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 РИС-20-1б

Ремянников А. В.

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**

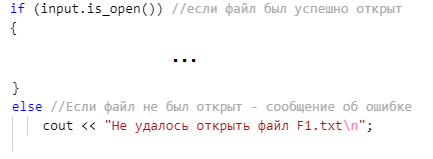
(Вариант 25)

1. Скопировать из файла F1 в файл F2 все строки, в которых содержится не менее двух одинаковых слов.
2. Определить номер слова, в котором больше всего цифр.

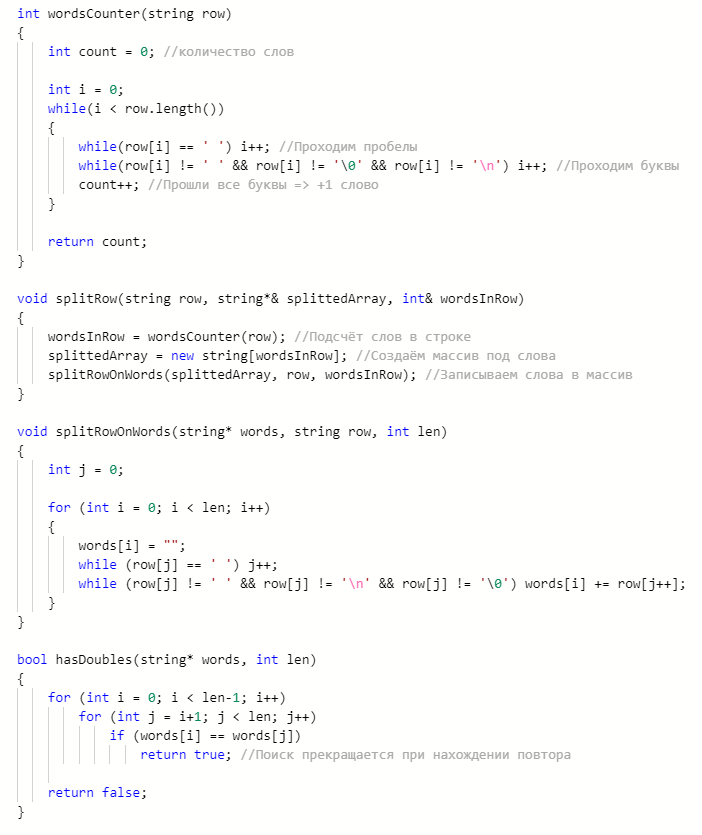
**Анализ задачи**

**1.** Для решения задачи необходимо:

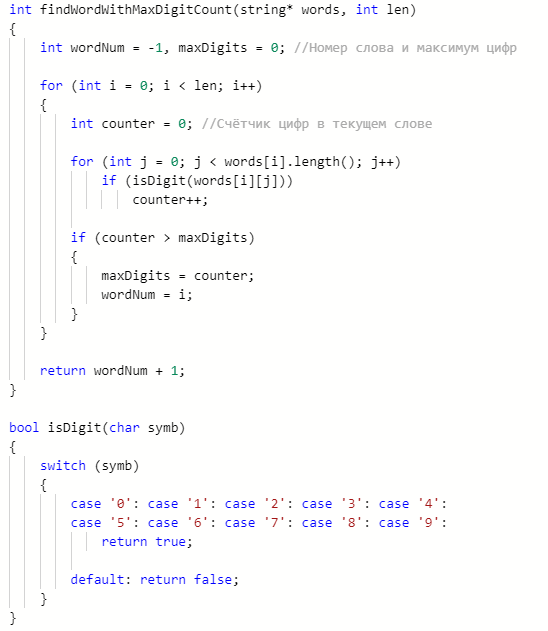
**1.1.** Проверить, существует ли файл ввода и в случае его отсутствия сообщить об ошибке.



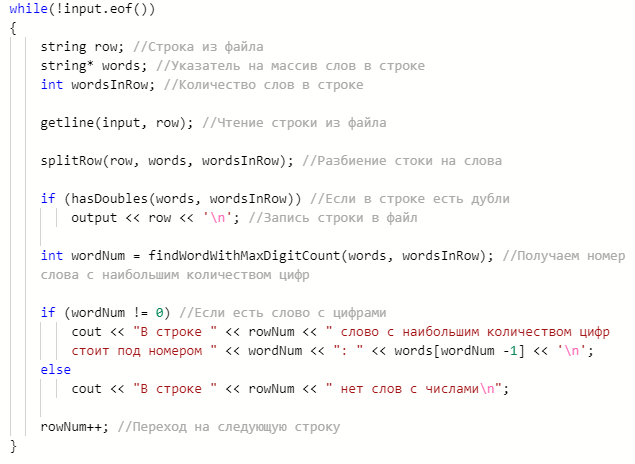
**1.2.** Разработать функцию для проверки строки на наличие слов-дубликатов, выделяющую слова из строки и заносящую их в созданный массив из string- splitRow, использующую вспомогательные функции splitRowOnWords (функция, которая непосредственно разделяет строку на слова и заполняет массив) и wordsCounter (функция для подсчета слов в строке). Проверка строки проводится попарным сравниванием слов друг с другом в функции hasDoubles.



**1.3.** Разработать функцию findWordWithMaxDigitCount, чтобы определить номер слова с наибольшим количеством цифр и фунцию isDigit проверяющую, является ли символ цифрой.



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



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

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

****

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



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

1. Тип string: Входная строка, которая будет разделена на слова.

2. Ссылка на указатель на string: Массив из string, полученный в результате работы функции.

3. Ссылка на int: Количество слов в строке.

Сама функция имеет тип void, так как в ходе ее работы нет необходимости возвращать значение для этой функции.



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

1. Тип string: Входная строка, в которой необходимо найти количество слов.

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



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

1. Указатель на string: Массив со словами.

2. Тип string: Входная строка, которая будет разбита на слова

3. Тип int: Количество слов в строке.

Сама функция имеет тип void, так как в ходе ее работы нет необходимости возвращать значение для этой функции.



**2.6.** Для функции findWordWithMaxDigitCount используются следующие аргументы:

1.Указатель на string: Массив слов.

2. Тип int: Количество слов в массиве.

Сама функция имеет тип int для возвращения позиции слова с наибольшим количеством цифр в составе.



**2.7.** Для функции isDigit используются следующие аргументы:

1. Тип char: Входной символ.

Сама функция имеет тип bool для передачи информации о том, является ли символ цифрой.



**2.8.** Для функции hasDoubles используются следующие аргументы:

1. Указатель на string: Массив слов.

2. Тип int: Количество слов в массиве.



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

**3.1.** Для обработки строк из файла данные были записанны 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.



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



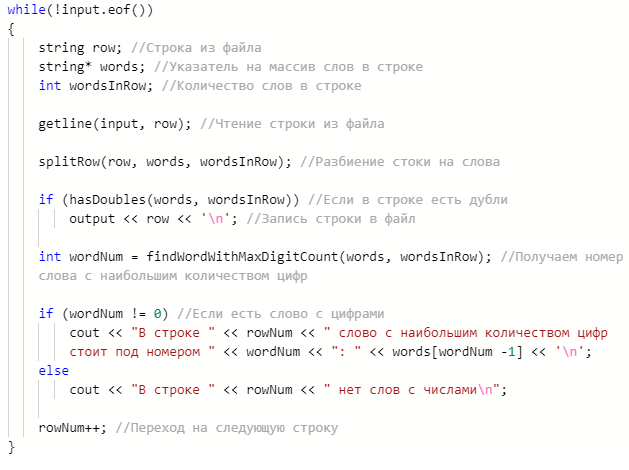


**4.3.** Вывод данных в файл реализован с помощью оператора вывода в файл.

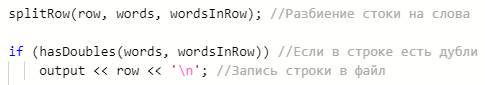


**5.** Поставленные задачи будут решены следующими действиями:

**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.



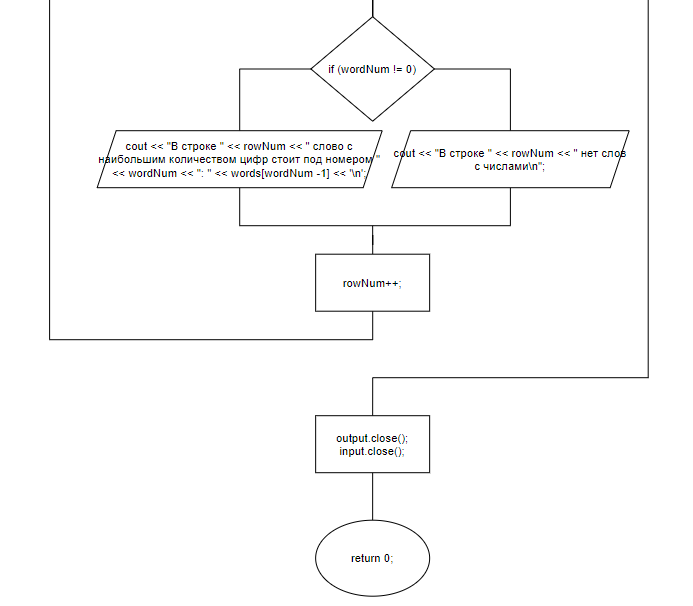
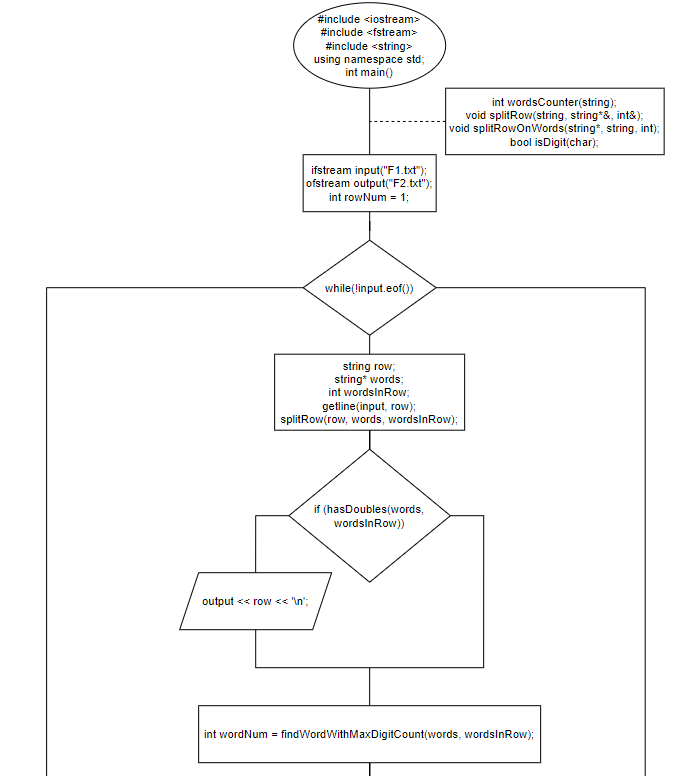
**5.2.** Поиск и копирование строк из файла F1 в F2 реализованы с помощью функций splitRow и hasDoubles.

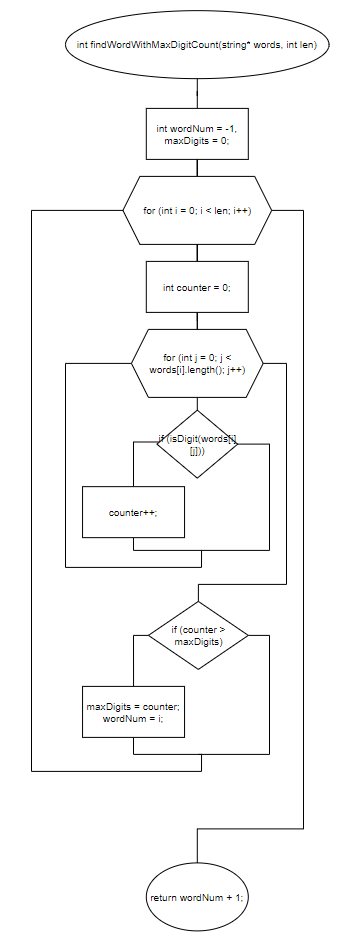
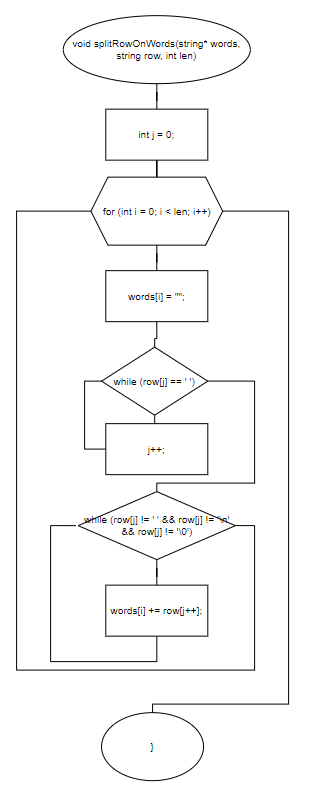
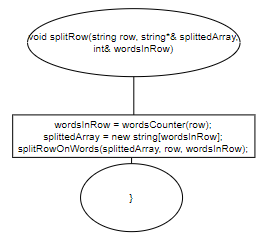
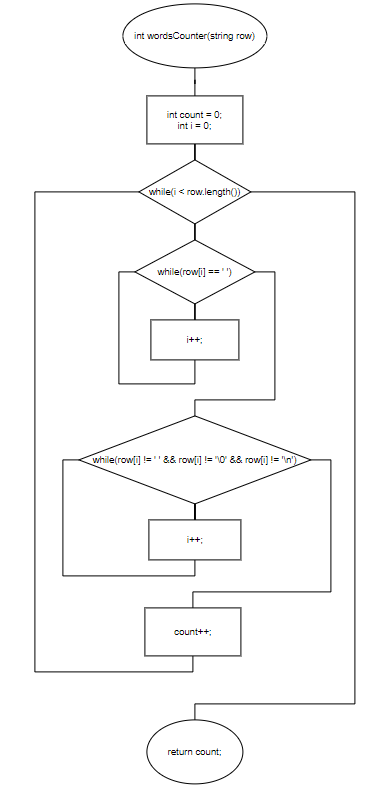
****

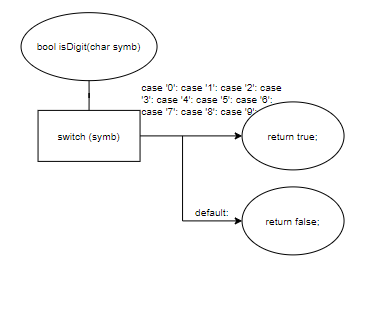
**5.3.** Определение позиции слова с наибольшим количеством цифр в составе реализовано с использованием функции findWordWithMaxDigitCount

****

**Блок-схема программы**







**Решение**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int wordsCounter(string); //Подсчёт слов в строке

void splitRow(string, string\*&, int&); //Разбиение строки по словам, нахождение их количества

void splitRowOnWords(string\*, string, int); //Разбиение строки по словам, запись в массив

bool hasDoubles(string\*, int); //Есть ли в строке дубли

int findWordWithMaxDigitCount(string\*, int); //Поиск слова с максимальным количеством цифр

bool isDigit(char); //Проверка является ли символ цифрой

int main()

{

ifstream input("F1.txt"); //Открываем файл с исходными данными

if (input.is\_open()) //если файл был открыт

{

ofstream output("F2.txt"); //открываем файл для записи

int rowNum = 1; //Номер обрабатываемой строки

while(!input.eof())

{

string row; //Строка из файла

string\* words; //Массив слов в строке

int wordsInRow; //Количество слов в строке

getline(input, row); //Чтение строки из файла

splitRow(row, words, wordsInRow); //Разбиение стоки на слова

if (hasDoubles(words, wordsInRow)) //Если в строке есть дубли

output << row << '\n'; //Запись строки в файл

int wordNum = findWordWithMaxDigitCount(words, wordsInRow); //Получаем номер слова с наибольшим количеством цифр

if (wordNum != 0) //Если есть слово с цифрами

cout << "В строке " << rowNum << " слово с наибольшим количеством цифр стоит под номером " << wordNum << ": " << words[wordNum -1] << '\n';

else

cout << "В строке " << rowNum << " нет слов с числами\n";

rowNum++; //Переход на следующую строку

}

output.close(); //Закрываем файл для вывода

}

else //Если файл не был открыт - сообщение об ошибке

cout << "Не удалось открыть файл F1.txt\n";

input.close(); //Закрываем файл с исходными данными

}

int wordsCounter(string row)

{

int count = 0; //количество слов

int i = 0;

while(i < row.length())

{

while(row[i] == ' ') i++; //Проходим пробелы

while(row[i] != ' ' && row[i] != '\0' && row[i] != '\n') i++; //Проходим буквы

count++; //Прошли все буквы => +1 слово

}

return count;

}

void splitRow(string row, string\*& splittedArray, int& wordsInRow)

{

wordsInRow = wordsCounter(row); //Подсчёт слов в строке

splittedArray = new string[wordsInRow]; //Создаём массив под слова

splitRowOnWords(splittedArray, row, wordsInRow); //Записываем слова в массив

}

void splitRowOnWords(string\* words, string row, int len)

{

int j = 0;

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

{

words[i] = "";

while (row[j] == ' ') j++;

while (row[j] != ' ' && row[j] != '\n' && row[j] != '\0') words[i] += row[j++];

}

}

bool hasDoubles(string\* words, int len)

{

for (int i = 0; i < len-1; i++)

for (int j = i+1; j < len; j++)

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

return true; //Поиск прекращается при нахождении повтора

return false;

}

int findWordWithMaxDigitCount(string\* words, int len)

{

int wordNum = -1, maxDigits = 0; //Номер слова и максимум цифр

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

{

int counter = 0; //Счётчик цифр в текущем слове

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

if (isDigit(words[i][j]))

counter++;

if (counter > maxDigits)

{

maxDigits = counter;

wordNum = i;

}

}

return wordNum + 1;

}

bool isDigit(char symb)

{

switch (symb)

{

case '0': case '1': case '2': case '3': case '4':

case '5': case '6': case '7': case '8': case '9':

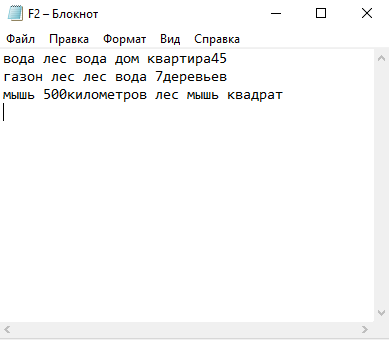
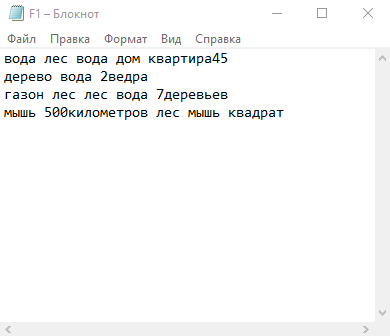
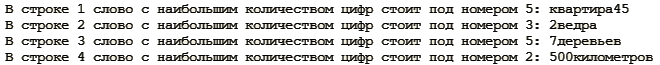
return true;

default: return false;

}

}

**Скриншоты результатов работы программы**

**** ****