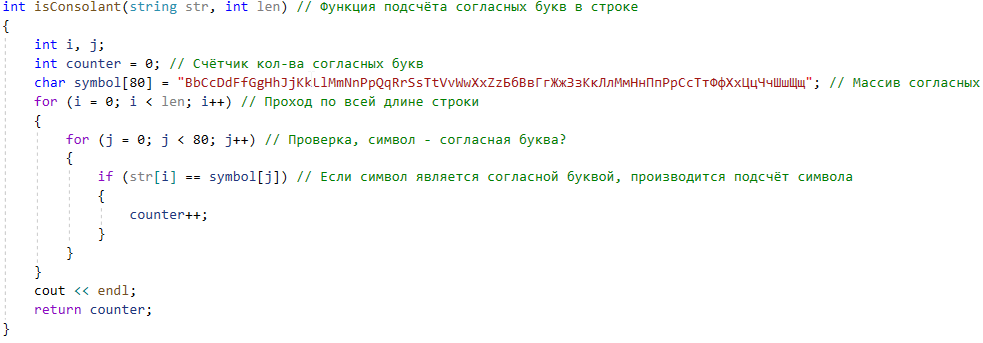


**5.4.** Поискпоследней найденной и соответствующей критериям поиска строки из файла F1 реализован через сохранение рассматриваемой строки в переменную типа string. Строка должна подходить по критериям отбора строк.

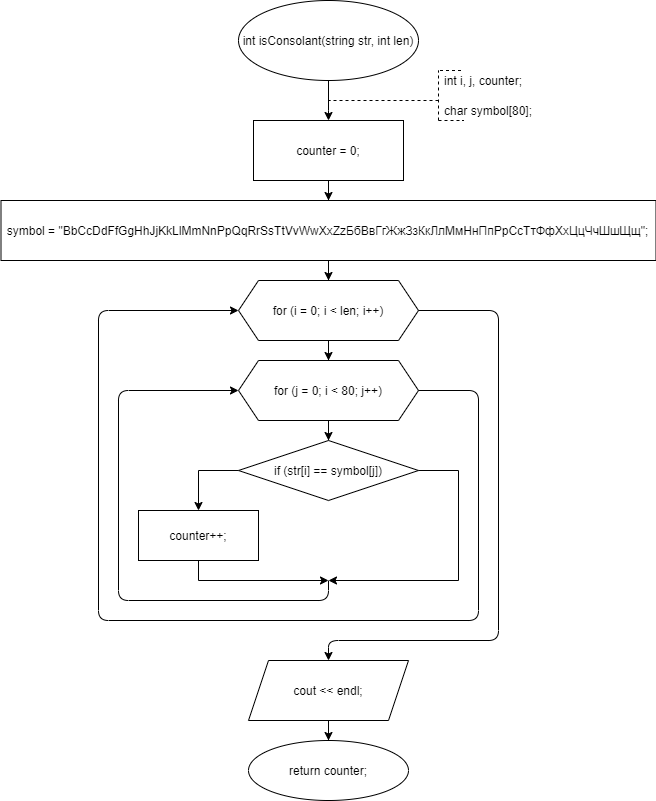
C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_dotxNp0A1C.png

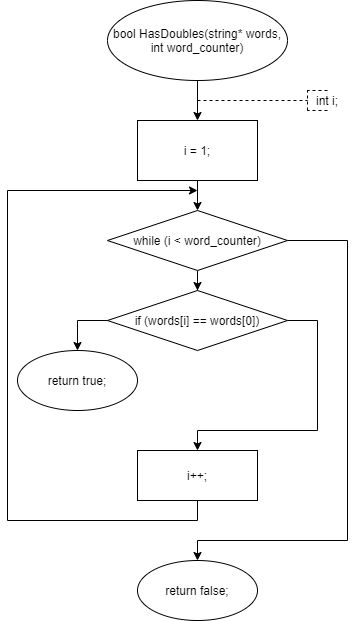


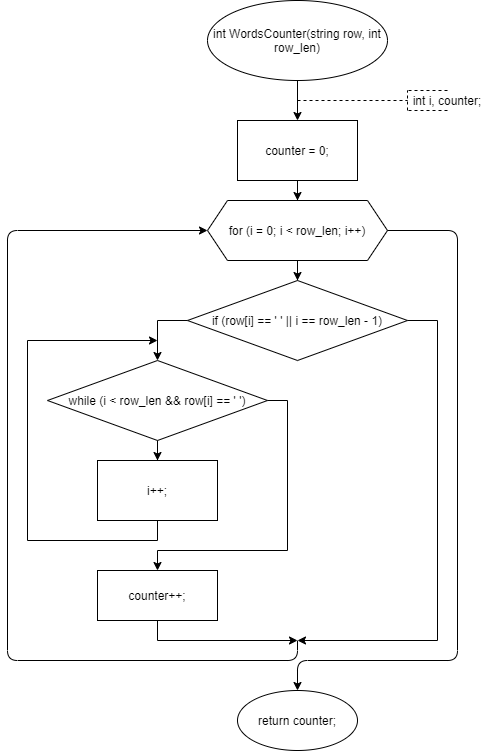
**5.5.** Подсчёт количества согласных букв в последней строке файла F2 реализован с помощью функции isConsolant.

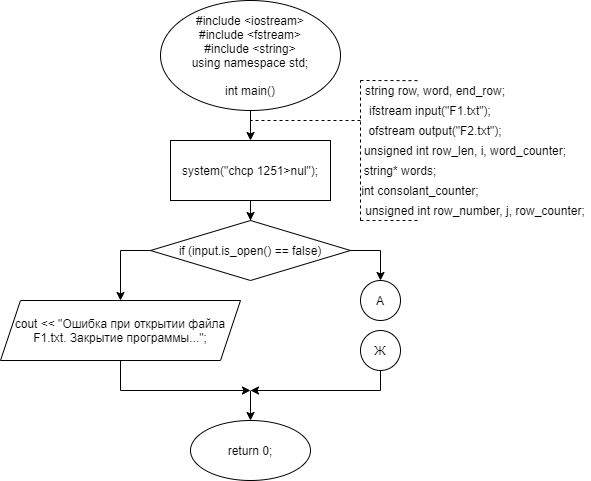


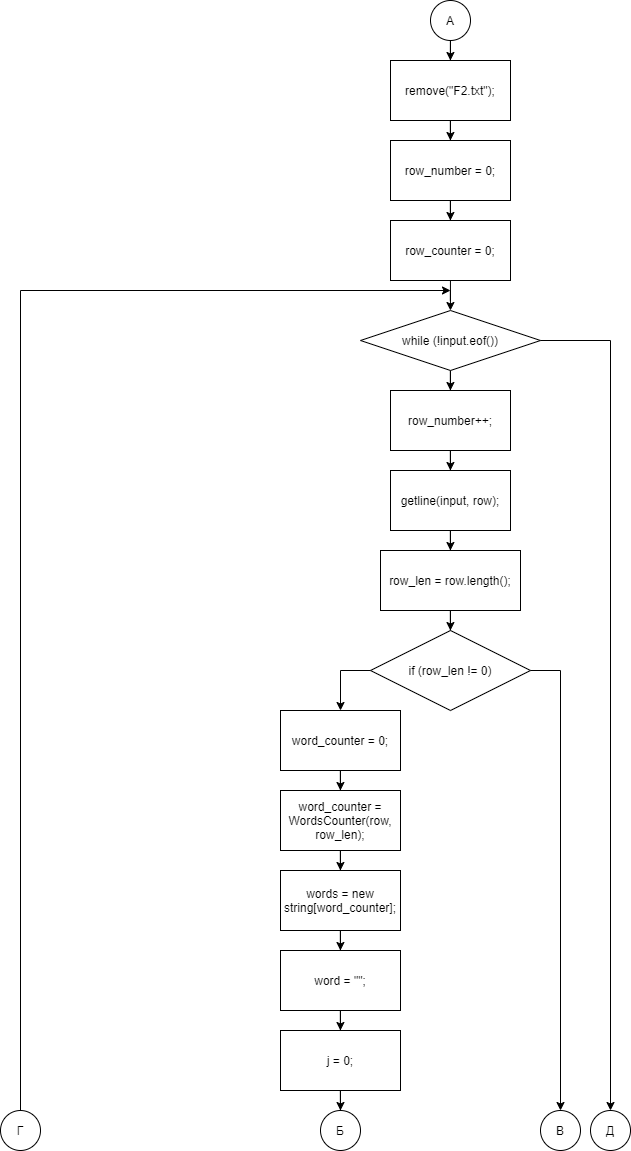
**Блок-схема программы**

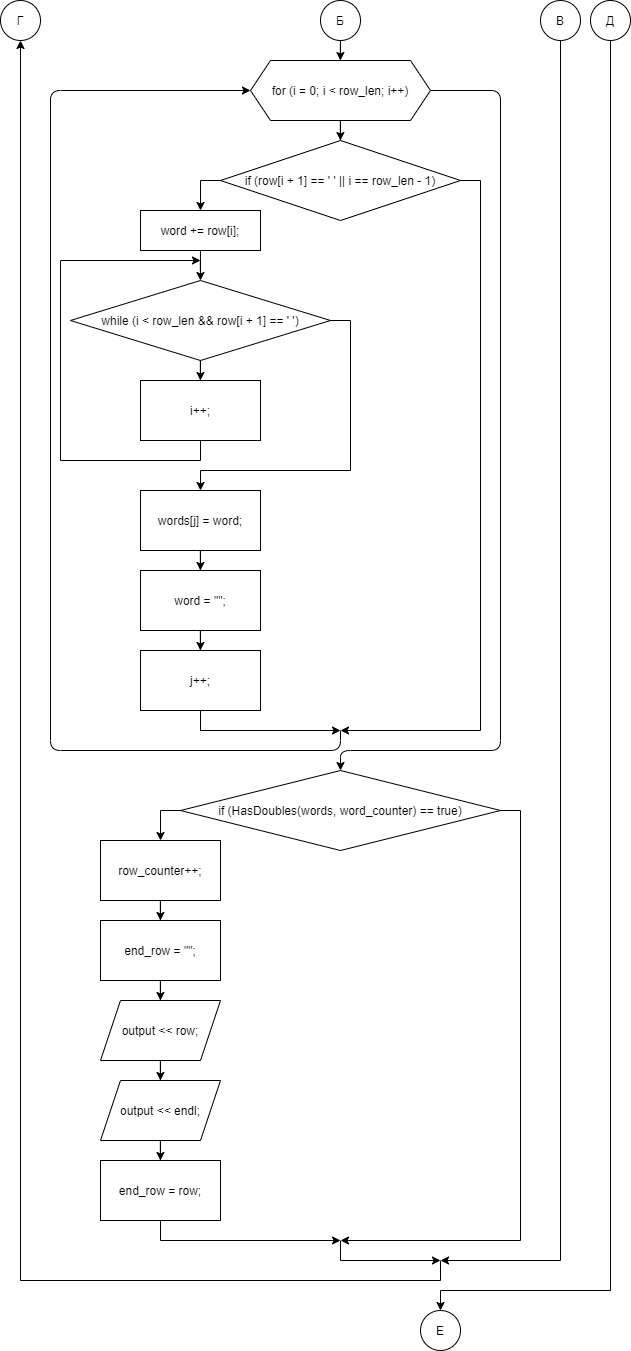
****

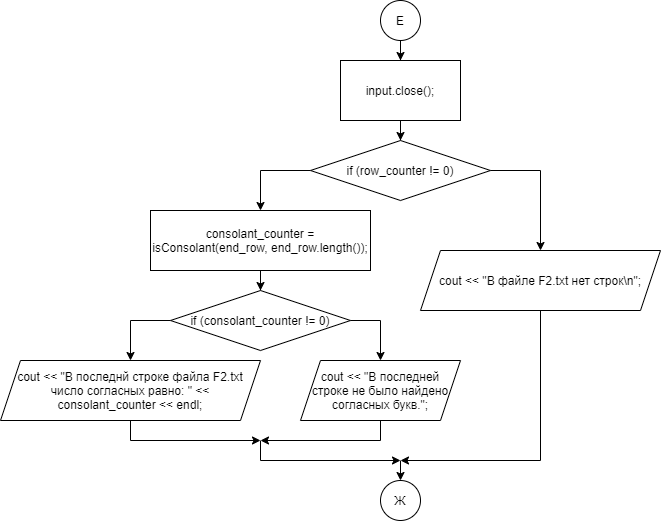
****

****

****

****

****

****

**Реализация задачи на языке С++**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int isConsolant(string str, int len)

{

int i, j;

int counter = 0;

char symbol[80] = "BbCcDdFfGgHhJjKkLlMmNnPpQqRrSsTtVvWwXxZzБбВвГгЖжЗзКкЛлМмНнПпРрСсТтФфХхЦцЧчШшЩщ";

for (i = 0; i < len; i++)

{

for (j = 0; j < 80; j++)

{

if (str[i] == symbol[j])

{

counter++;

}

}

}

cout << endl;

return counter;

}

bool HasDoubles(string\* words, int word\_counter)

{

int i = 1;

while (i < word\_counter)

{

if (words[i] == words[0])

{

return true;

}

i++;

}

return false;

}

int WordsCounter(string row, int row\_len)

{

int i, counter = 0;

for (i = 0; i < row\_len; i++)

{

if (row[i] == ' ' || i == row\_len - 1)

{

while (i < row\_len && row[i] == ' ')

{

i++;

}

counter++;

}

}

return counter;

}

int main()

{

system("chcp 1251>nul");

string row;

ifstream input("F1.txt");

if (input.is\_open() == false)

{

cout << "Ошибка при открытии файла F1.txt. Закрытие программы...";

exit(1); }

ofstream output("F2.txt");

if (output.is\_open() == false)

{

cout << "Ошибка при открытии файла F2.txt. Закрытие программы...";

exit(2);

}

remove("F2.txt");

unsigned int row\_len, i, word\_counter

unsigned int row\_number, j, row\_counter;

string\* words;

string word, end\_row;

row\_number = 0;

row\_counter = 0;

while (!input.eof())

{

row\_number++;

getline(input, row);

row\_len = row.length();

if (row\_len != 0)

{

word\_counter = 0;

word\_counter = WordsCounter(row, row\_len);

words = new string[word\_counter];

word = "";

j = 0;

for (i = 0; i < row\_len; i++)

{

if (row[i + 1] == ' ' || i == row\_len - 1)

{

word += row[i];

while (i < row\_len && row[i + 1] == ' ')

{

i++;

}

words[j] = word;

word = "";

j++;

}

else

{

word += row[i];

}

}

if (HasDoubles(words, word\_counter) == true)

{

row\_counter++;

end\_row = "";

output << row;

output << endl;

end\_row = row;

}

}

}

input.close();

if (row\_counter != 0)

{

int consolant\_counter;

consolant\_counter = isConsolant(end\_row, end\_row.length());

if (consolant\_counter != 0)

{

cout << "В последнй строке файла F2.txt число согласных равно: " << consolant\_counter << endl;

}

else

{

cout << "В последней строке не было найдено согласных букв.";

}

}

else

{

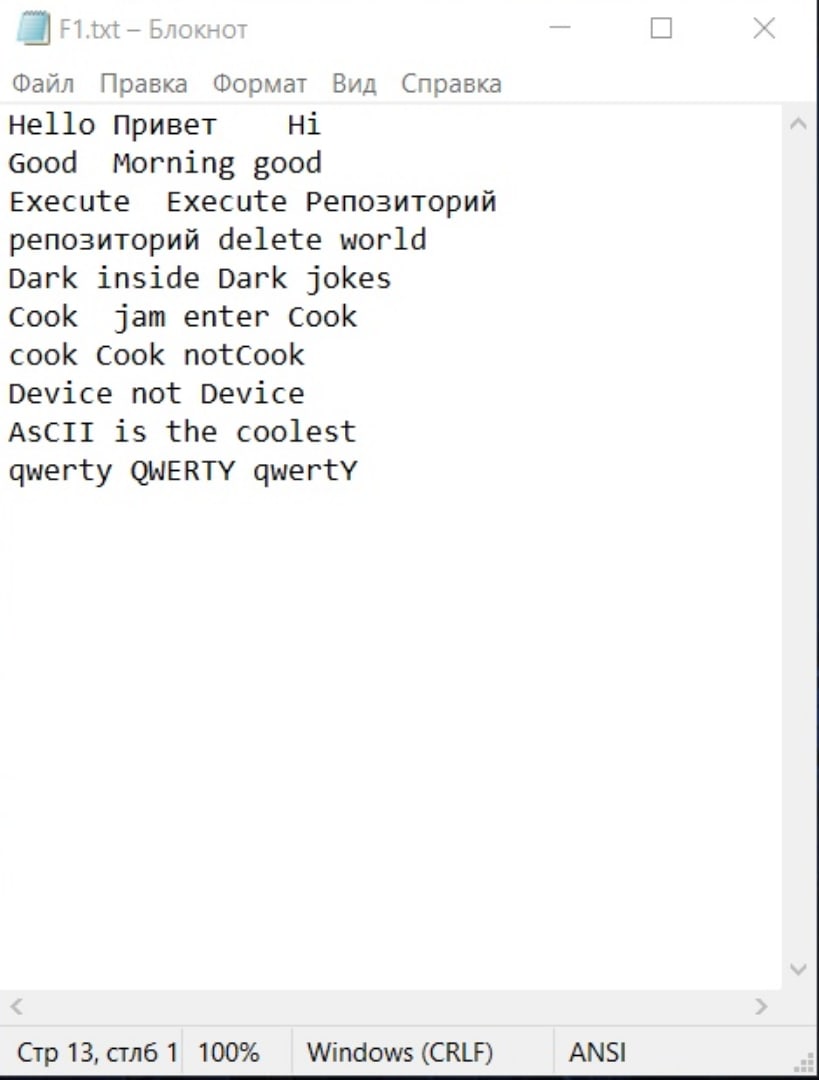
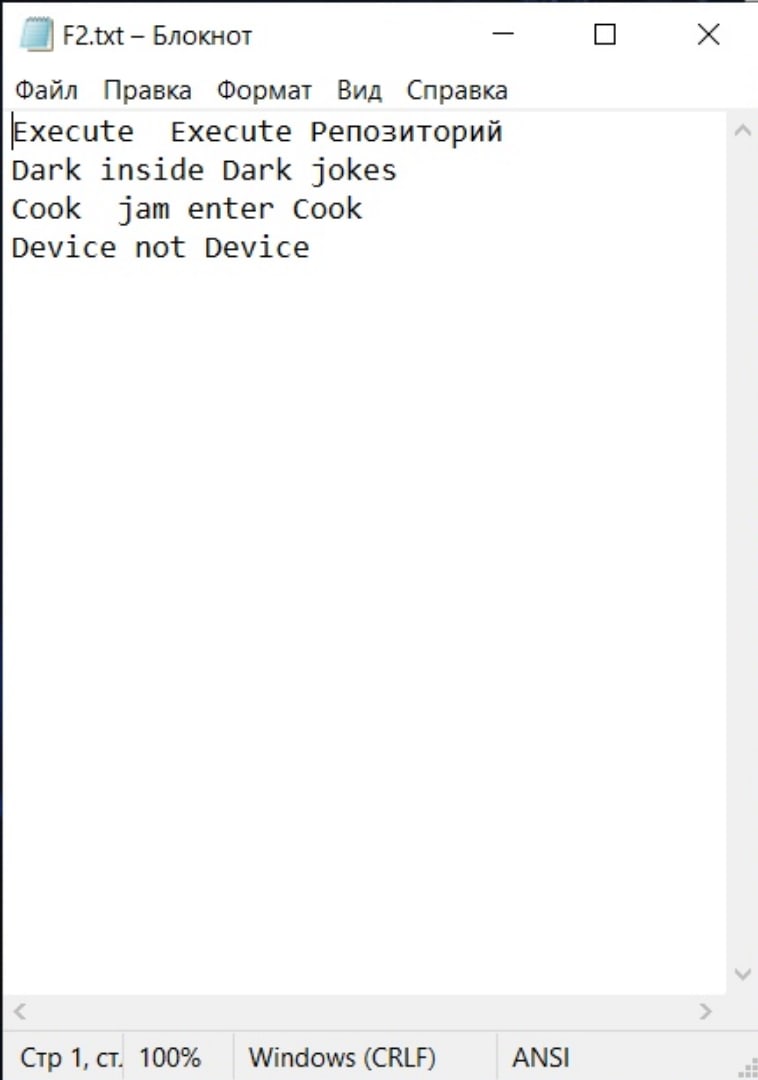
cout << "В файле F2.txt нет строк\n" ;

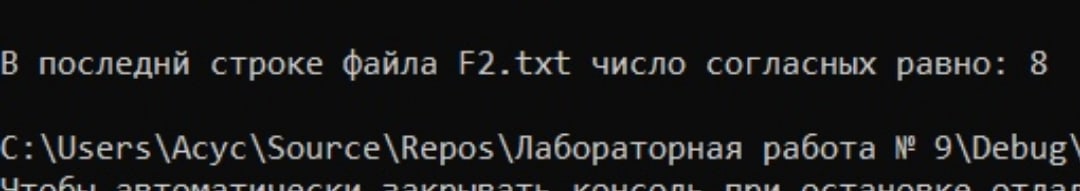
}

output.close();

}

**Скриншоты результатов работы программы**

****  ****

****