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: основы алгоритмизации и программирования

Тема: Файлы

Вариант 24

Выполнила:

Студентка группы ИВТ-20-2б

Ананина Арина Юрьевна

Проверил:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2020**

1. Постановка задачи

Создать текстовый файл F1 не менее, чем из 10 строк и записать в него информацию. Скопировать из файла F1 в файл F2 все строки, в которых содержится два одинаковых слова. Определить номер слова, в котором больше всего букв «А».

2. Анализ задачи

1. Какие переменные и типы данных будут использованы:

* Будет использован массив строк splited, для хранения слов из строки.
* За количество строк в файле F1 отвечает переменная row типа int.
* За строку из файла F1 отвечает переменная n типа string.
* Количество слов в строке запишется в переменную wordInN типа int.
* Переменная answ типа bool показывает, есть ли в строке два одинаковых слова.

2.Переменные типа int используются как счетчики и нужны для подсчёта кол-ва элементов массива, слов в строке, строк в файле и т.д.

Каждая новая строка из F1 записывается в переменную n с помощью функции getline.

Переменной answ присваивается значение функции isHaveDuble, которая отвечает за наличие в строке двух одинаковых слов.

3. В задачи используются файлы. В файлы записываются данные в виде строк.

4. Ввод и вывод будут организованы внутри файлов. А сообщения о словах, в которых будет найдено наибольшее количество букв «а» будут выводиться в консоль.

cout << "В строке под номером " << row << "слово, в котором больше всего букв 'а', под номером " << maxAIndex + 1 << ": " << splited[maxAIndex] << endl;

cout << "В строке под номером " << row << " нет слов с буквой 'а'" << endl;

5. С помощью функции splitedOn текущая строка разделяется на слова, и они записываются в массив splited.

С помощью цикла просматривается каждый символ строки и записывается в элемент массива, если встречается пробел или конец строки, то индекс массива становится на 1 больше , и запись начинается в следующий элемент массива.

int splitedOn(string n, string\*& split) // функция разбивает строку на слова и записывает их в массив

{

int curIndex = 0;

for (int i = 0; i < n.size(); i++)

{

if (n[i] == ' ' || n[i] == '\n' || n[i] == '\0')

{

curIndex++;

continue;

}

split[curIndex] += n[i];

}

return curIndex;

}

С помощью функции isHaveDuble можно понять, есть ли в строке ровно два одинаковых слова, если результат функции true, то данная строка записывается в файл F2.

В функции начинается проход по массиву из слов в строке, создаётся новый массив, в который записываются слова, которые уже были проверены на дубли.

string\* watched = new string[curIndex + 1];

Таким образом программа не будет проверять на дубли уже проверенные слова. Далее с помощью вложенного цикла проверяются все слова на дубли, если они есть, то значение флажка становится равно 1 и функция возвращает значение true, иначе функция возвращает значение false.

f = 0;

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

{

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

{

f++;

}

}

if (f == 1)

{

return true;

}

Это значение присваивается переменной answ и уже с помощью оператора if, в зависимости от значения переменной в файл F2 заносится строка.

if (answ)

{

F2 << n << endl;

}

Функция findWord находит слова в строке с наибольшим кол-вом букв «а» и выводит сообщение об этом.

С помощью цикла осуществляется проход по массиву из слов в строке, далее с помощью вложенного цикла осуществляется проход по каждому символу слова и ищется слово с максимальным кол-вом искомой буквы.

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

{

int a = 0;

for (int j = 0; j < size(splited[i]); j++)

{

if (splited[i][j] == 'а')

{

a++;

}

}

if (maxA < a) {

maxA = a;

maxAIndex = i;

}

}

Если переменная, отвечающая за максимальное количество символов, была изменена, то это говорит о наличии в строке хотя бы одного слова с нужным символом и выводится сообщение о номере слова, в котором больше всего букв «а». Иначе выводится сообщение о том, что в строке нет слов с искомой буквой.

if (maxA > 0)

{

cout << "В строке под номером " << row << "слово, в котором больше всего букв 'а', под номером " << maxAIndex + 1 << ": " << splited[maxAIndex] << endl;

}

else

{

cout << "В строке под номером " << row << " нет слов с буквой 'а'" << endl;

}

3. Код программы

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

void findWord(string\*& splited, int curIndex, int row) // функция находит слова с наибольшим количеством букв «а»

{

int maxA = 0;

int maxAIndex = 0;

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

{

int a = 0;

for (int j = 0; j < size(splited[i]); j++) {

if (splited[i][j] == 'а') {

a++;

}

}

if (maxA < a) {

maxA = a;

maxAIndex = i;

}

}

if (maxA > 0) {

cout << "В строке под номером " << row << "слово, в котором больше всего букв 'а', под номером " << maxAIndex + 1 << ": " << splited[maxAIndex] << endl;

}

else {

cout << "В строке под номером " << row << " нет слов с буквой 'а'" << endl;

}

}

bool isHaveDuble(int curIndex, string \*&splited ) // функция определяет, есть ли в строке два одинаковых слова

{

int f;

int watchedIndex = 0;

string\* watched = new string[curIndex + 1];

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

{

int pos = -1;

for (int k = 0; k < watchedIndex; k++) {

if (watched[k] == splited[i]) {

pos = k;

}

}

if (pos != -1)

{

continue;

}

watched[watchedIndex++] = splited[i];

f = 0;

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

{

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

{

f++;

}

}

if (f == 1)

{

return true;

}

}

return false;

}

int splitedOn(string n, string\*& split) // функция разбивает строку на слова и записывает их в массив

{

int curIndex = 0;

for (int i = 0; i < n.size(); i++) {

if (n[i] == ' ' || n[i] == '\n' || n[i] == '\0') {

curIndex++;

continue;

}

split[curIndex] += n[i];

}

return curIndex;

}

int main()

{

setlocale(LC\_ALL, "RUS");

ifstream F1("Fail1.txt");

if (F1.is\_open())

{

ofstream F2("Fail2.txt");

int row = 1;

while (!F1.eof())

{

string n;

int wordInN;

getline(F1, n);

wordInN = count(n.begin(), n.end(), ' ') + 1;

string\* splited = new string[wordInN];

int curIndex = splitedOn(n, splited);

bool answ = isHaveDuble(curIndex, splited);

if (answ) {

F2 << n << endl;

}

findWord(splited, curIndex, row);

row++;

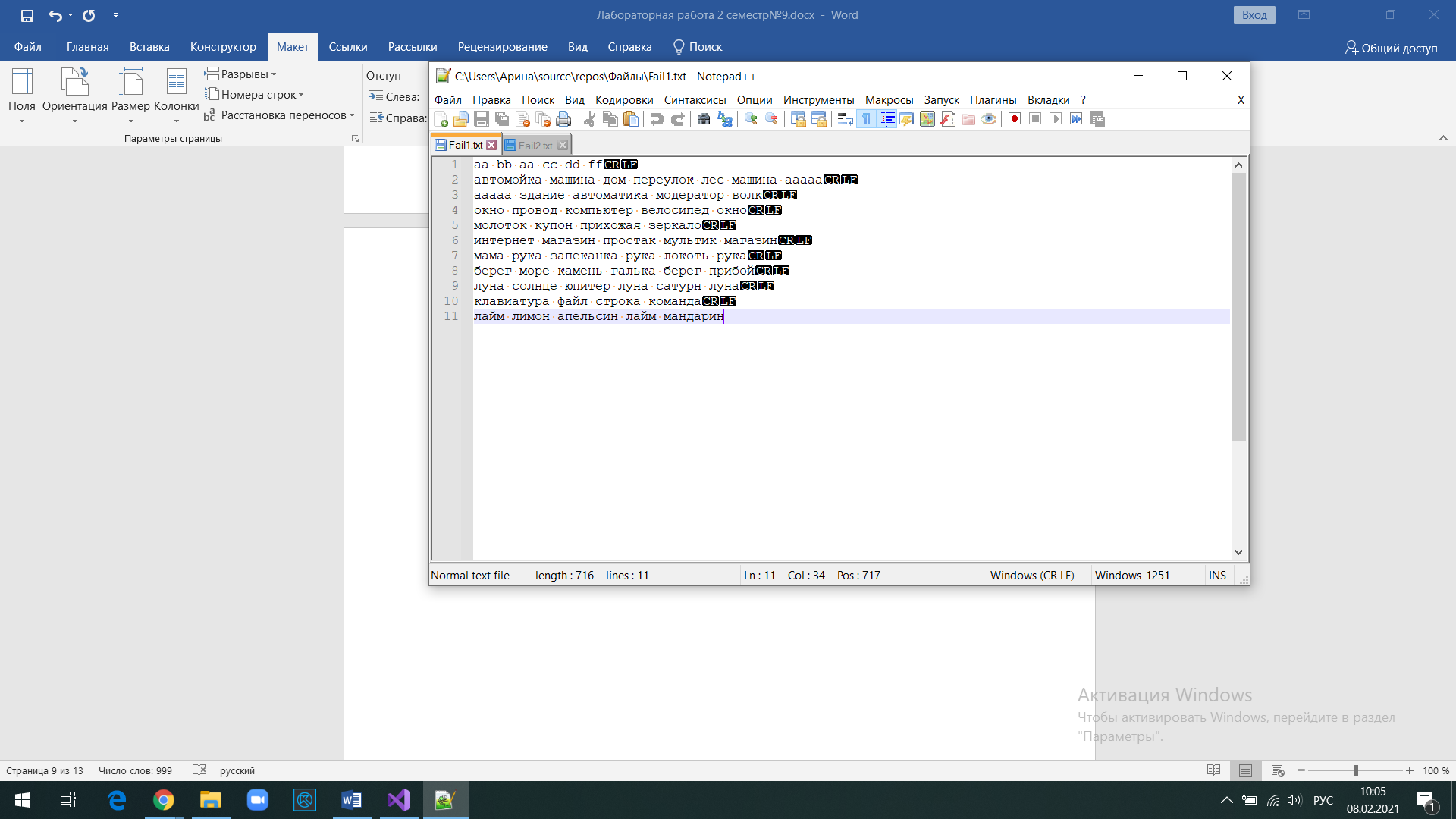
}

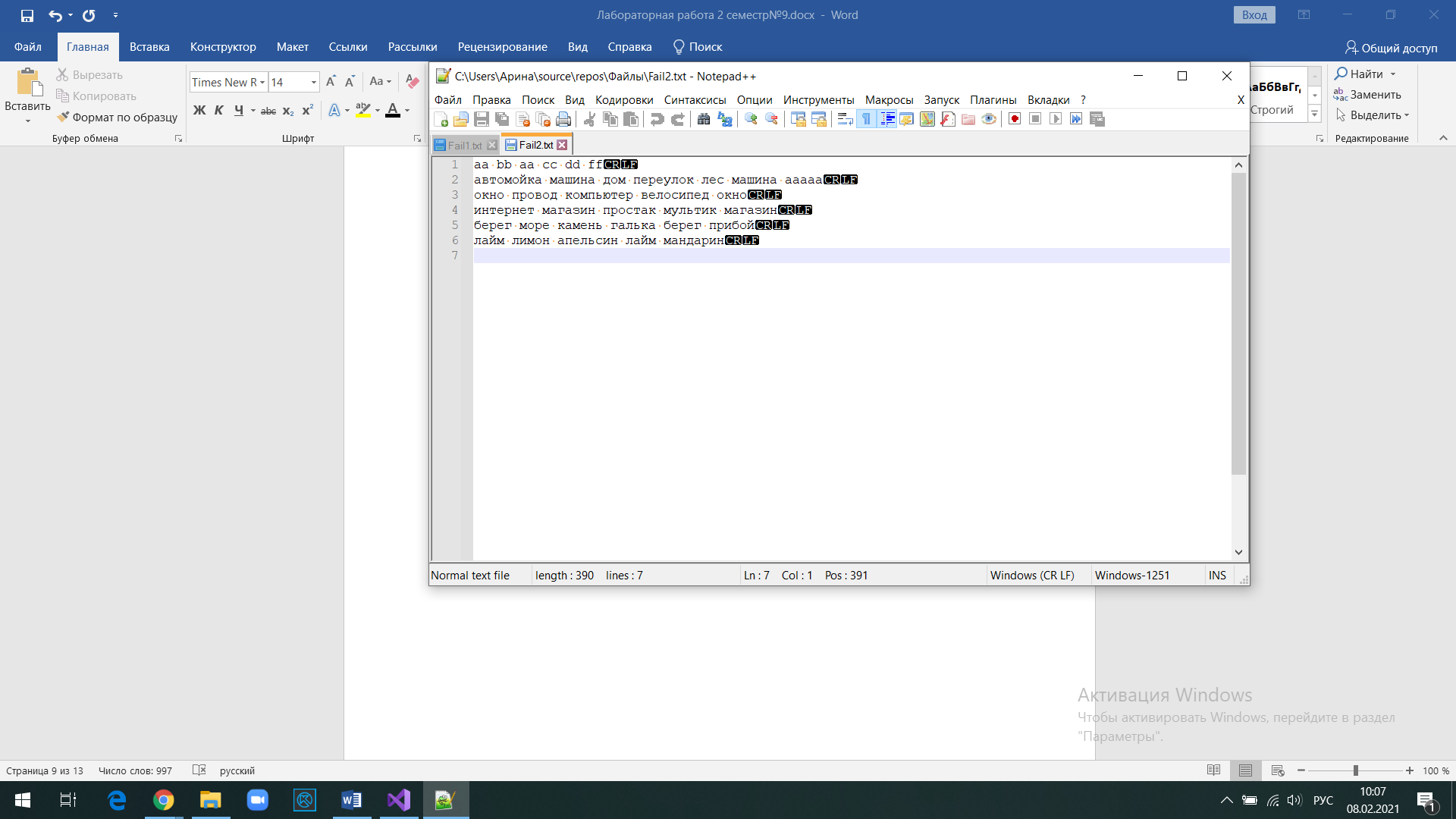
}

return 0;

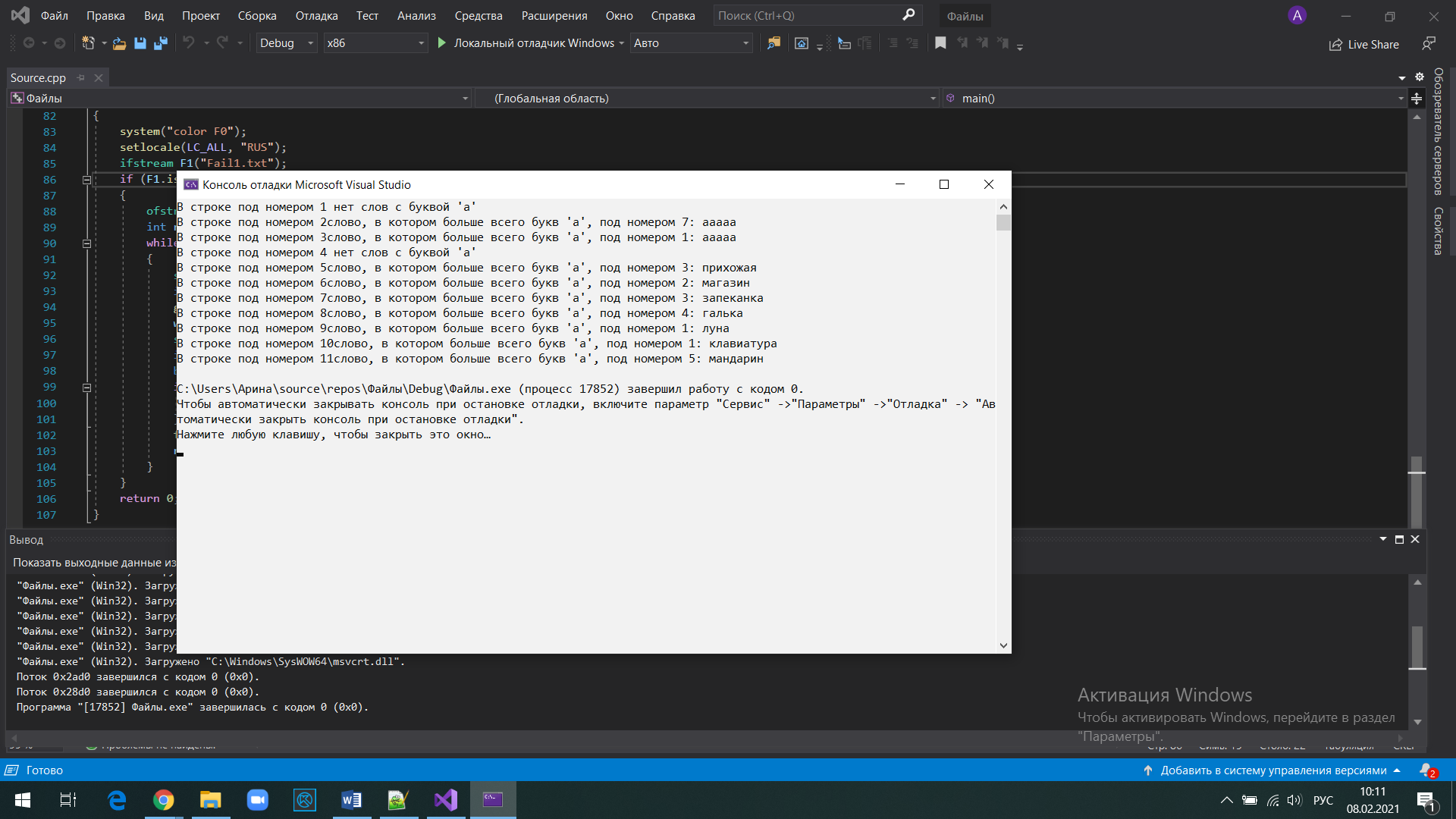
}

4. Выполнение программы.





Программа перенесла из файла F1 в файл F2 все необходимые строки, независимо от того на каком языке была написана строка. По условию необходимо перенести только те строки, в которых содержится ровно два одинаковых слова, если в строке есть три и больше одинаковых слова, то эта строка нам не подходит.



Поскольку проверка идёт на русскую букву «а», то в строке с английскими символами программа не нашла ни одну подходящую букву и вывела сообщение об этом.

5. Блок схема

