Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.01 – «Информатика и вычислительная техника»

**Лабораторная работа №9**

**“Строковый ввод-вывод”**

**Вариант 25.**

|  |
| --- |
| Выполнила студентка гр. ИВТ-24-2б  Косиненко Ксения Николаевна \_\_\_\_\_\_ |
| Проверил:  Доц. Каф. ИТАС\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Ольга Андреевна Полякова\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка) (подпись)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) |

г. Пермь, 2025

# Постановка задачи

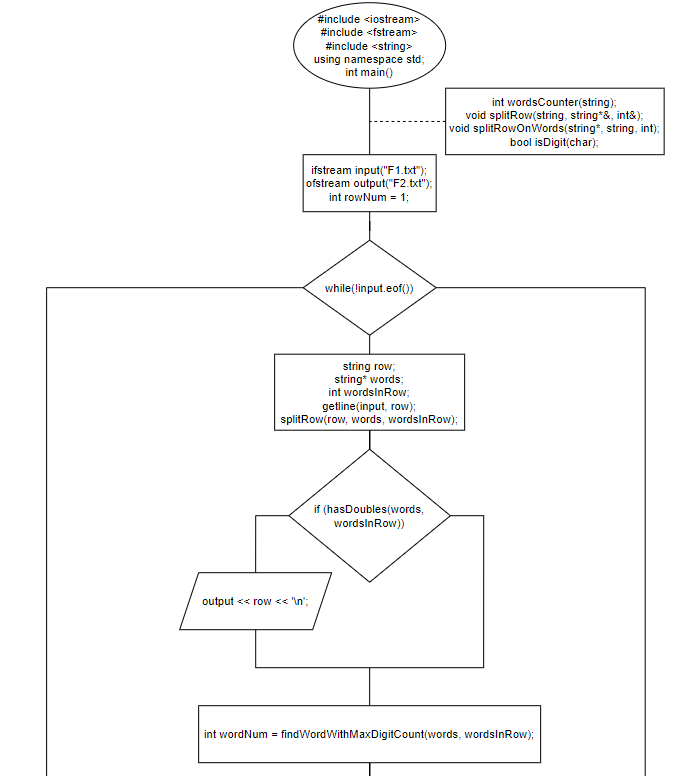
Создать текстовый файл F1 не менее, чем из 10 строк и записать в него информацию

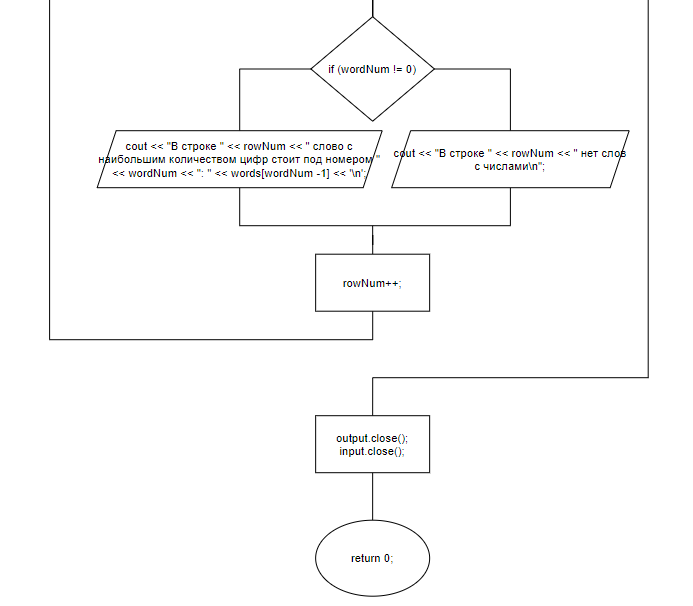
Выполнить задание:

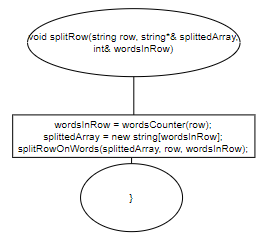
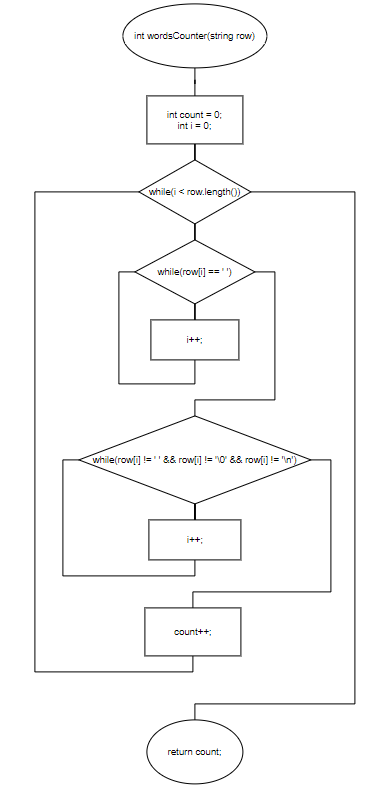
1) Скопировать из файла F1 в файл F2 все строки, в которых содержится не менее двух одинаковых слов.

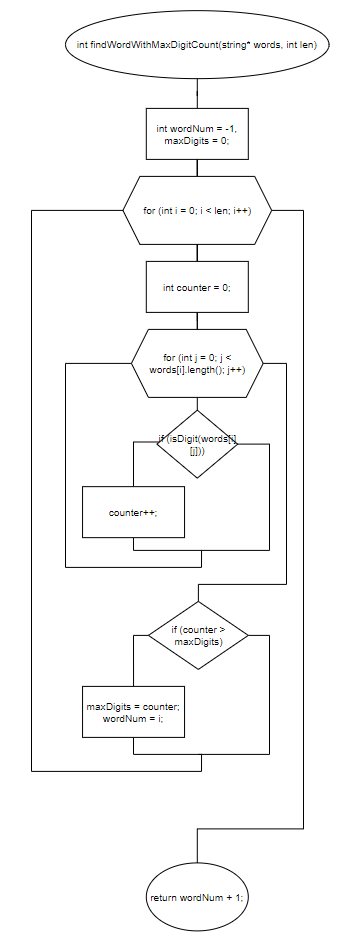
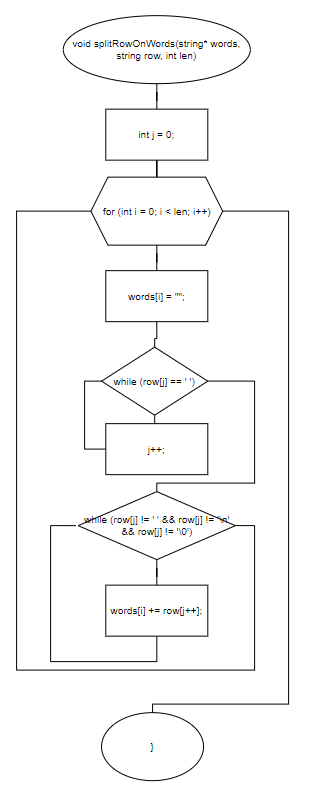
2) Определить номер слова, в котором больше всего цифр.

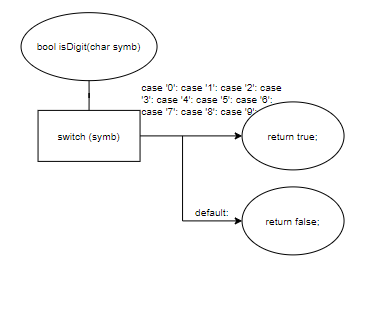
**Блок-схема программы**











# Код программы

#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

void solution()

{

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':

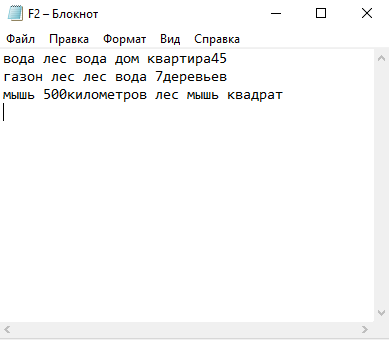
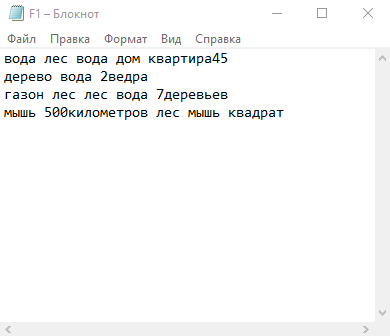
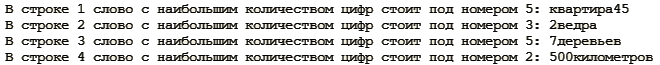
return true;

default: return false;

}

# }

# Результат

**** ****

**GitHub**

<https://github.com/kitmammaw/->

