МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ»

(КНИТУ-КАИ)

Институт компьютерных технологий и защиты информации

(наименование института (факультета), филиала)

Кафедра Прикладной Математики и Информатики

(наименование кафедры)

09.03.04 «Программная инженерия»

(шифр и наименование направления подготовки (специальности))

Лабораторная работа № 6-7

По дисциплине: Операционные системы

На тему: "Работа с файловой системой"

Выполнил: студент группы 4310

**Гауиш М.Г**.

Проверил: доц. **Урахчиский И. Н**

Казань, 2023 год

Цель работы:

1) Изучение объектов WSH работы с файловой системой.

2) Изучение приемов программирования с использованием объектов работы с файловой системой.

3) Получение навыков разработки сценариев.

Содержание работы:

1) Изучение основных свойств и методов объектов FileSystemObject, Drive, Folder, File и TextStream .

2) Выработка навыков использования объектов FileSystemObject, Drive, Folder, File, TextStream.

3) Изучение и выполнение всех рассмотренных в руководстве примеров.

4) Разработка процедуры решения поставленной задачи.

5) Отладка и выполнение процедуры.

Краткие теоретические сведения

Объект FileSystemObject нужен для дисковых операций. Он позволяет читать, писать, удалять файлы и создавать каталоги. ProgId этого компонента – Scripting.FileSystemObject. FileSystemObject имеет методы, описанные в таблице 6.1. Специальные обозначения: […] – необязательный параметр src, dst – источник и приёмник. Если особо не оговаривается, то это строки с путями path – путь к файлу или каталогу name – имя файла или каталога spec – идентификатор системного каталога. (0 – Windows, 1 – System, 2 – Temp) unicode – определяет режим текста (TristateTrue – Unicode, TristateFalse – ASCII, TristateUseDefault – по умолчанию) iomode – определяет режим открытия (1 –для чтения, 2 – для записи, 8 – для добавления в конец файла).

Пример 1:

// fsoDirs.js - получение системных каталогов var fso=WScript.CreateObject("Scripting.FileSystemObject"); var s="Специальные каталоги:"; s+="\nWindows "+fso.GetSpecialFolder(0); s+="\nSystem "+fso.GetSpecialFolder(1); s+="\nTemp "+fso.GetSpecialFolder(2); WScript.Echo(s);

Пример 2:

// fsoEnumDrv.js – вывод списка подключенных дисков var fso=WScript.CreateObject("Scripting.FileSystemObject"); var e=new Enumerator(fso.Drives); var s=""; var n="" for(;!e.atEnd();e.moveNext()) { var x=e.item(); s=s+x.DriveLetter; s+=" - "; switch(x.DriveType) { case 1: n="Отключаемый";break; case 2: n="Жёсткий";break; case 3: n="Сетевой";break; case 4: n="CD-ROM";break; case 5: n="RAM диск";break; default: n="Unknown"; } s+=n+", "; if(x.DriveType==3) n=x.ShareName; else if (x.IsReady) n=x.VolumeName; else n="[Drive not ready]"; s+="\""+n+"\"\n"; } WScript.Echo(s);

Пример 3:

// fsoCreateText.js - создание файла с деревом каталогов var of; function ProcessDirectory(dir,prefix) { var foldPrefix=prefix+"-"; prefix+="|"; var newPrefix=prefix+" "; var fc = new Enumerator(dir.SubFolders); while(!fc.atEnd()) { of.WriteLine(foldPrefix+fc.item().Name); ProcessDirectory(fc.item(),newPrefix); fc.moveNext(); } fc = new Enumerator(dir.Files); while(!fc.atEnd()) { of.WriteLine(prefix+fc.item().Name); fc.moveNext(); } } if(WScript.Arguments.Count()==2) { var fso=WScript.CreateObject("Scripting.FileSystemObject"); var of=fso.CreateTextFile(WScript.Arguments(0),true); ProcessDirectory(fso.GetFolder(WScript.Arguments(1)),"") of.Close() }else WScript.Echo("Использование: cscript //nologo "+WScript.ScriptFullName+ " tree.txt c:\\windows\\temp");

Пример 4:

// fsoCreateText.js - создание файла с таблицей умножения if(WScript.Arguments.Count()==1) { var fso=WScript.CreateObject("Scripting.FileSystemObject"); var tf=fso.CreateTextFile(WScript.Arguments(0),true); tf.WriteLine("Таблица Пифагора:"); var i; tf.Write(" |"); for(i=1;i<10;i++) { tf.Write(" "+i+"|"); } tf.WriteLine(""); tf.WriteLine("-+--+--+--+--+--+--+--+--+--+"); var j; for(i=1;i<10;i++) { tf.Write(i); for(j=1;j<10;j++) { tf.Write("|"); if(i\*j<10) tf.Write(" "); tf.Write(i\*j); } tf.WriteLine("|"); } tf.WriteLine("-+--+--+--+--+--+--+--+--+--+"); tf.Close(); }else WScript.Echo("Использование: cscript //nologo "+WScript.ScriptFullName+ " pifagor.txt");

Постановка задачи

1) Разработать процедуру, выполняющую заданную в индивидуальном задании последовательность операций с папками (каталогами) с помощью различных объектов работы с файловой системой. Организовать ввод всех исходных данных в процедуру с помощью параметров командной строки при вызове процедуры. Предусмотреть возможность передачи результатов выполнения процедуры в файл.

**Задание 1:**

Разработать процедуру, выполняющую заданную в индивидуальном задании последовательность операций над файлами (каталогами) с помощью различных объектов работы с файловой системой. Организовать ввод всех исходных данных в процедуру с помощью параметров командной строки при вызове процедуры. Предусмотреть возможность перенаправления результатов выполнения процедуры в файл.

**1. Копирование файлов:**

1.1. Создать исходный каталог;

1.2.С помощью Блокнота создать исходный файл;

1.3. Создать результирующий каталог;

1.4. Скопировать исходный файл в результирующий каталог;

1.5. Запустить Блокнот для редактирования файла в результирующем каталоге

1.6. Сравнить файлы в исходном и результирующем каталогах;

**Листинг программы:**

var fileObject=WScript.CreateObject("Scripting.FileSystemObject");

var shell = WScript.CreateObject("WScript.Shell");

fileObject.CreateFolder("Test");

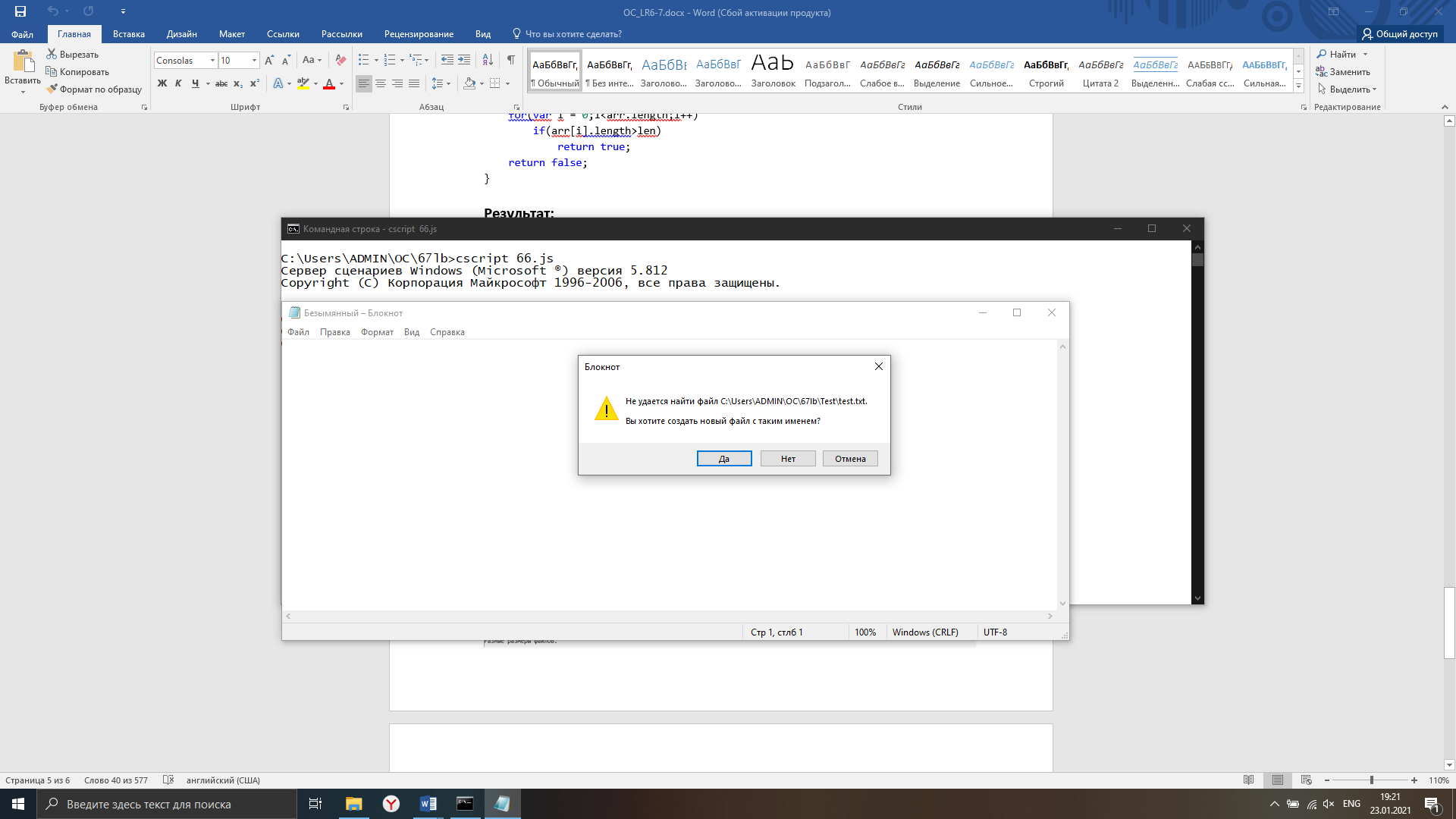
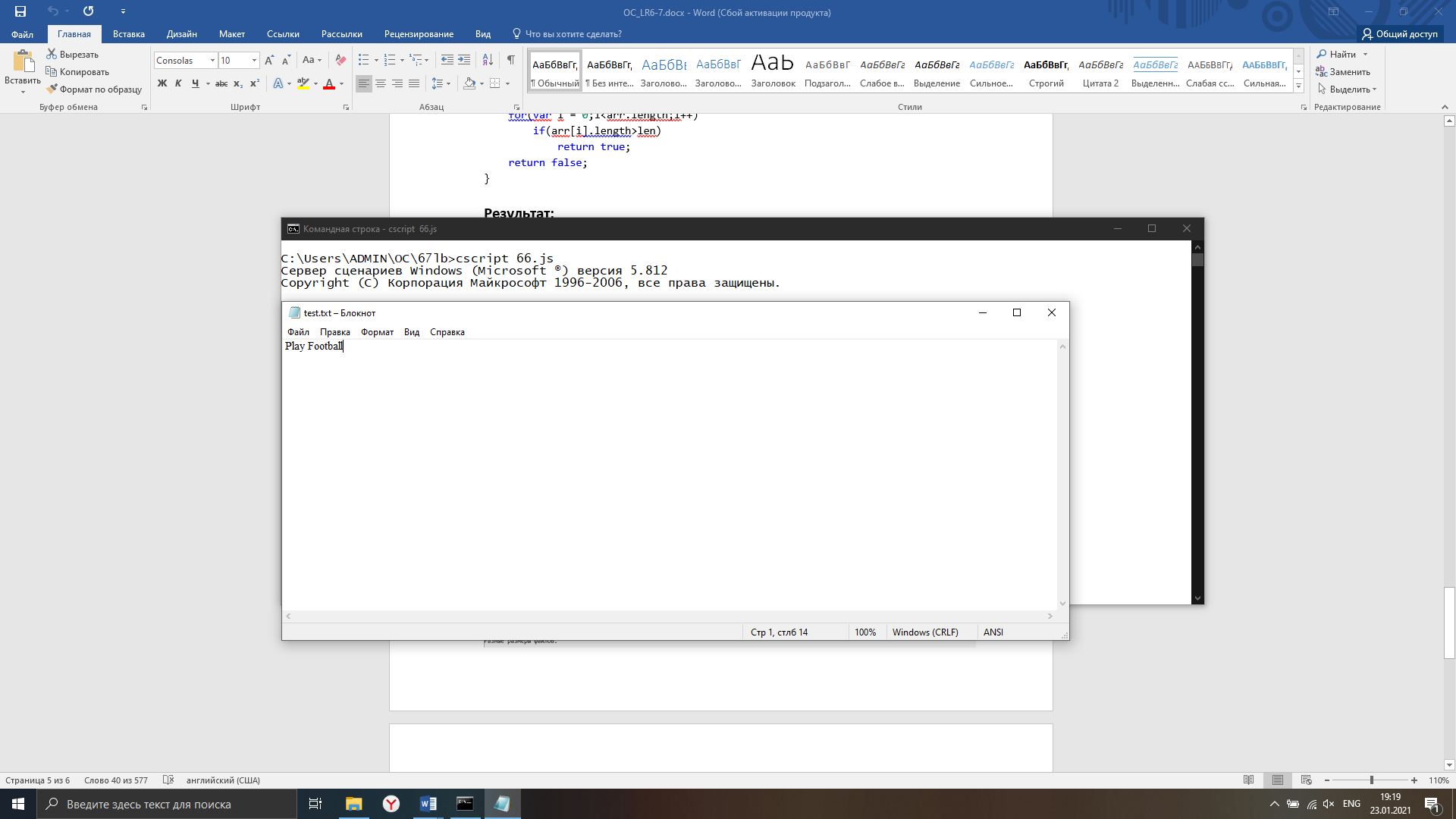
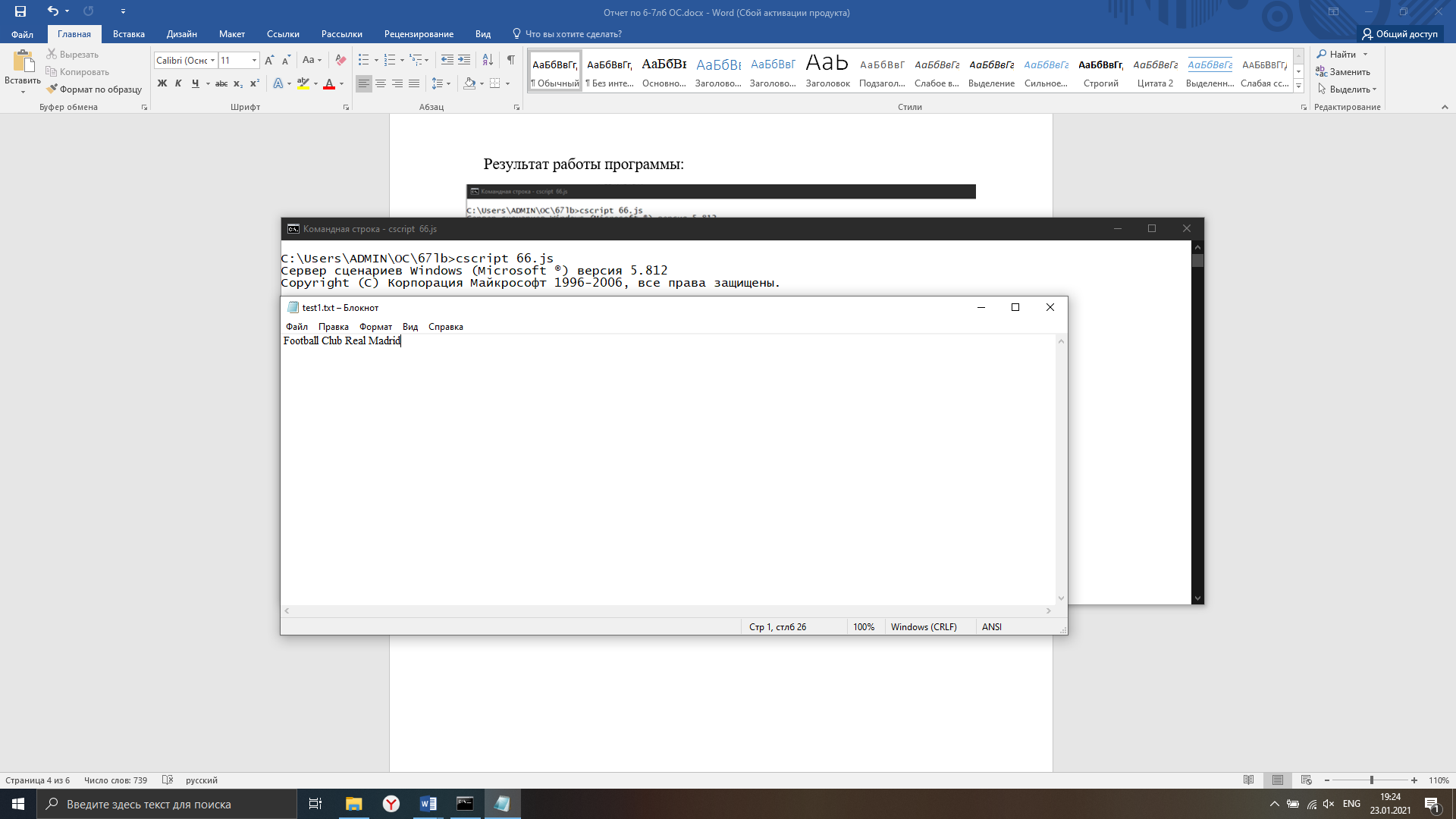
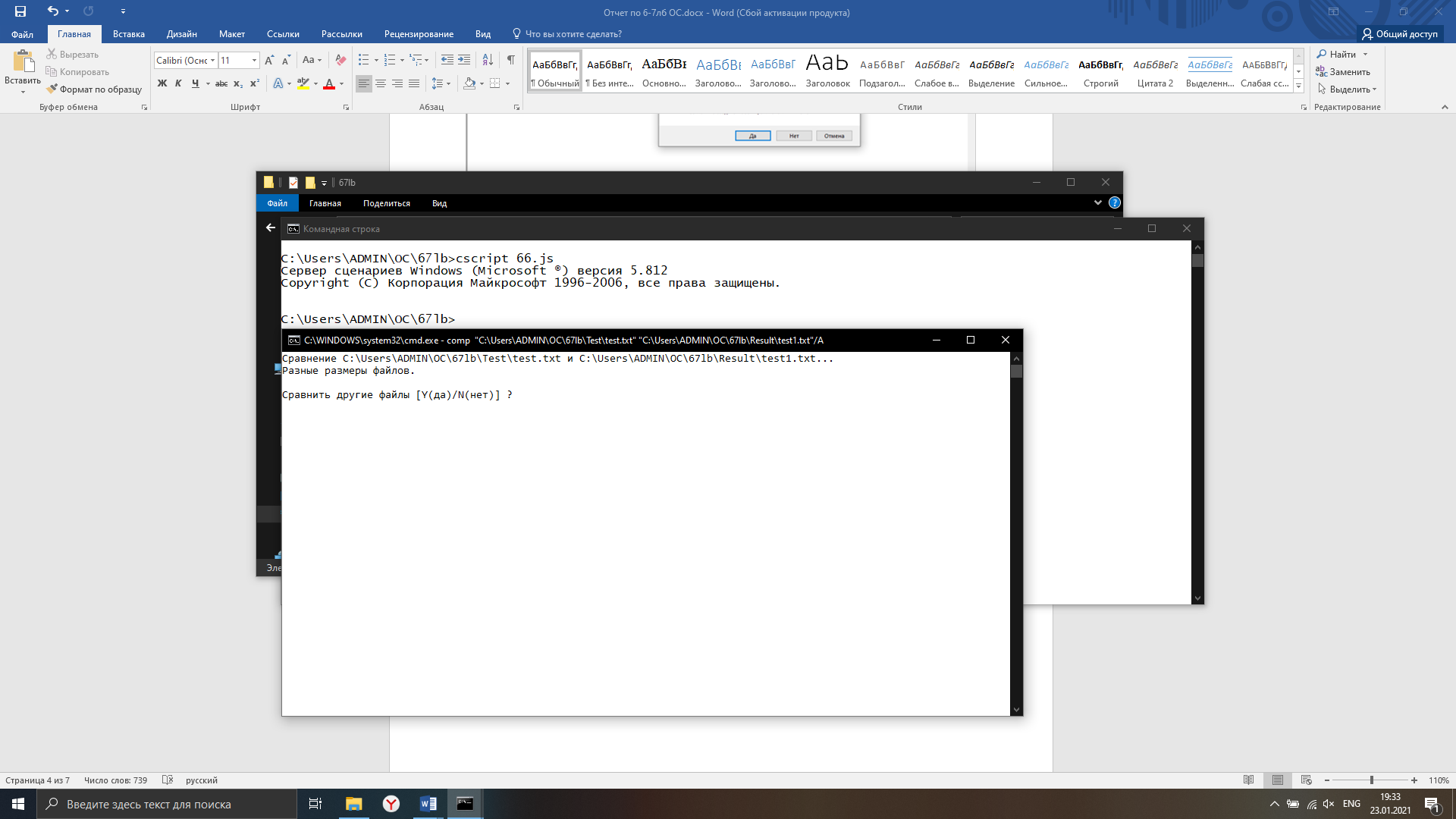
shell.Run(("notepad.exe '" + shell.CurrentDirectory+"\\" + "Test\\test.txt" + "'"),1,true)

fileObject.CreateFolder("Result");

fileObject.CopyFile("Test\\test.txt","Result\\test1.txt")

shell.Run(("notepad.exe '" + shell.CurrentDirectory+"\\" + "Result\\test1.txt" + "'"),1,true)

shell.Run('%COMSPEC% /K comp "'+ shell.CurrentDirectory +'\\'+ "Test\\test.txt" + '" "' + shell.CurrentDirectory +'\\'+"Result\\test1.txt" + '"'+'/A');

Результат работы программы:

2) Разработать процедуру обработки текстового файла, в соответствии с индивидуальным заданием. Организовать ввод исходных данных в процедуру с помощью параметров командной строки при вызове процедуры. Создание исходного текста на русском языке в текстовом файле произвести в процедуре путем вызова Блокнота. Предусмотреть возможность перенаправления результатов выполнения процедуры в файл.

Задан исходный текст на русском языке. Длина текста - не более NL символов, длина строки - не более NS символов, длина слова – не более NW символов. Исходный текст должен заканчиваться точкой. Составить процедуру, которая в заданном тексте находит слова, начинающиеся с заданной буквы. В качестве результата вывести исходный текст, найденные слова и их количество.

**Листинг программы:**

var dir = WScript.Arguments(0);

var filename = WScript.Arguments(1);

var NL = WScript.Arguments(2)

var NS = WScript.Arguments(3)

var NW = WScript.Arguments(4)

var currentLetter=WScript.Arguments(5)

var shell = WScript.CreateObject("WScript.Shell");

var fso = WScript.CreateObject("Scripting.FileSystemObject");

var outfile = fso.CreateTextFile("result.txt");

if(fso.FolderExists(dir) == false){ fso.CreateFolder(dir); }

shell.Run(shell.ExpandEnvironmentStrings("%WINDIR%\\notepad.exe " + dir + filename),1,true);

var textFile =fso.OpenTextFile(dir + filename, 1, true);

var text = ''

while(!textFile.AtEndOfLine){

    var line = textFile.ReadLine()

    if(CheckWord(line,NW)){

        WScript.Echo("Word length > NW");

        WScript.Quit()

    }

    if(CheckLen(line,NS)){

        WScript.Echo("Line length > NS");

        WScript.Quit()

    }

    text += line;

}

textFile.Close()

if(CheckLen(text,NL)){

    WScript.Echo("Text length > NL");

    WScript.Quit()

}

text = text.replace(/\./g,'');

text = text.replace(/\,/g,'');

var count = 0;

var res = ''

var list = text.split(" ")

for(var i = 0; i<list.length;i++){

if (list[i].toLowerCase().indexOf(currentLetter)==0){

        count++;

        res += "word "+i+": "+list[i]+'\n'

    }

}

res +="total count "+": "+count+'\n'+"text: "+text

WScript.Arguments.count()==7 ? outfile.Write(res) : WScript.Echo(res)

function CheckLen(text,len){

    return text.length>len?true:false;

}

function CheckWord(line,len){

    line = line.replace(/\./g,'');

    line = line.replace(/\,/g,'');

    var arr = line.split(' ')

    for(var i = 0;i<arr.length;i++)

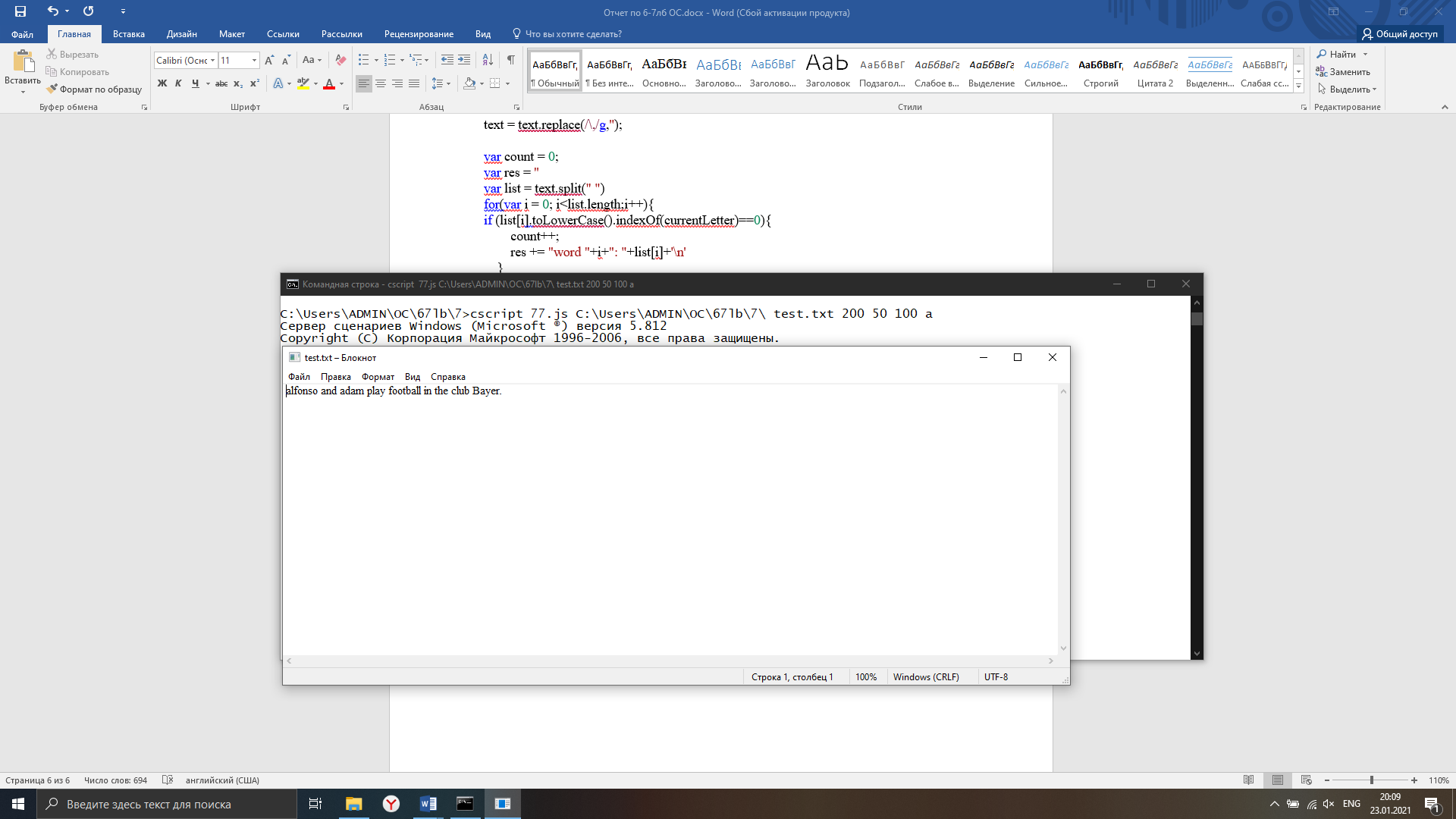
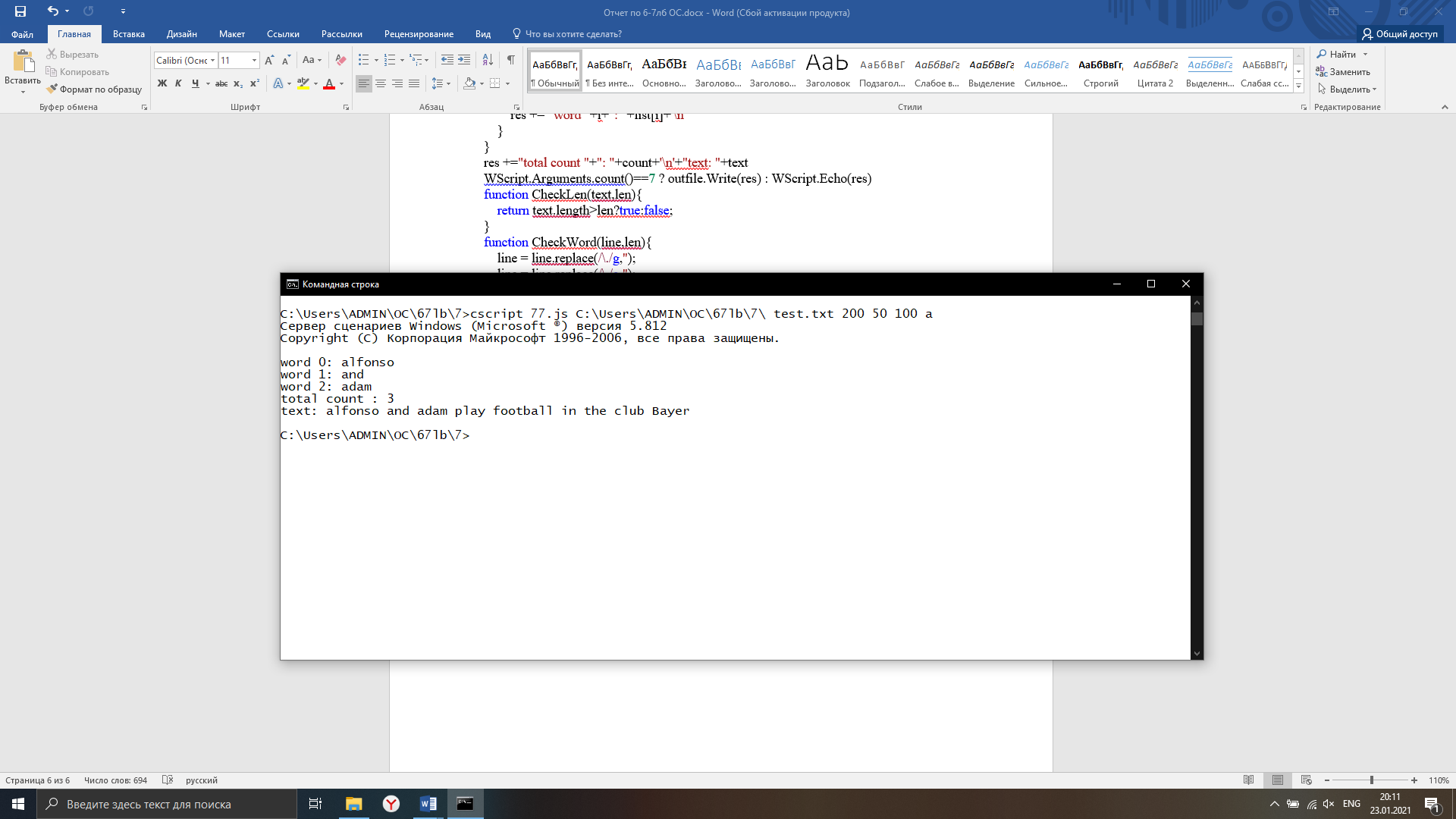
        if(arr[i].length>len)

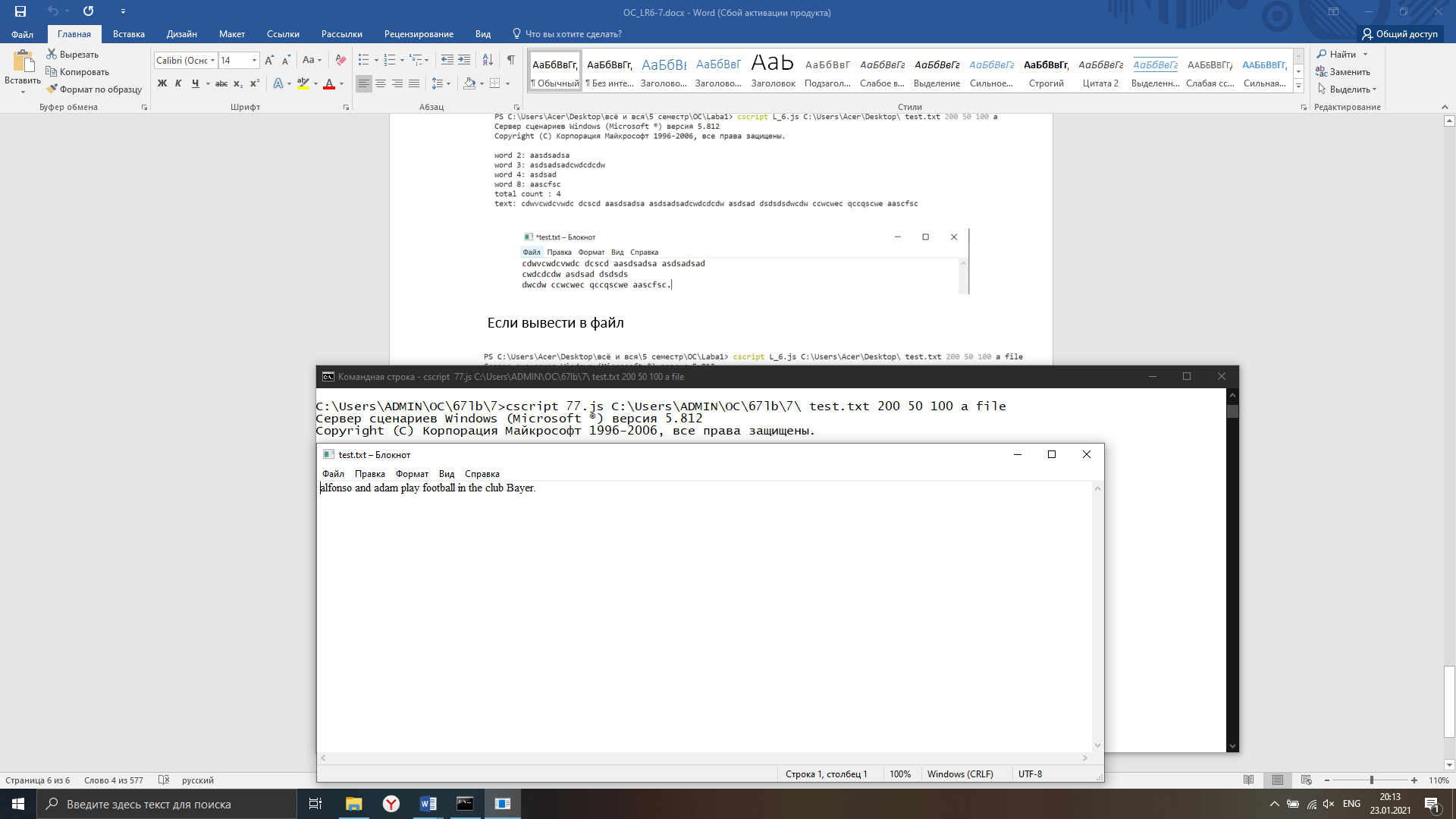
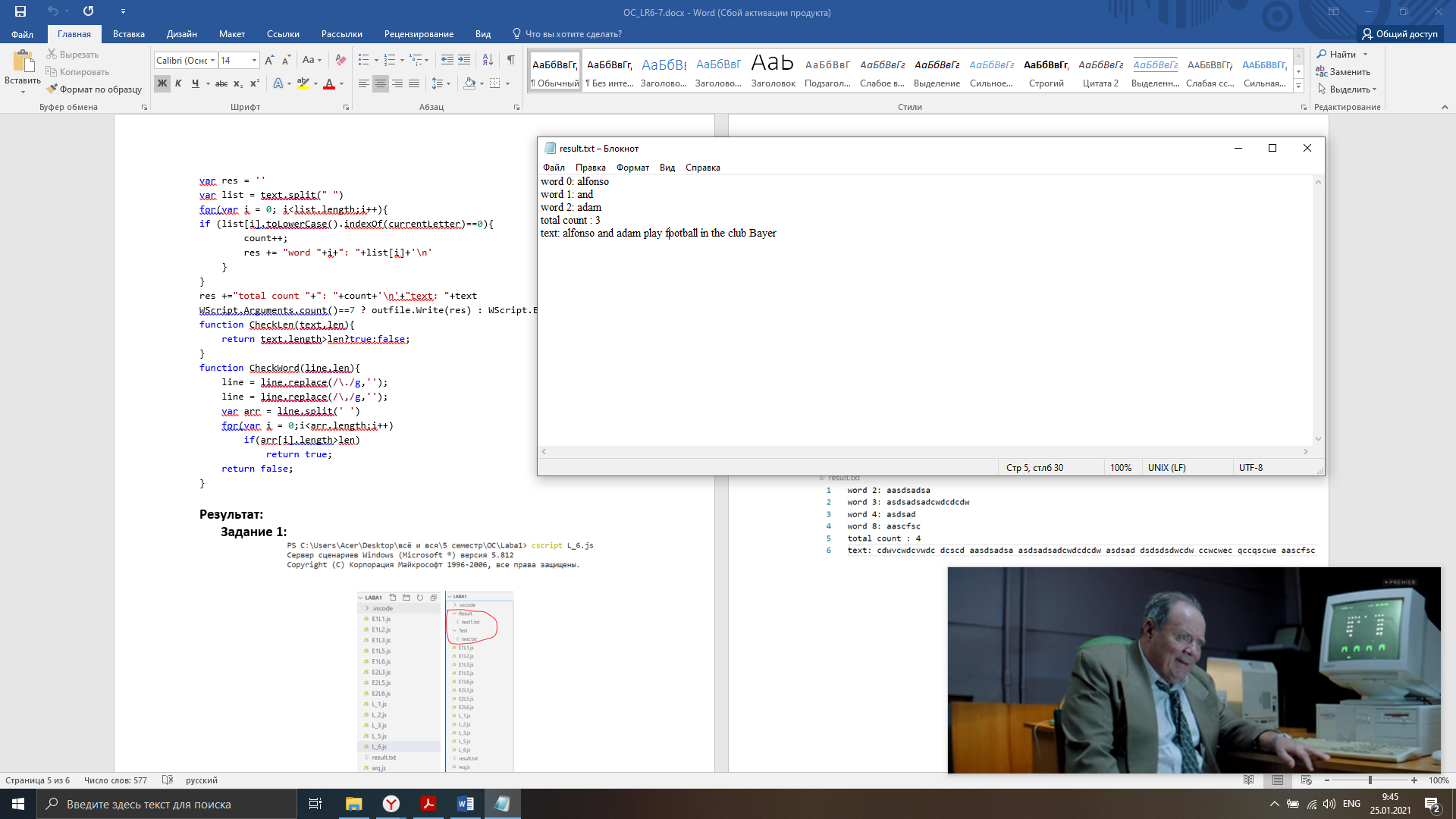
            return true;

    return false;

}

Результат работы программы:

Если не выводить в файл

Если вывести в файл