Правительство Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Кафедра «Компьютерная безопасность»

Отчет
к лабораторной работе №5

по дисциплине

«Языки программирования»

Работу выполнил
студент группы СКБ-201 _____ А.Н. Ушаков
подпись, дата

Работу проверила _____ М.Ю. Монина
подпись, дата

Москва 2021

Содержание

Постановка задачи

Разработать программу на языке Си++ (ISO/IEC 14882:2014), демонстрирующую решение поставленной задачи.

Общая часть

Тема: Создание приложений. *Меню, диалоги, работа с файлами.*

Задачи

Задание на 6 баллов.

Разработать графическое приложение с использованием библиотеки Qt – текстовый редактор:

1) с подсветкой текущей строки;

2) с нумерацией строк;

3) с подсветкой синтаксиса (переключение): Си (стандарт 2011), Си++ (стандарт 2014)

Заголовок окна должен содержать имя редактируемого файла (ограниченное 32 символами + троеточие) и признак того что файл был изменен (звездочка в начале имени) с момента последнего сохранения.

Окно (наследуется от QMainWindow) и содержит главное меню состоящее из пунктов:

1) Файл [кнопки]

а. Новый

б. Открыть

в. Сохранить

г. Сохранить как

д. Выход

2) Правка [кнопки]

а. Отменить

б. Повторить

в. Копировать

г. Вырезать

д. Вставить

е. Найти

ж. Найти и заменить

з. Выделить все

3) Формат

а. Перенос по словам [галочка]

б. Выбор шрифта – открывается модальный диалог выбора шрифта

4) Вид [галочки, где не указано иное]

а. Выбор цвета фона [кнопка] – открывает модальное окно выбора цвета

б. Выбор цвета текущей строки [кнопка] – открывает модальное окно выбора цвета

в. Вкл/Выкл отображения нумерации строк

г. Вкл/Выкл отображения панели инструментов

д. Вкл/Выкл отображения строки состояния

е. Вкл/Выкл подсветки синтаксиса

ж. Выбор синтаксиса (для подсветки) [дочернее меню] – доступно всегда, один синтаксис в дочернем меню выбран всегда

з. Выбор/Редактирование стиля подсветки [дочернее меню] – для текущего синтаксиса, по умолчанию выбран *Default*

i) Изменить [кнопка] – измененный стиль сохраняется в файл, имя файла становится именем стиля, стиль становится активным

ii) Загрузка стиля из файла [кнопка] – имя файла становится именем стиля, стиль становится активным

iii) Обязательная кнопка *Default*

iv) ***доступные стили***

5) Справка

а. О программе – открывает модальное окно, содержащее фото и имя автора, дату сборки, версию Qt с которой собиралось, версию Qt с которой запущено, кнопку закрывающую окно

Ниже главного меню располагается *панель инструментов* (отображение которой контролируется в меню **Вид**) с кнопками (с картинками, текстовое описание во всплывающей подсказке):

1) Новый документ

2) Открыть

3) Сохранить

4) Отменить

5) Повторить

6) Копировать

7) Вырезать

8) Вставить

9) Найти / Найти и заменить (как выпадающая кнопка) – открывающая *(немодальное)* диалоговое окно

В центральной части окна располагается область для редактирования текста. При **нажатии левой** кнопки курсор вставляется в позицию. При **двойном нажатии левой** кнопки выделяется слово под курсором. При **нажатии правой кнопки** *(далее – если нет=*, есть=** выделения)* курсор вставляется в позицию и выдается контекстное меню (кнопки могут быть неактивны): отменить, повторить, выделить*, выделить строку*, копировать**, вырезать**, вставить (** или если есть текст в буфере обмена), удалить**, выделить все.

Нижнюю часть окна занимает строка состояния. Информация разделена на *три* столбца: текущая позиция курсора (строка:столбец); время (и дата если другие сутки) последней операции (сохранения/изменения); количество строк, слов, символов, размер в килобайтах.

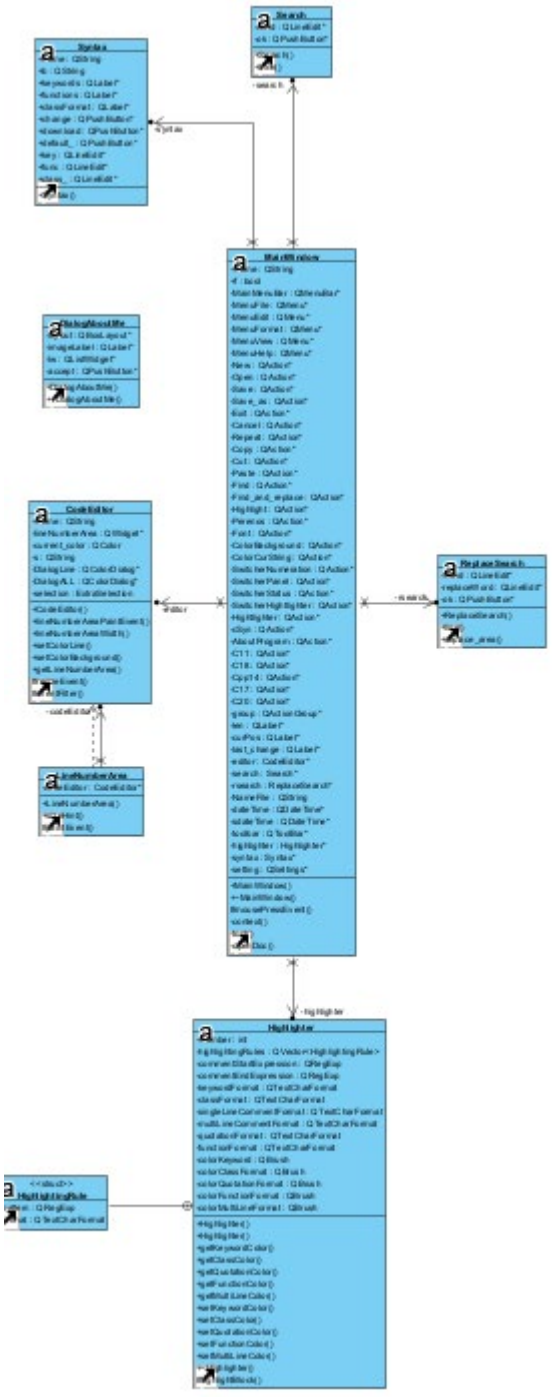
Задание на +1 балл. Реализовать подсветку синтаксиса Си (стандарт 2018), Си++17, Си++20

Задание на +1 балл. Реализовать сохранение настроек приложения *в ini-файл*.

Задание на +2 балла. Реализовать сохранение истории изменения редактируемого файла (бесконечную очередь команд повторения/отмены) во внешнем файле (обязательно хранить дату и время изменения; при отсутствии файла – создавать, при наличии – дописывать).

1. Алгоритм решения задачи

Для решения поставленных задач были разработаны следующие классы.



1.1 Класс CodeEditor

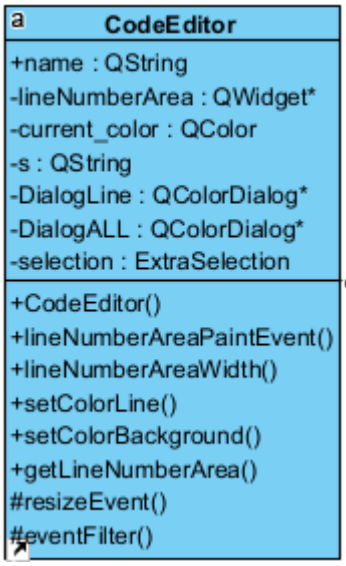


Рис. 1: UML-диаграмма класса *CodeEditor*

1.1.1 Конструктор

1) CodeEditor()

1.1.2 Перегрузки виртуальных функций

1) resizeEvent()
2) eventFilter()

1.1.3 Методы класса

1) lineNumberAreaPaintEvent()
2) lineNumberAreaWidth()

1.1.4 Методы доступа к компонентам класса

1) setColorLine()
2) setLineNumberArea()
3) getLineNumberArea()

1.1.5 Закрытые методы-слоты

1) updateLineNumberArea()
2) highlightCurrentLine()
3) updateLineNumberAreaWidth()
4) undo1()
5) redo1()

1.2 Класс LineNumberArea

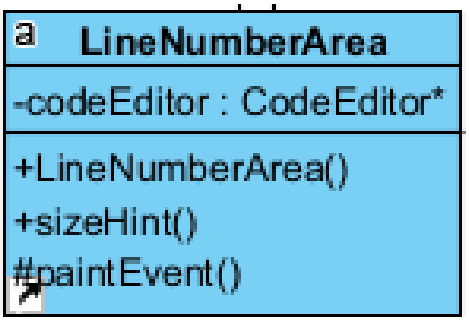


Рис. 2: UML-диаграмма класса LineNumberArea

1.2.1 Конструкторы

1) LineNumberArea()

1.2.2 Перегрузка виртуальных функций

1) paintEvent()

1.2.3 Методы класса

1) sizeHint()

1.3 Класс DialogAboutMe

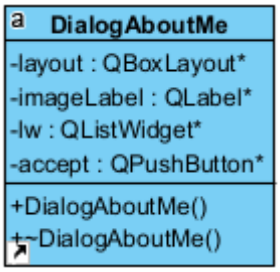


Рис. 3: UML-диаграмма класса DialogAboutMe

1.3.1 Конструкторы

1) DialogAboutMe()

1.3.2 Деструкторы

1) ~DialogAboutMe()

1.4 Класс ReplaceSearch

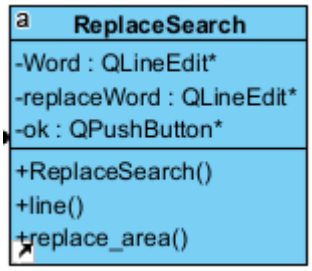


Рис. 4: UML-диаграмма класса ReplaceSearch

1.4.1 Конструкторы

1) ReplaceSearch()

1.4.2 Методы класса

1) line()
2) replace_area()

1.4.3 Закрытые методы-слоты

1) Replace()

1.5 Класс Search

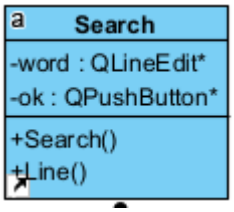


Рис. 5: UML-диаграмма класса Search

1.5.1 Конструкторы

1) Search()

1.5.2 Методы класса

1) Line()

1.6 Класс Syntax

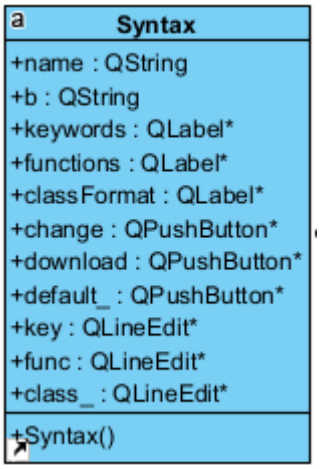


Рис. 6: UML-диаграмма класса Syntax

1.6.1 Конструкторы

1) Syntax()

1.6.2 Закрытые методы-слоты

1) Default()
2) Download()
3) Download1()

1.7 Класс Highlighter

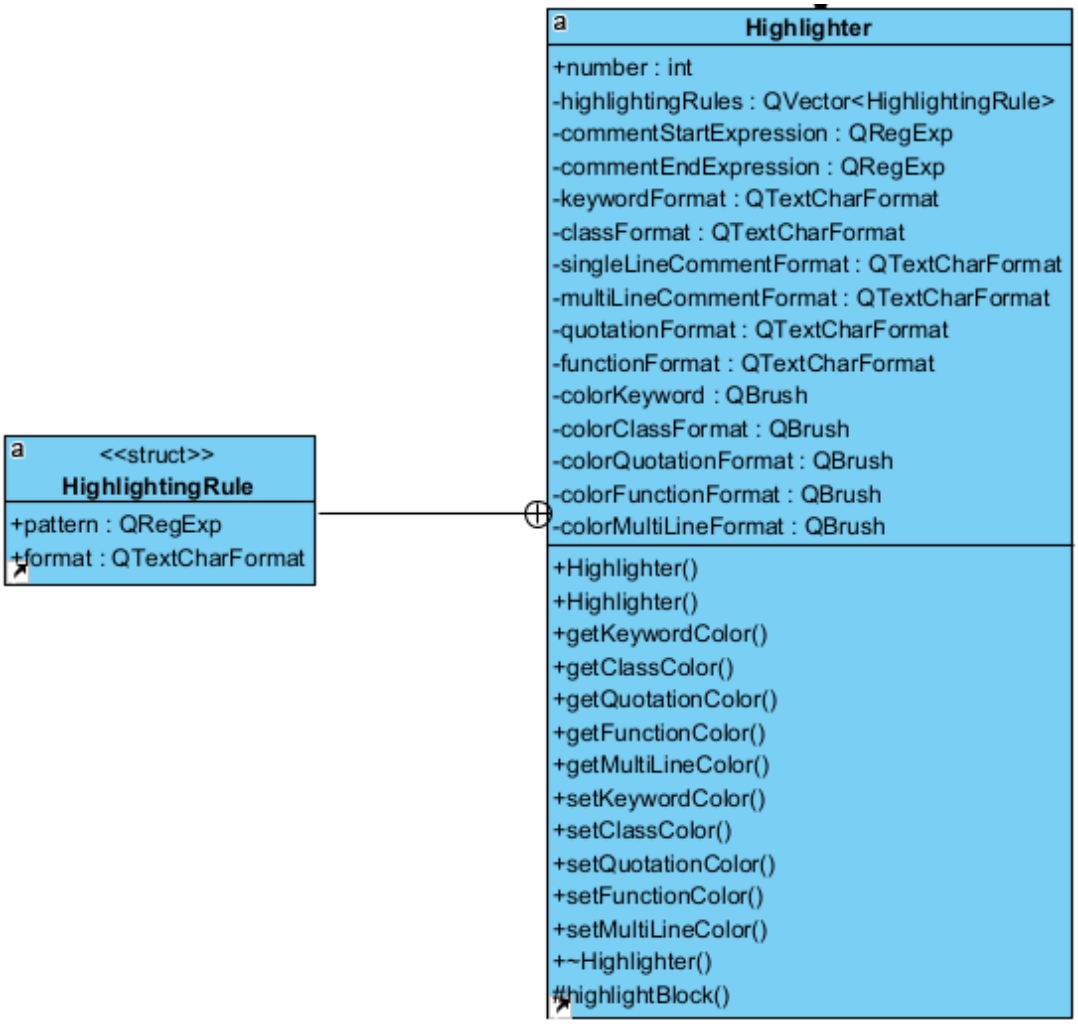


Рис. 7: UML-диаграмма класса Highlighter

1.7.1 Конструкторы

1) Highlighter(QTextDocument *p, int n=0)
2) Highlighter(QTextDocument *p, int n=0, QColor a, QColor b, QColor c)

1.7.2 Деструкторы

1) ~Highlighter()

1.7.3 Перегрузки виртуальных функций

1) highlightBlock()

1.7.4 Методы доступа к компонентам класса

1) getKeywordColor()
2) getClassColor()
3) getQuotationColor()

4) getFunctionColor()
5) getMultiLineColor()
6) setKeywordColor()
7) setClassColor()
8) setQuotationColor()
9) setFunctionColor()
10) setMultiLineColor()

1.8 Класс MainWindow

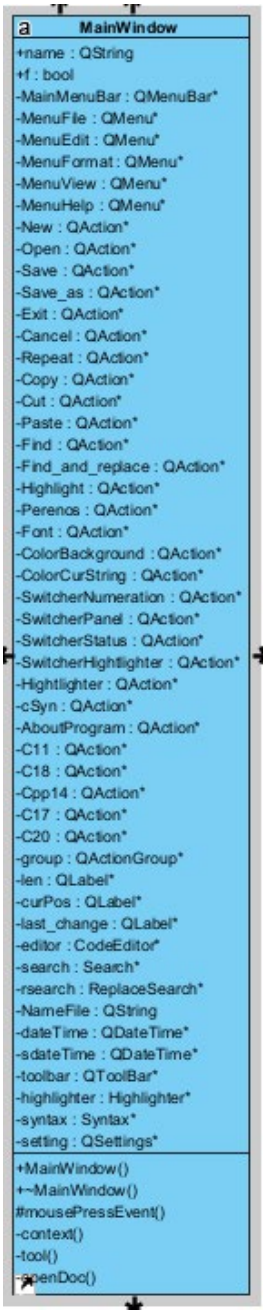


Рис. 8: UML-диаграмма класса MainWindow

1.8.1 Конструкторы

1) MainWindow()

1.8.2 Деструкторы

1) ~MainWindow()

1.8.3 Перегрузки виртуальных функций

1) mousePressEvent()

1.8.4 Методы класса

1) context()
2) tool()
3) openDoc()

1.8.4 Закрытые методы-слоты

1) exit()
2) showAboutProgram()
3) New_()
4) Open_()
5) Save_()
6) Save_As_()
7) setCurName()
8) RefreshStatusBar()
9) SetFontSelection()
10) HideStatusBar()
11) WrapWord()
12) HideNumberArea()
13) ChangeColorLine()
14) SetColorBackground()
15) ActiveSearch()
16) FindWord()
17) SearchandPeplace()
18) replaceEND()
19) SelectWord()
20) SelectLine()
21) Delete()
22) HideToolBar()
23) RefreshZv()

24) HideHighlighter()

25) v11()

26) v18()

27) v14()

28) v17()

29) v20()

30) cSyntax()

2. Выполнение задач

2.1 Класс CodeEditor

2.1.1 Конструкторы

CodeEditor() - конструктор, создающий объект класса LineNumberArea, устанавливающий фильтр событий на данном объекте класса CodeEditor, а также сигналы на счётчик строк, размеров окна, изменению позиции курсора соответствующим слотам на изменения счётчика строк и отрисовку области окна, а также подсветки линни, на которой расположен курсор в текущий момент времени.

2.1.2 Перегрузки виртуальных функций

resizeEvent() – перегруженная виртуальная **защищенная** функция, выполняющая роль «растягивания» нашего виджета пропорционально размерам изменяющегося окна.

eventFilter() – перегруженная виртуальная защищенная функция, «отлавливающая» нажатие клавиш и заносящая в отдельный текстовый историю изменения данного объекта класса CodeEditor.

2.1.3 Методы класса

lineNumberAreaPaintEvent() – закрытый метод данного класса, выполняющий отрисовку линии, на которой располагаются номера строк внутри данного виджета.

lineNumberAreaWidth() – закрытый метод данного класса, отвечающий за отрисовку счётчика строк внутри линии.

2.1.4 Методы доступа к компонентам класса

setColorLine() – открытый метод данного класса, выполняющий присваивание закрытому полю данного класса – current_color – переданное значение.

setColorBackground() - открытый метод данного класса, выполняющий изменение фона виджета, открытием диалогового окна с выбором цвета

getLineNumberArea() – открытый метод данного класса, выполняющий получение значения закрытого поля данного класса – lineNumberArea.

2.1.5 Закрытые методы-слоты

updateLineNumberAreaWidth() – закрытый метод-слот данного класса, выполняющий обновление ширины линии, на которой отображается нумерация строк так, чтобы номер строки «влезал» в эту область.

highlightCurrentLine() – закрытый метод-слот данного класса, создающий подсветку строки, на которой расположен курсор.

updateLineNumberArea – закрытый метод-слот данного класса, выполняющий обновление области линии, на которой отображается нумерация строк в зависимости от ширины окна.

undo1() – закрытый метод-слот данного класса, выполняющий возврат к предыдущему действию внутри виджета, и записывает это изменение в соответствующий текстовый файл, как историю изменения файла.

redo1() – закрытый метод-слот данного класса, выполняющий повторение последнего действия внутри виджета, и записывает это изменение в соответствующий текстовый файл, как историю изменения файла.

2.2 Класс LineNumberArea

2.2.1 Конструкторы

LineNumberArea() – конструктор данного класса, выполняющий инициализацию закрытого поля данного класса – CodeEditor.

2.2.2 Перегрузка виртуальных функций

PaintEvent() – перегрузка виртуальной защищенной функции, которая передаёт событие на отрисовку области с нумерацией строк соответствующему методу родительского элемента.

2.2.3 Методы класса

sizeHint() – метод, возвращающий размер исходной области с нумерацией строк.

2.3 Класс DialogAboutMe

2.3.1 Конструкторы

DialogAboutMe() – конструктор данного класса, выполняющий заполнение диалогового окна сведениями об авторе и Qt.

2.3.2 Деструкторы

~DialogAboutMe() – деструктор по умолчанию.

2.4 Класс ReplaceSearch

2.4.1 Конструкторы

ReplaceSearch() – конструктор данного класса, создающий диалоговое окно с соответствующими кнопками и полями.

2.4.2 Методы класса

line() - открытый метод класса, возвращающий значение поля класса в текстовом формате.

replace_area() - открытый метод класса, возвращающий значение поля класса в текстовом формате.

2.4.3 Закрытые методы-слоты

Replace() – закрытый метод-слот, возвращающий сигнал данного класса об замене найденного слова.

2.5 Класс Search

2.5.1 Конструкторы

Search() – конструктор данного класса, создающий диалоговое окно с соответствующими кнопками и полями.

2.5.2 Методы класса

Line() – открытый метод, возвращающий значение поля данного класса в текстовом формате.

2.6 Класс Syntax

2.6.1 Конструкторы

Syntax() – конструктор данного класса, создающий диалоговое окно с соответствующими кнопками и полями.

2.6.2 Закрытые методы-слоты

Default() – закрытый метод-слот, который устанавливает полям класса соответствующие значения и закрывает данное диалоговое окно.

Download() – закрытый метод-слот, который открывает файловое диалоговое окно и предоставляет пользователю выбрать файл. Потом считывает данный файл определенного формата и заносит данные в соответствующие поля нашего класса и закрывает диалоговое окно.

Download1() – закрытый метод-слот, который вызывается в зависимости от нажатой кнопки выбора стиля и заносит данные из соответствующего файла в поля данного класса.

2.7 Класс Highlighter

2.7.1 Конструкторы

Highlighter(QTextDocument *p,int n=0) – конструктор данного класса, которому передаём указатель на родителя и целочисленное значение, отвечающее за выбор одного из 5 видов синтаксиса.

Highlighter(QTextDocument *p,int n=0, QColor a, QColor b, QColor c) – конструктор данного класса, которому передаём указатель на родителя и целочисленное значение, отвечающее за выбор одного из 5 видов синтаксиса. Также передаем объекты класса QColor для заполнения полей, связанных с цветом подсветки.

2.7.2 Деструкторы

~Highlighter() – деструктор по умолчанию.

2.7.3 Перегрузки виртуальных функций

highlightBlock() – перегруженный виртуальный защищённый метод, отвечающий за блокировку подсветки синтаксиса внутри комментариев, а также задающий правила подсветки синтаксиса.

2.7.4 Методы доступа к компонентам класса

getKeywordColor() – открытый метод, возвращающий значение поля данного класса – colorKeyword.

getClassColor() – открытый метод, возвращающий значение поля данного класса – colorClassFormat.

getQuotationColor() – открытый метод, возвращающий значение поля данного класса – colorQuotationFormat.

getFunctionColor() – открытый метод, возвращающий значение поля данного класса – colorFunctionFormat.

getMultiLineColor() – открытый метод, возвращающий значение поля данного класса – colorMultiLineFormat.

setKeywordColor() – открытый метод, задающий значение поля данного класса – colorKeyword.

setClassColor() – открытый метод, задающий значение поля данного класса – colorClassFormat.

setQuotationColor() – открытый метод, задающий значение поля данного класса – colorQuotationFormat.

setFunctionColor() – открытый метод, задающий значение поля данного класса – colorFunctionFormat.

setMultiLineColor() – открытый метод, задающий значение поля данного класса – colorMultiLineFormat.

2.8 Класс MainWindow

2.8.1 Конструкторы

MainWindow() – конструктор данного класса, создающий основное меню, статус бар и вызывает метод, отвечающий за создание контекстного меню, а также метод, отвечающий за создание панели инструментов.

2.8.2 Деструкторы

~MainWindow() – деструктор по умолчанию**.**

2.8.3 Перегрузки виртуальных функций

mousePressEvent() – перегруженный виртуальный метод, который отлавливает двойное нажатие левой кнопки мышки и вызывает контекстное меню.

2.8.4 Методы класса

context() – закрытый метод класса, отвечающий за создание контекстного меню.

tool() – закрытый метод класса, отвечающий за создание панели инструментов.

openDoc() – закрытый метод класса, которому передаётся строка, как имя файла, которое необходимо найти и записать данные с полей класса. В случае удачной попытки возвращает 1, в противном случае – 0.

2.8.5 Закрытые методы-слоты

exit() – закрытый метод-слот класса, отвечающий за выход из приложения и сохранения настроек приложения в ini файл.

showAboutProgram() – закрытый метод-слот класса, вызывающий диалоговое окно с информацией об авторе и Qt.

New_() – закрытый метод-слот класса, отвечающий за создание нового документа.

Open_() – закрытый метод-слот класса, вызывающей диалоговое файловое окно с выбором файла, которой необходимо открыть внутри приложения.

Save_() – закрытый метод-слот класса, вызывающий диалоговое файловое окно с выбором директории, в которую необходимо сохранить файл, и заполнением имени файла, если файл был создан впервые. Если файл был создан ранее, то документ сохраняется с внесенными изменениями.

Save_As_() – закрытый метод-слот класса, вызывающий диалоговое файловое окно с выбором директории, в которую необходимо сохранить файл, и заполнением имени файла.

setCurName() – закрытый метод-слот класса, устанавливающий имя окна в зависимости от того, какой файл открыт.

RefreshStatusBar() – закрытый метод-слот класса, обновляющий значения в статус баре.

SetFontSelection() – закрытый метод-слот класса, вызывающий диалоговое окно с выбором шрифта документа.

HideStatusBar() – закрытый метод-слот класса, включающий/отключающий статус бар.

WrapWord() – закрытый метод-слот класса, включающий/отключающий перенос символов в строке.

HideNumberArea() – закрытый метод-слот класса, включающий/отключающий нумерацию строк.

ChangeColorLine() – закрытый метод-слот класса, вызывающий диалоговое окно с выбором цвета подсветки строки.

ActiveSearch() – закрытый метод-слот класса, вызывающий диалоговое окно с функцией поиска слов в документе. Если в документе найдено данное слово – оно подчеркивается голубым. Иначе открывается окно с уведомлением об неудачном поиске слова.

FindWord() – закрытый метод-слот класса, выполняющий поиск по тексту соответствующего слова.

SearchandReplace() – закрытый метод-слот класса, вызывающий диалоговое окно с функцией поиска и замены символов в документе.

replaceEND() – закрытый метод-слот класса, выполняющий поиск по тексту соответствующего слова и его замену.

SelectWord() – закрытый метод-слот класса, выполняющий выделение слова, находящегося под курсором.

SelectLine() – закрытый метод-слот класса, выполняющий выделение строки, где находится курсор.

Delete() – закрытый метод-слот класса, выполняющий удаление выделенных символов в документе.

HideToolBar() – закрытый метод-слот класса, включающий/отключающий панель инструментов.

RefreshZV() – закрытый метод-слот класса, отвечающий за добавление/удаление звездочки в начале имени окна приложения.

HideHighlighter() – закрытый метод-слот класса, включающий/отключающий подсветку синтаксиса.

v11() – закрытый метод-слот класса, включающий подсветку синтаксиса для стандарта C 2011 года.

v18() – закрытый метод-слот класса, включающий подсветку синтаксиса для стандарта C 2018 года.

v14()– закрытый метод-слот класса, включающий подсветку синтаксиса для стандарта C++ 2014 года.

v17()– закрытый метод-слот класса, включающий подсветку синтаксиса для стандарта C++ 2017 года.

v20()– закрытый метод-слот класса, включающий подсветку синтаксиса для стандарта C++ 2020 года.

cSyntax()– закрытый метод-слот класса, вызывающий диалоговое окно с выбором подсветки синтаксиса. Выбранные значения записываются в отдельный файл.

3. Получение исполняемых модулей

С помощью системы сборки QMake производится компоновка (сборка) исполняемого модуля из объектных модулей. Объектные модули – файлы, которые уже могут быть запущены, получаются, соответственно, из файлов исходного кода.

Листинг-1 labaWORD.pro

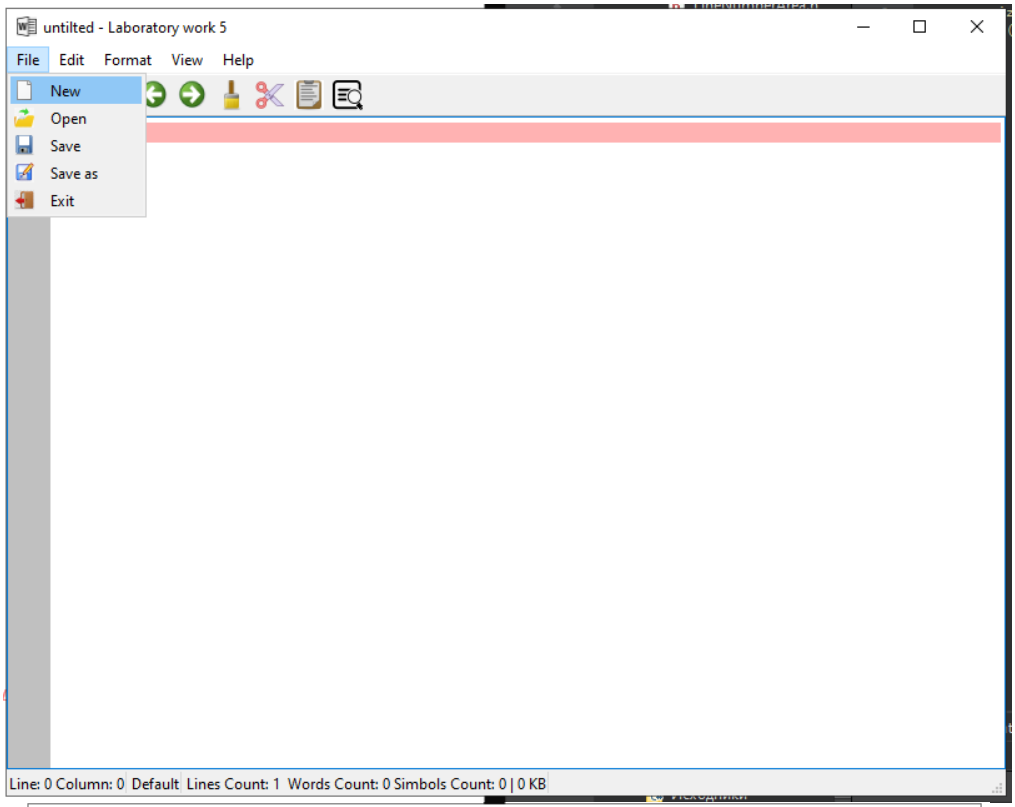
```
QT += core5compat
QT       += core gui

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets


CONFIG += c++11

# You can make your code fail to compile if it uses deprecated APIs.
# In order to do so, uncomment the following line.
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000    # disables all the APIs deprecated before Qt 6.0.0

SOURCES += \
    CodeEditor.cpp \
    DialogAboutMe.cpp \
    LineNumberArea.cpp \
    ReplaceSearch.cpp \
    Search.cpp \
```

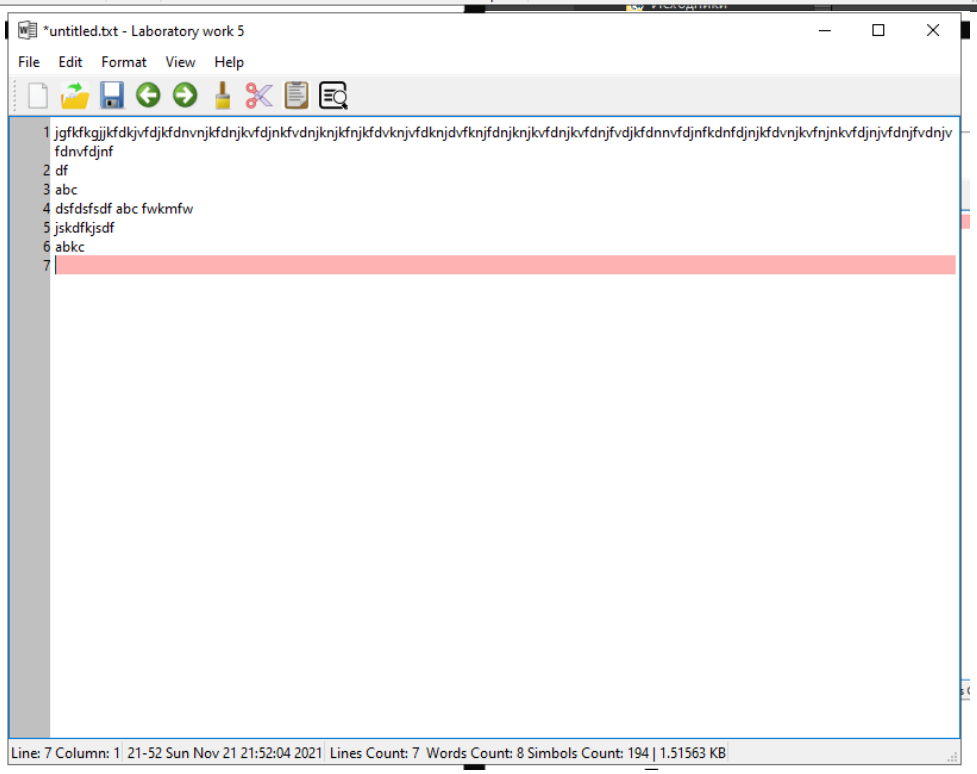
```
    Syntax.cpp \
    highlight.cpp \
    main.cpp \
    mainwindow.cpp

HEADERS += \
    CodeEditor.h \
    DialogAboutMe.h \
    LineNumberArea.h \
    ReplaceSearch.h \
    Search.h \
    Syntax.h \
    highlighter.h \
    mainwindow.h

# Default rules for deployment.
qnx: target.path = /tmp/$${TARGET}/bin
else: unix:!android: target.path = /opt/$${TARGET}/bin
!isEmpty(target.path): INSTALLS += target

RESOURCES +=
```

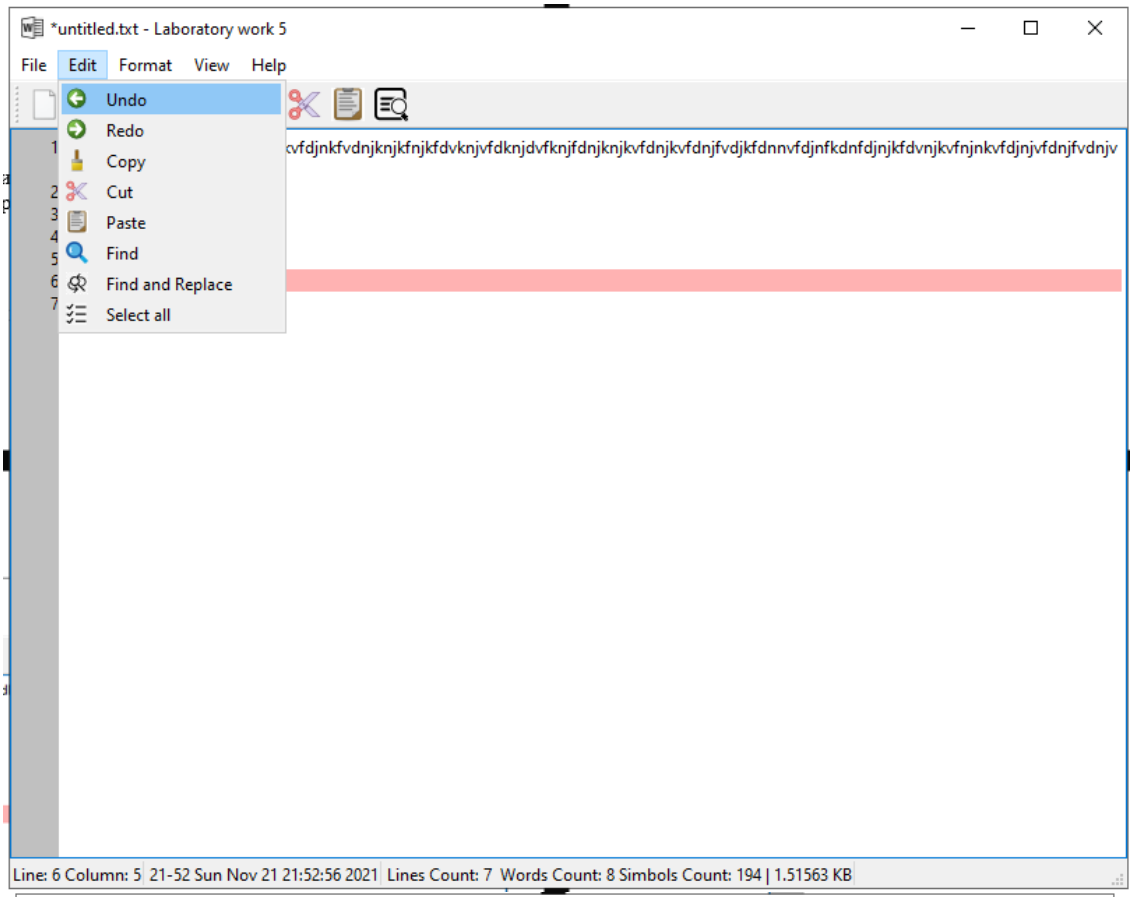
4. Тестирование

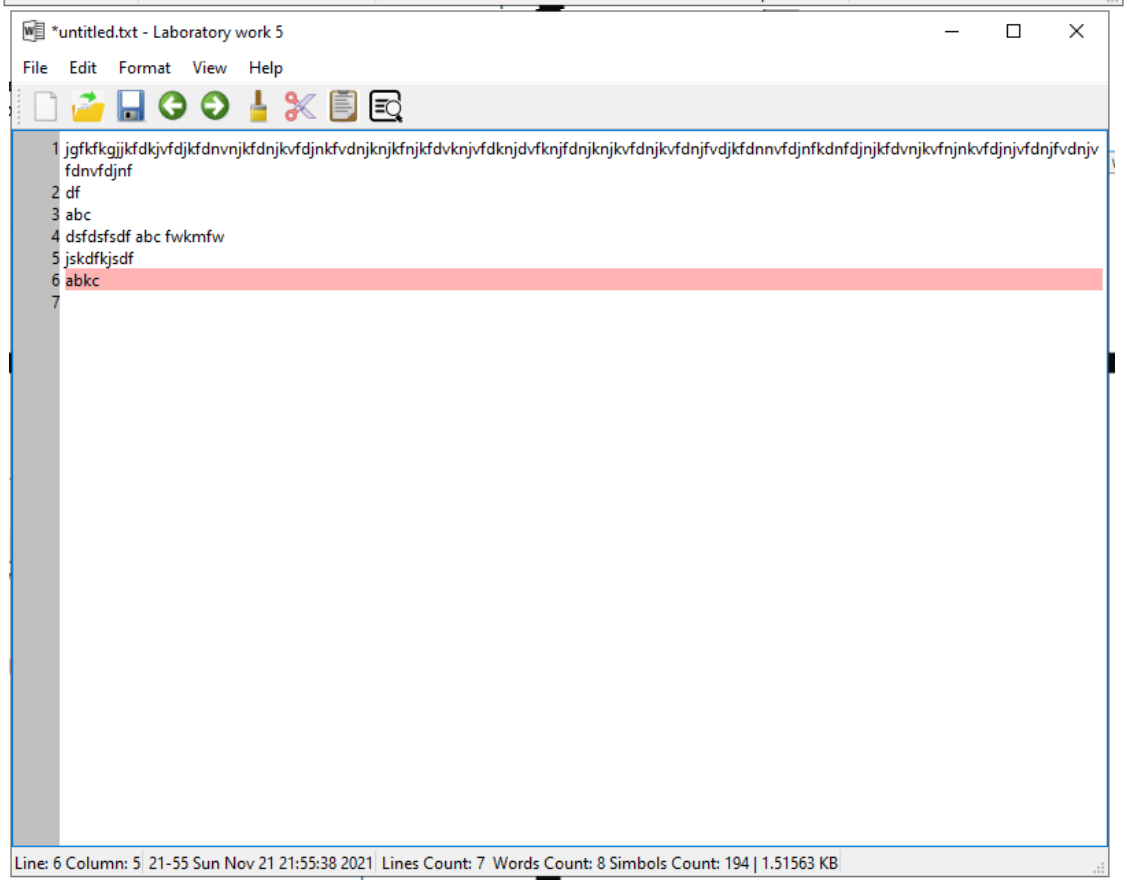
Test №1

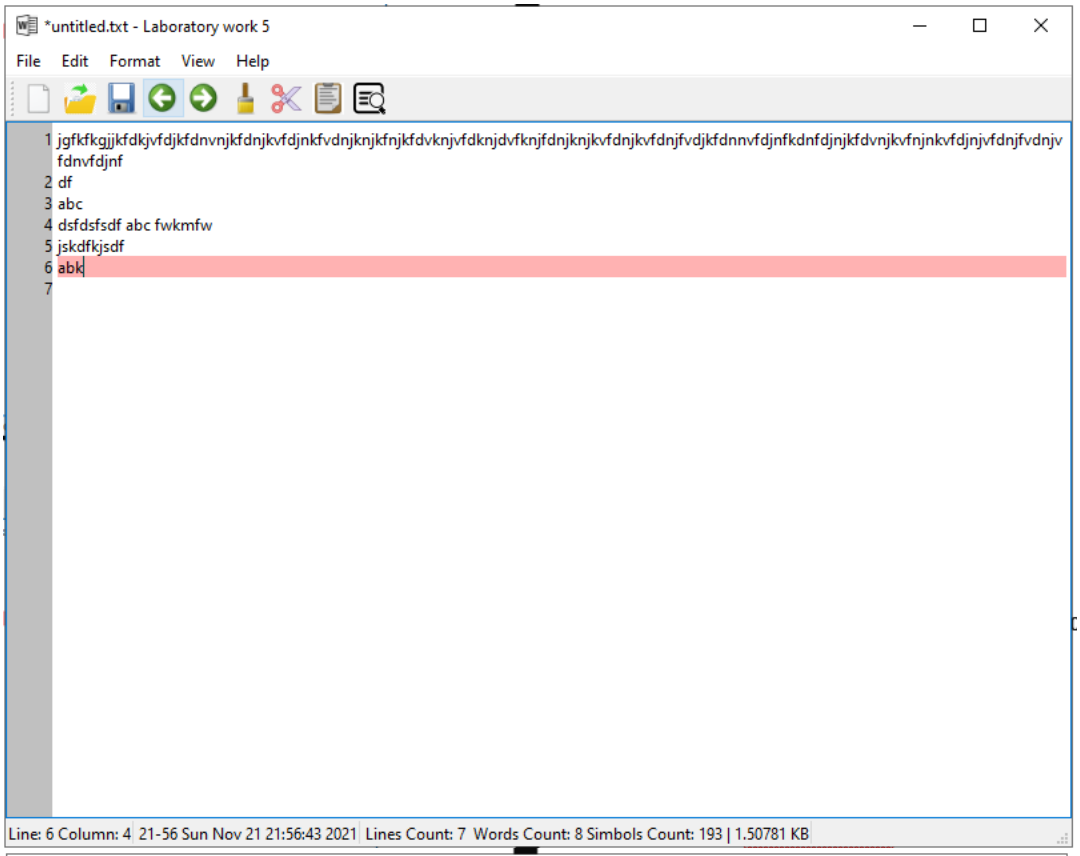
Происходит создание нового документа и с клавиатуры вводится случайная последовательность символов. Дальше проверяем все кнопки undo и redo. Затем все кнопки copy, cut и paste. Затем проверяются все кнопки Find и Find and Replace, Select all. Затем оставшиеся кнопки в контекстном меню. Проверяем работу меню Format и часть View до подсветки синтаксиса.

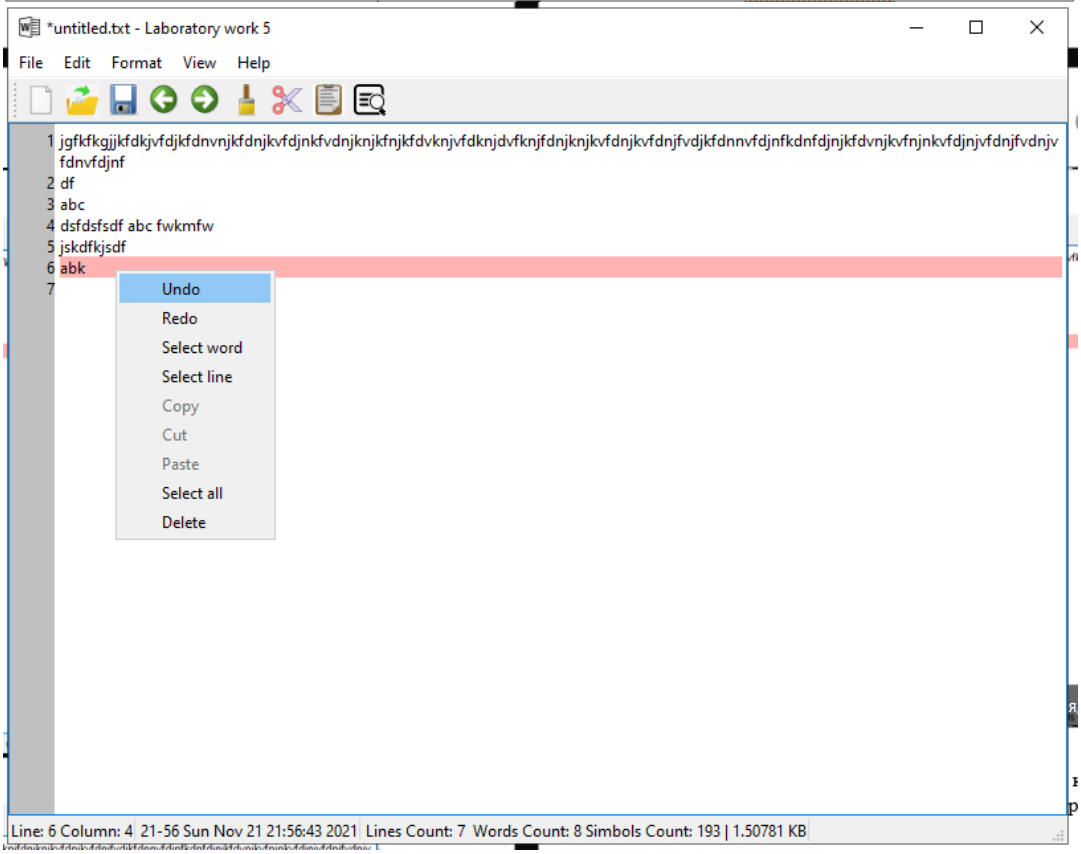


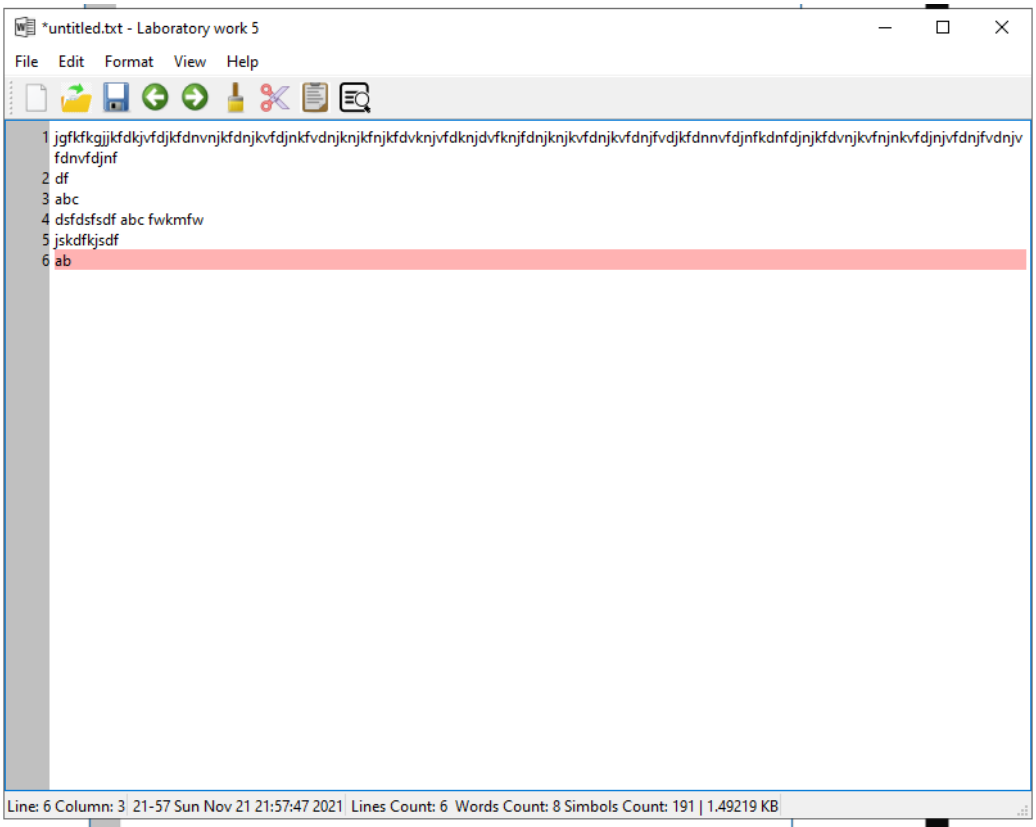


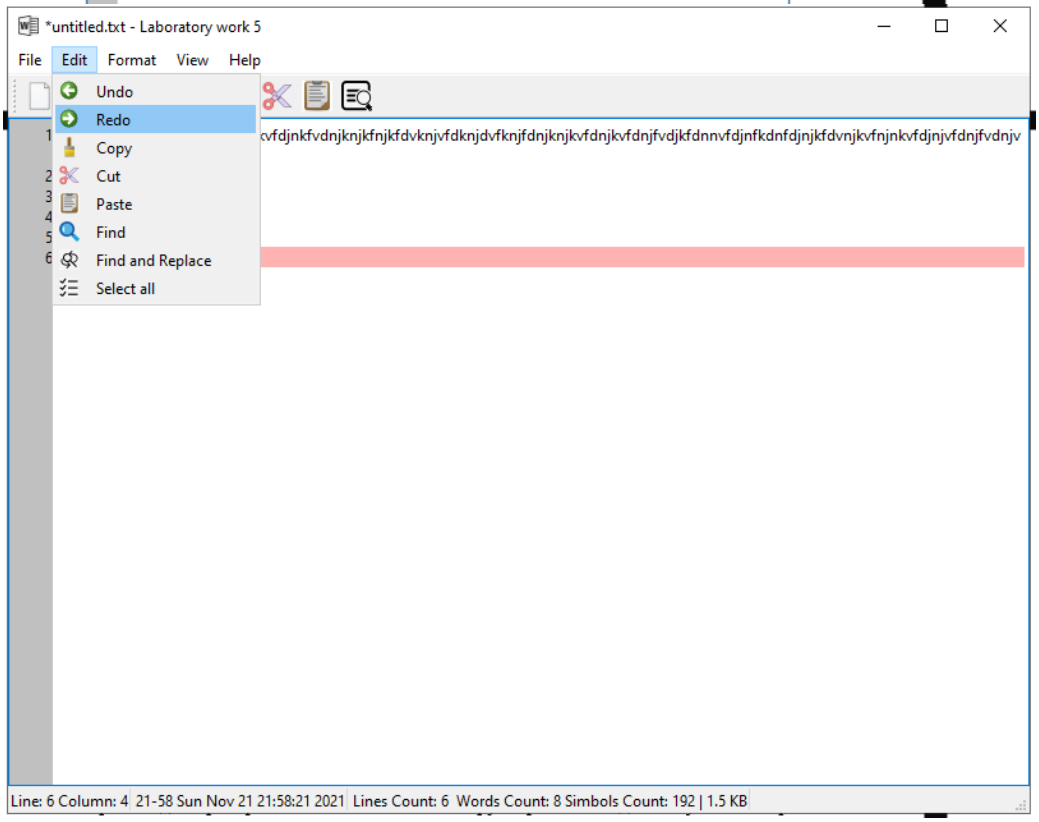

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
Undo
Redo
Copy
Cut
Paste
Find
Find and Replace
Select all
Line: 6 Column: 5 21-52 Sun Nov 21 21:52:56 2021 Lines Count: 7 Words Count: 8 Simbols Count: 194 | 1.51563 KB

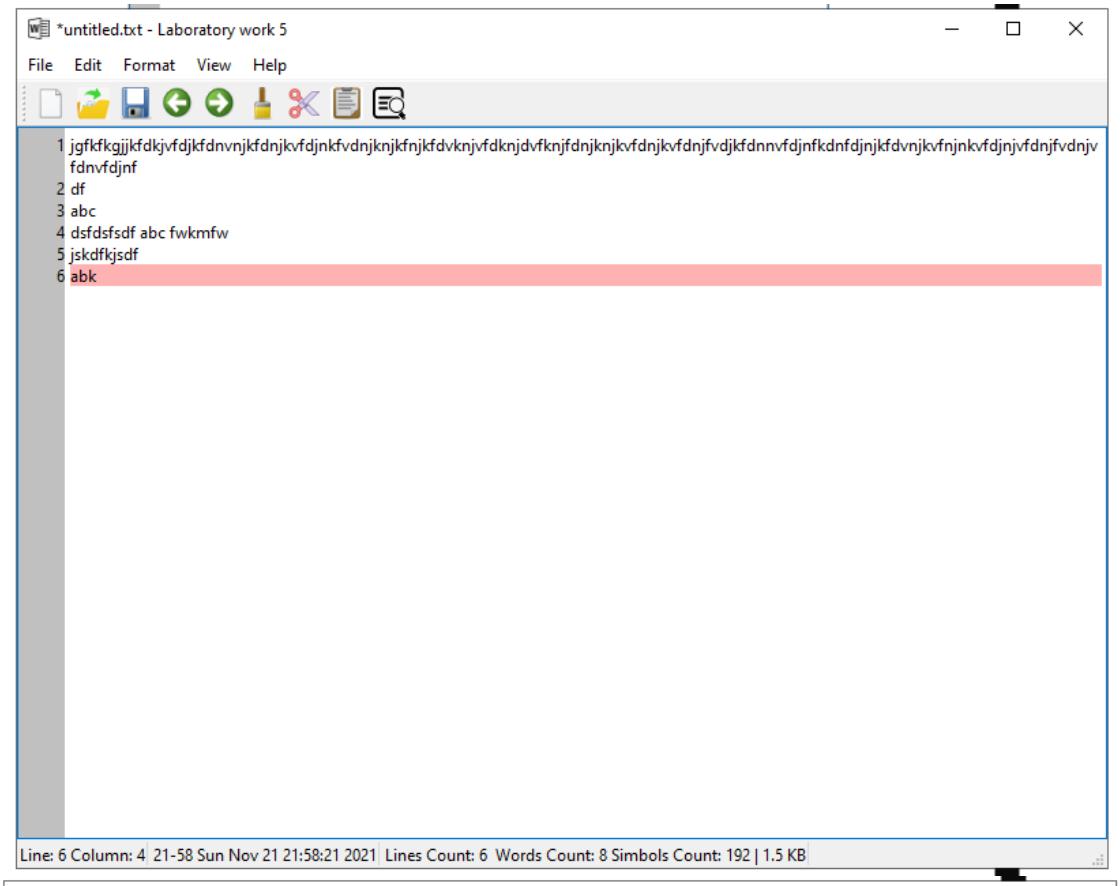

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
df
abc
dsfdsfsdf abc fwkmfw
jskdfkjsdf
abkc
Line: 6 Column: 5 21-55 Sun Nov 21 21:55:38 2021 Lines Count: 7 Words Count: 8 Simbols Count: 194 | 1.51563 KB

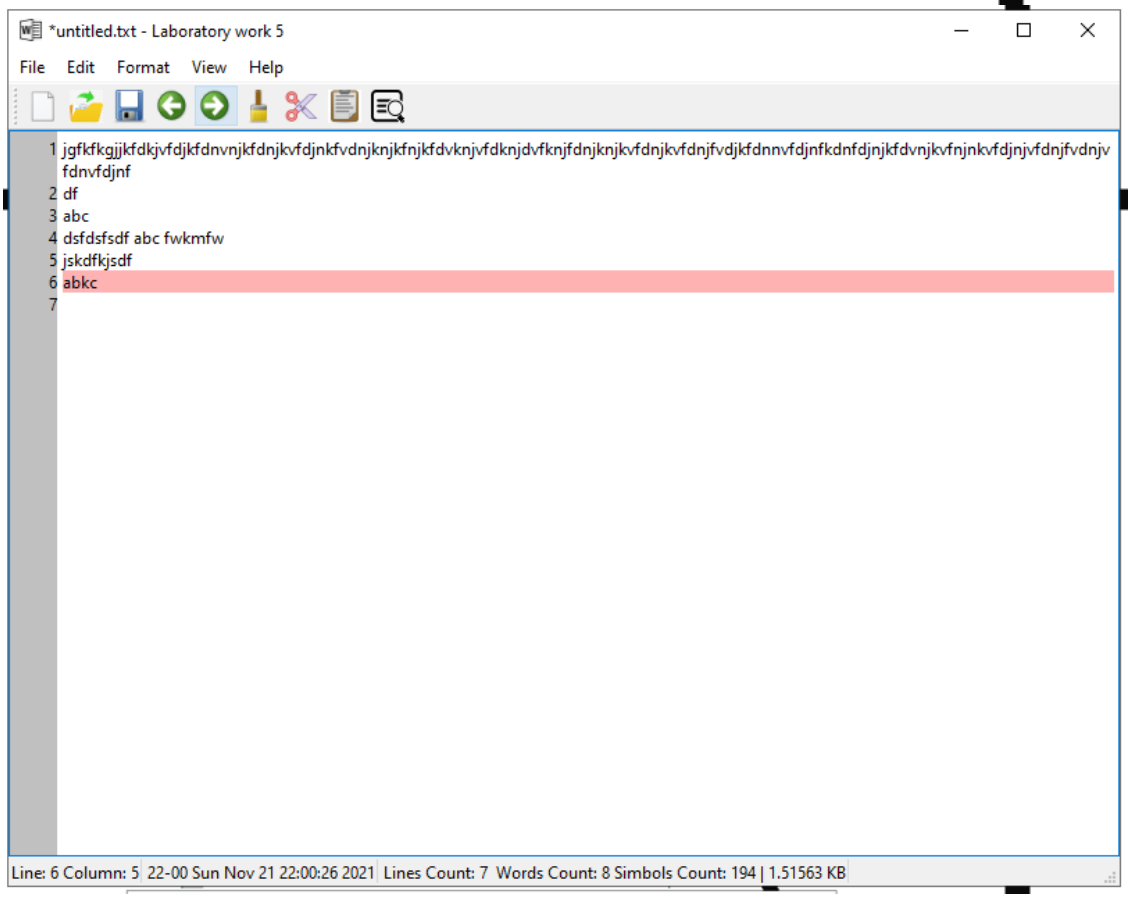

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abk
7
Line: 6 Column: 4 21-56 Sun Nov 21 21:56:43 2021 Lines Count: 7 Words Count: 8 Simbols Count: 193 | 1.50781 KB

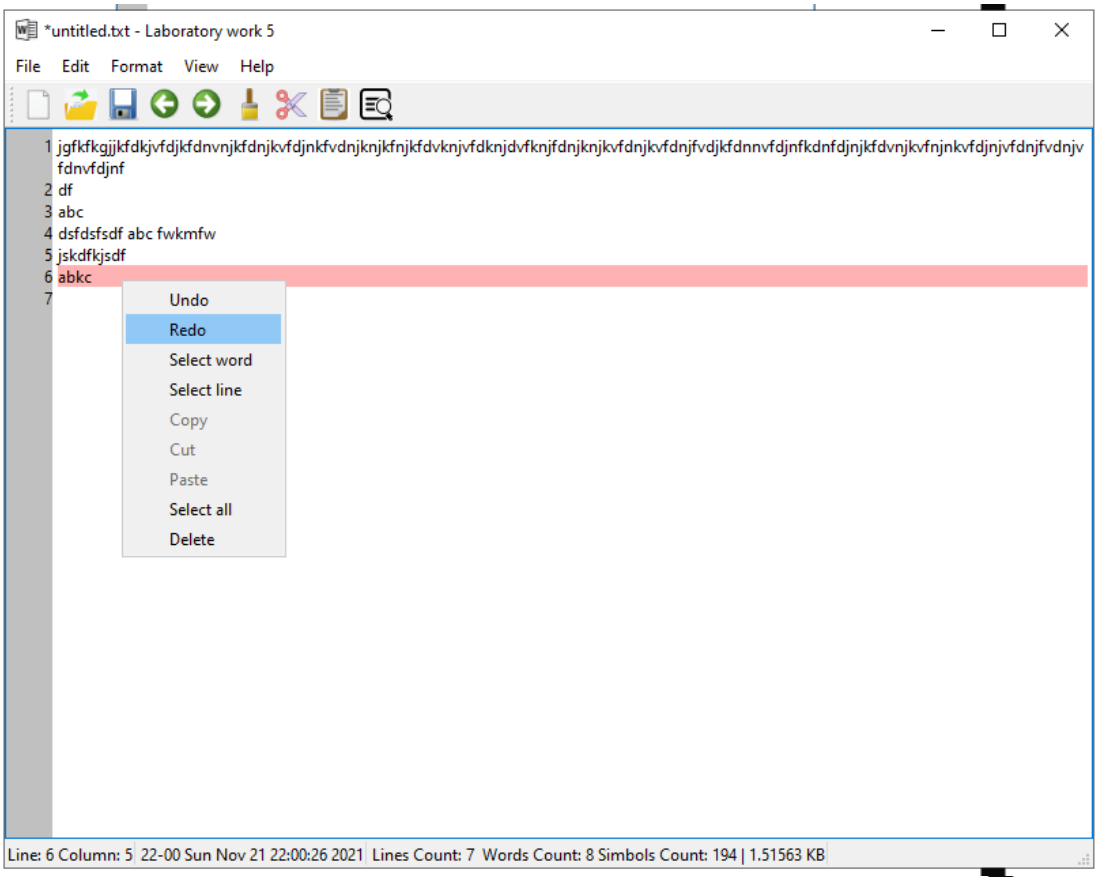

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abk
7
Undo
Redo
Select word
Select line
Copy
Cut
Paste
Select all
Delete
Line: 6 Column: 4 21-56 Sun Nov 21 21:56:43 2021 Lines Count: 7 Words Count: 8 Simbols Count: 193 | 1.50781 KB

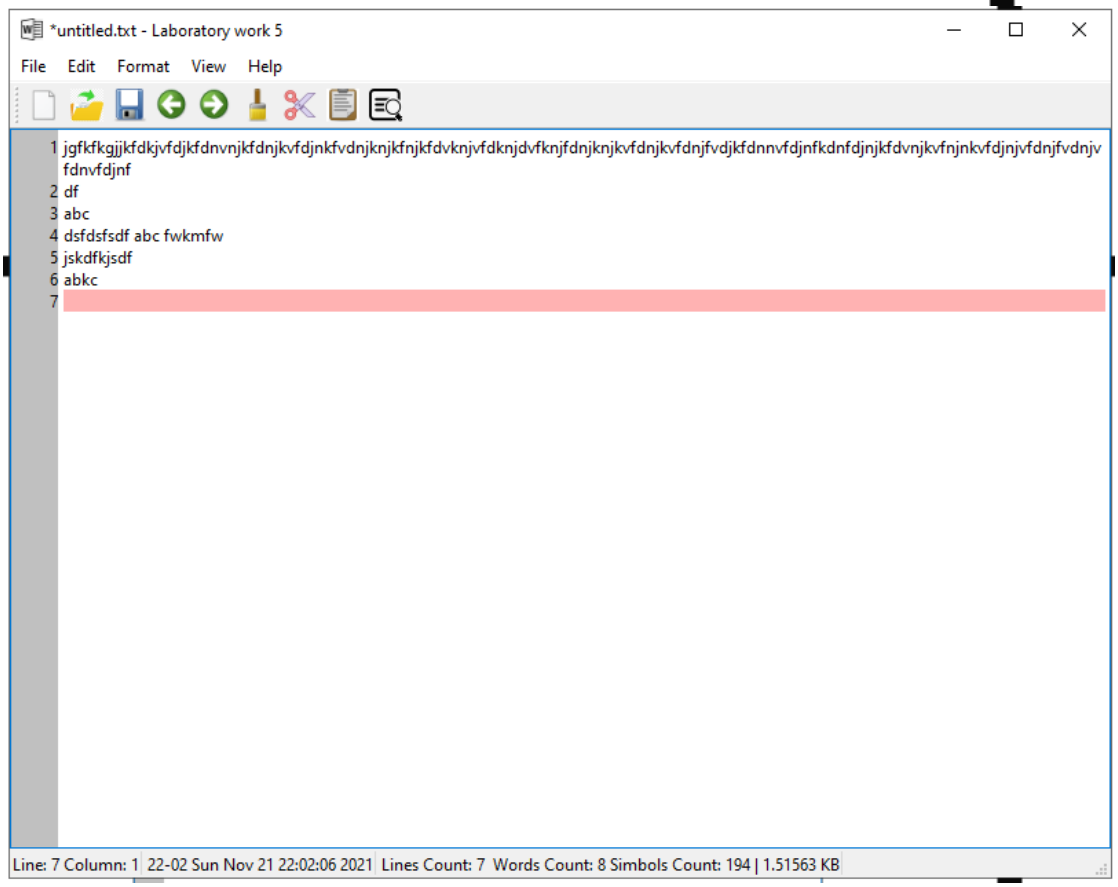

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 ab
Line: 6 Column: 3 21-57 Sun Nov 21 21:57:47 2021 Lines Count: 6 Words Count: 8 Simbols Count: 191 | 1.49219 KB

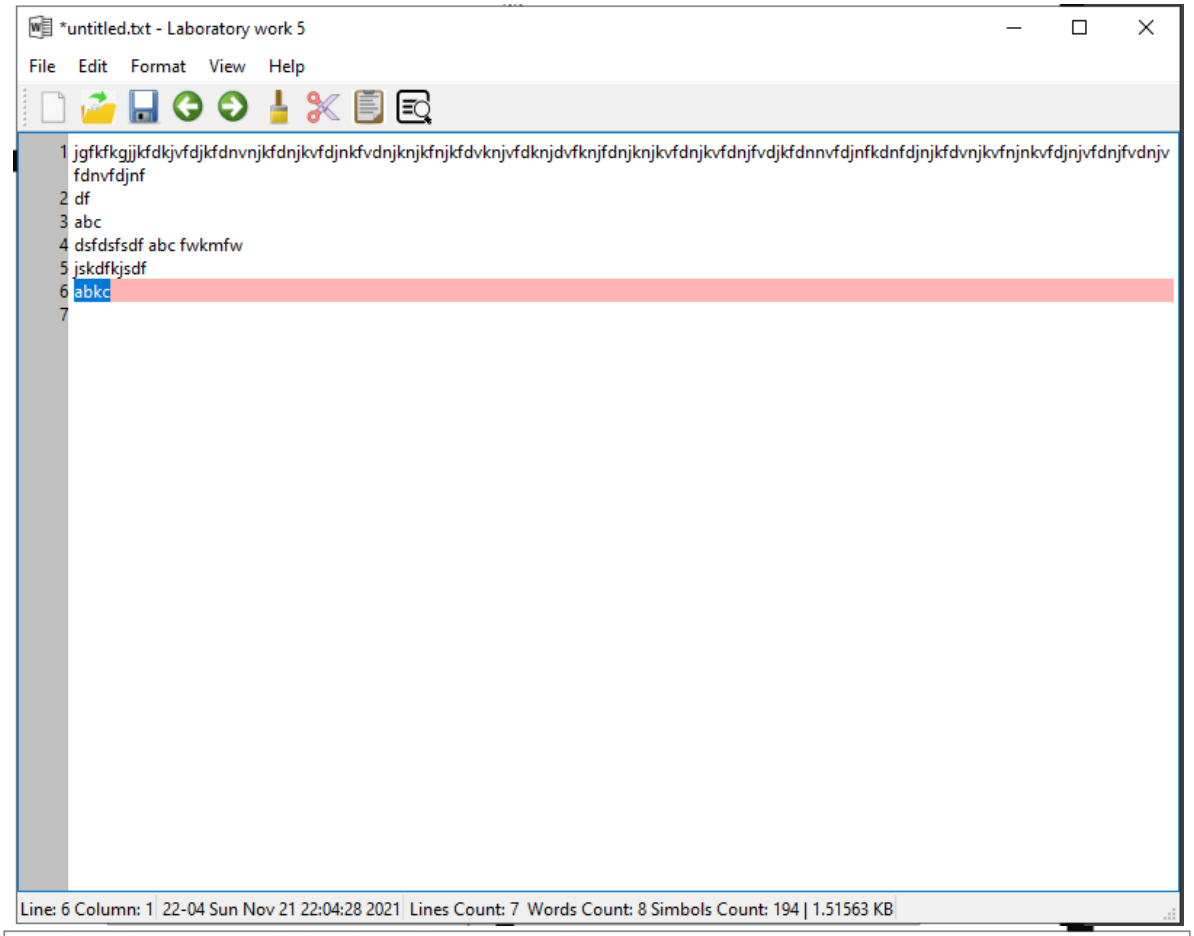

*untitled.txt - Laboratory work 5
File Edit Format View Help
Undo
Redo
Copy
Cut
Paste
Find
Find and Replace
Select all
cvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
Line: 6 Column: 4 21-58 Sun Nov 21 21:58:21 2021 Lines Count: 6 Words Count: 8 Simbols Count: 192 | 1.5 KB

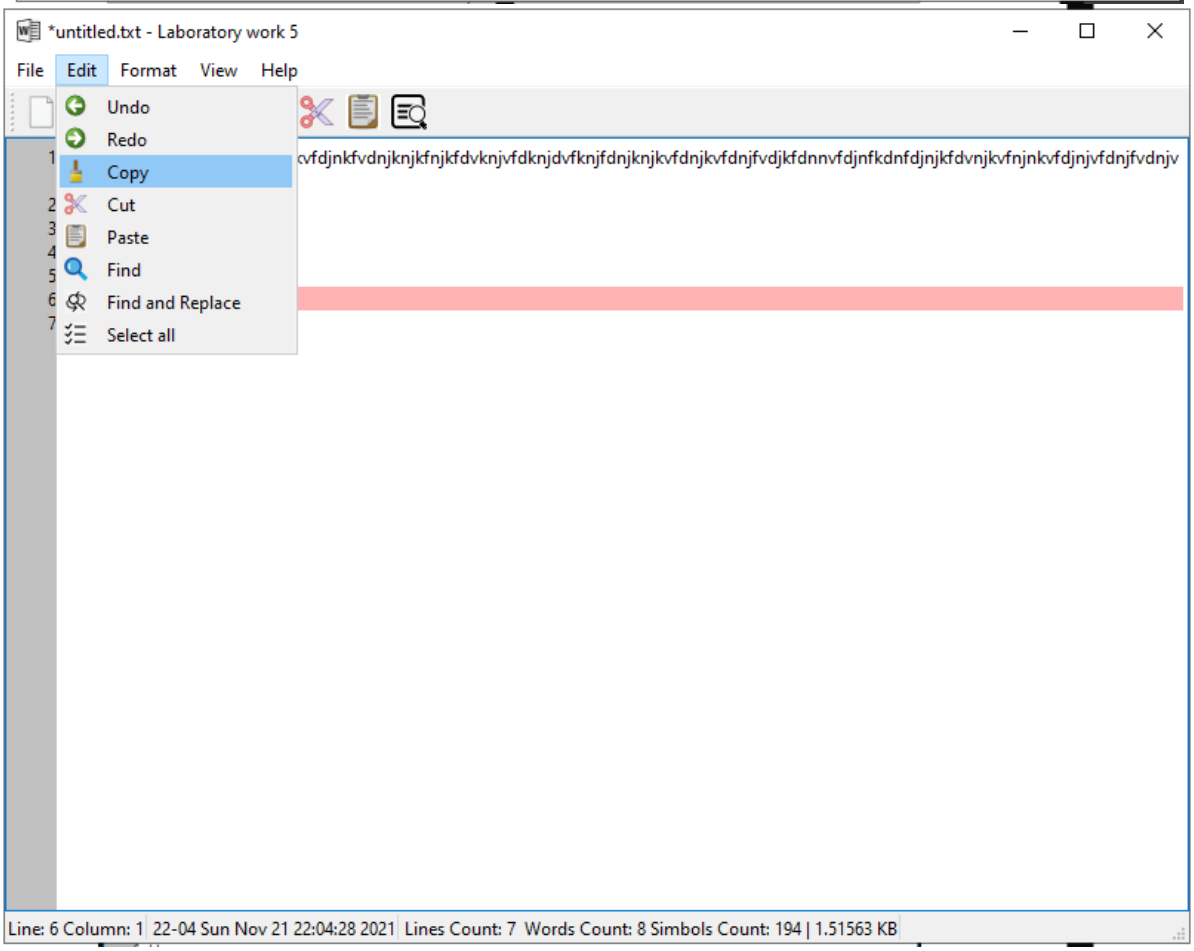

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abk
Line: 6 Column: 4 21-58 Sun Nov 21 21:58:21 2021 Lines Count: 6 Words Count: 8 Simbols Count: 192 | 1.5 KB

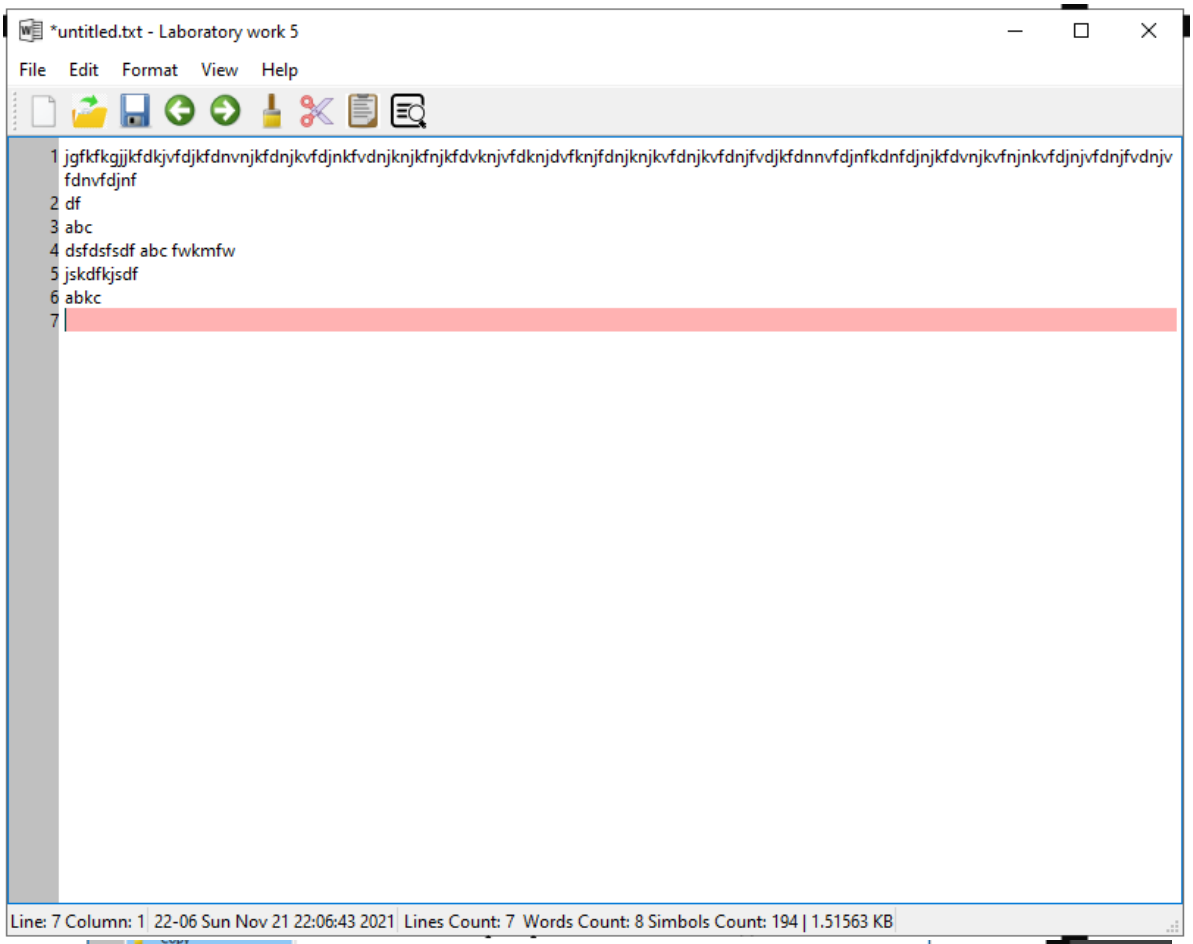

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7
Line: 6 Column: 5 22-00 Sun Nov 21 22:00:26 2021 Lines Count: 7 Words Count: 8 Simbols Count: 194 | 1.51563 KB

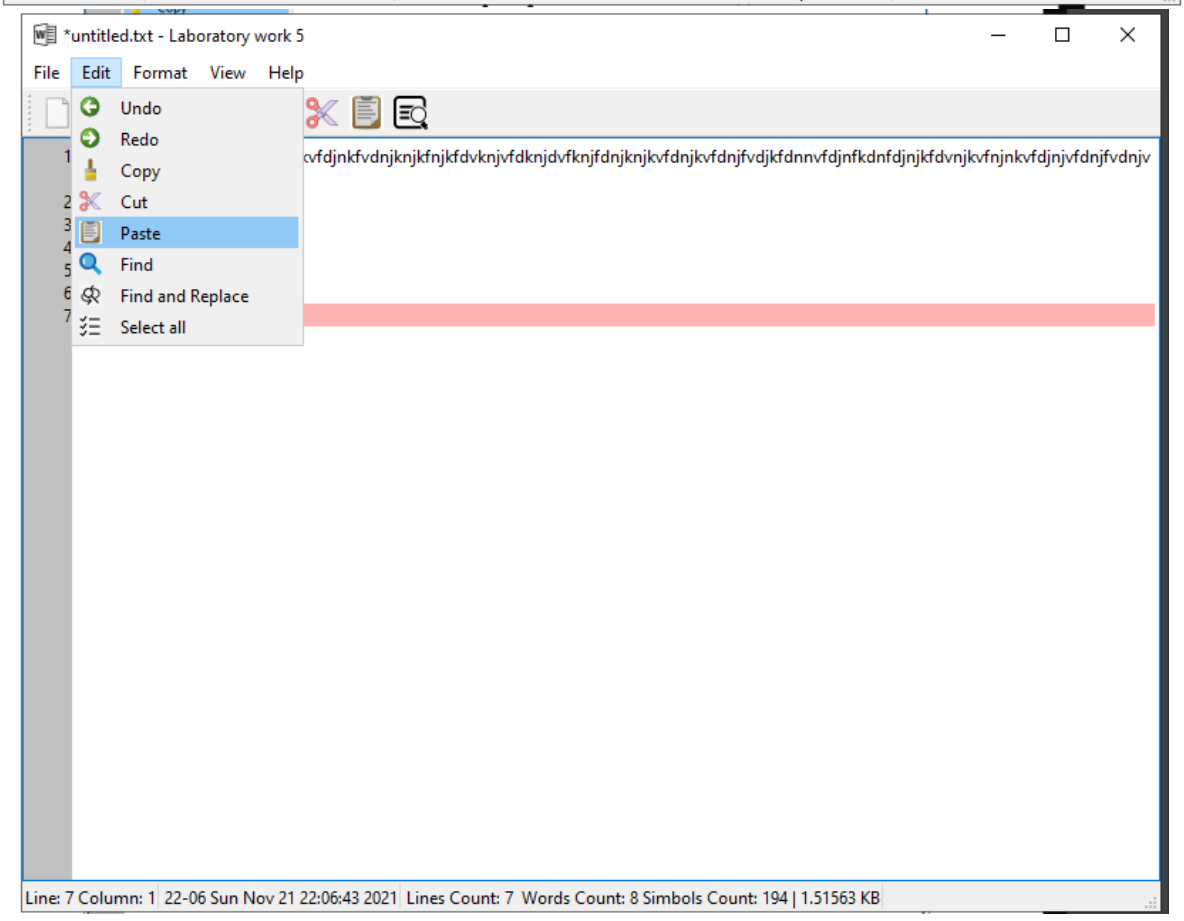

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7
Undo
Redo
Select word
Select line
Copy
Cut
Paste
Select all
Delete
Line: 6 Column: 5 22-00 Sun Nov 21 22:00:26 2021 Lines Count: 7 Words Count: 8 Simbols Count: 194 | 1.51563 KB

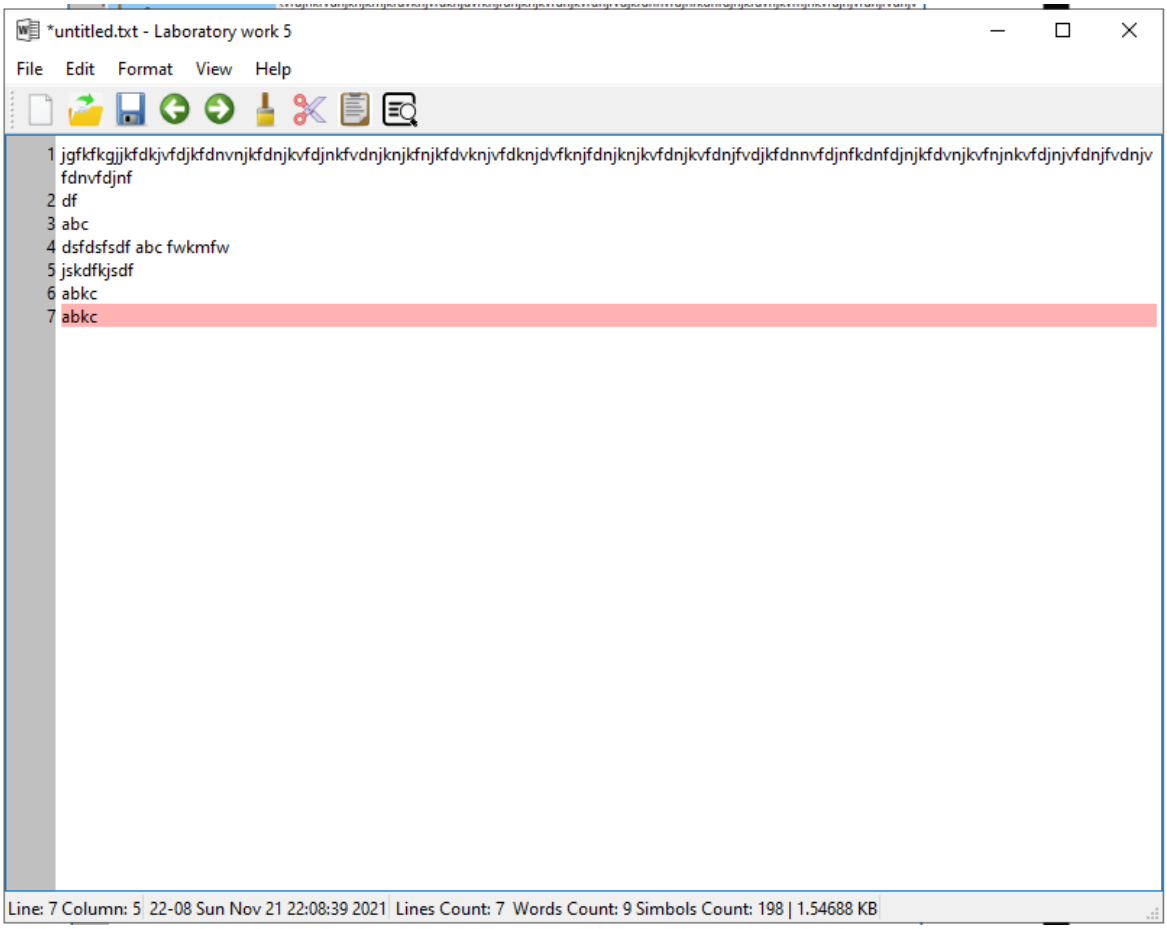

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7
Line: 7 Column: 1 22-02 Sun Nov 21 22:02:06 2021 Lines Count: 7 Words Count: 8 Simbols Count: 194 | 1.51563 KB

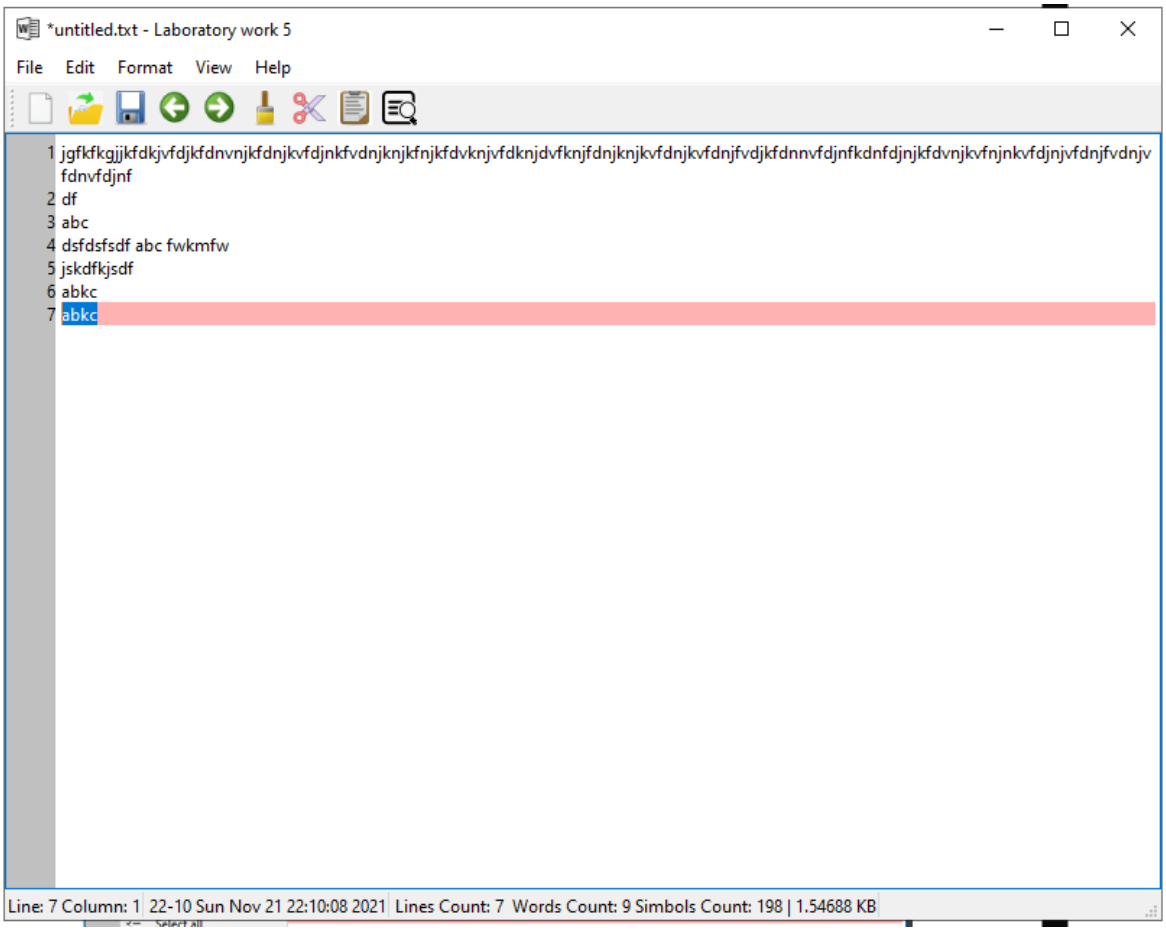

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7
Line: 6 Column: 1 22-04 Sun Nov 21 22:04:28 2021 Lines Count: 7 Words Count: 8 Simbols Count: 194 | 1.51563 KB

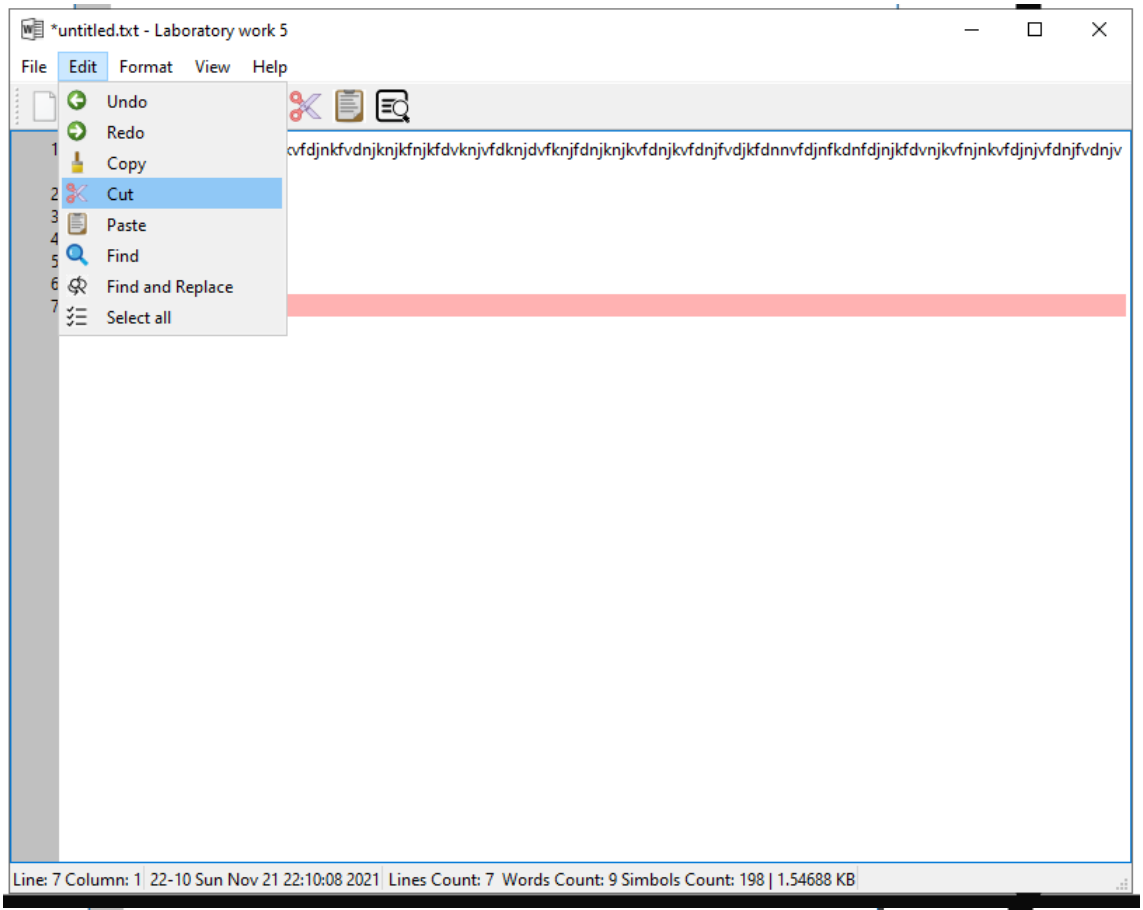

*untitled.txt - Laboratory work 5
File Edit Format View Help
Undo
Redo
Copy
Cut
Paste
Find
Find and Replace
Select all
cvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
Line: 6 Column: 1 22-04 Sun Nov 21 22:04:28 2021 Lines Count: 7 Words Count: 8 Simbols Count: 194 | 1.51563 KB

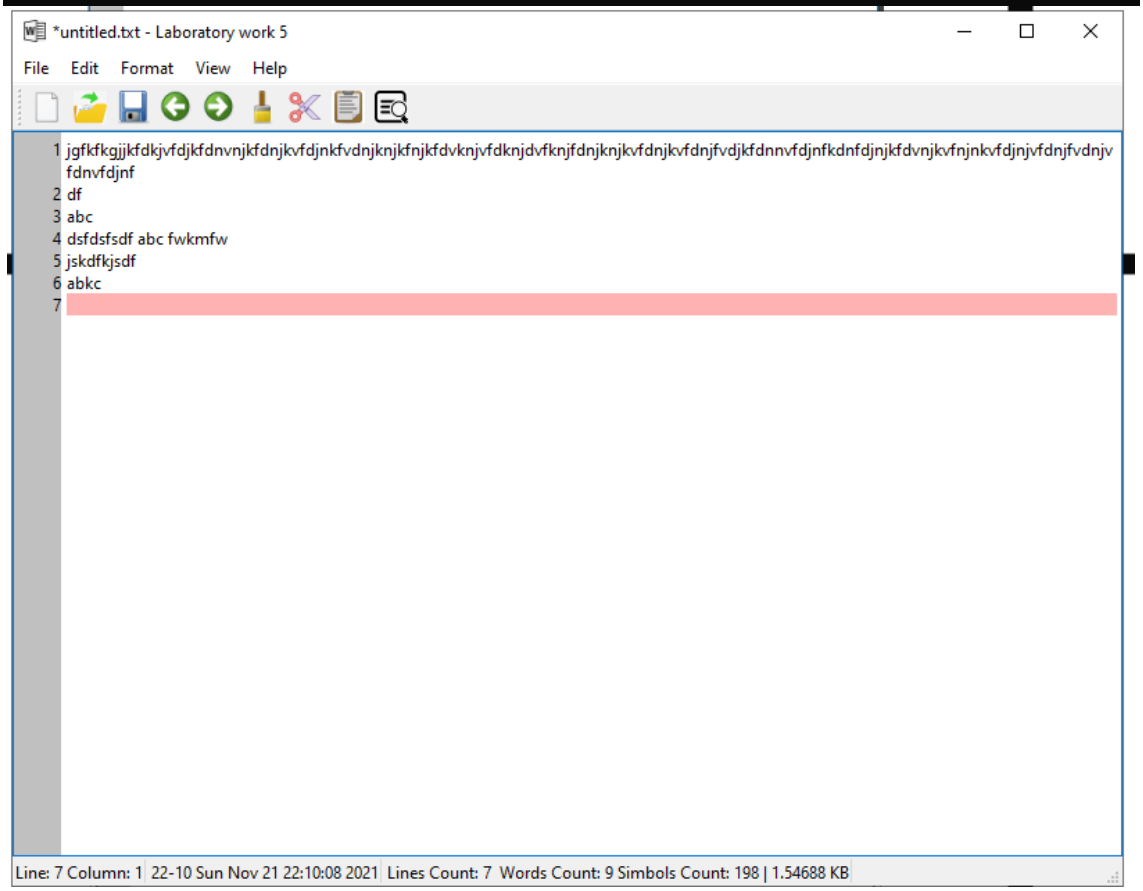

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7
Line: 7 Column: 1 22-06 Sun Nov 21 22:06:43 2021 Lines Count: 7 Words Count: 8 Simbols Count: 194 | 1.51563 KB

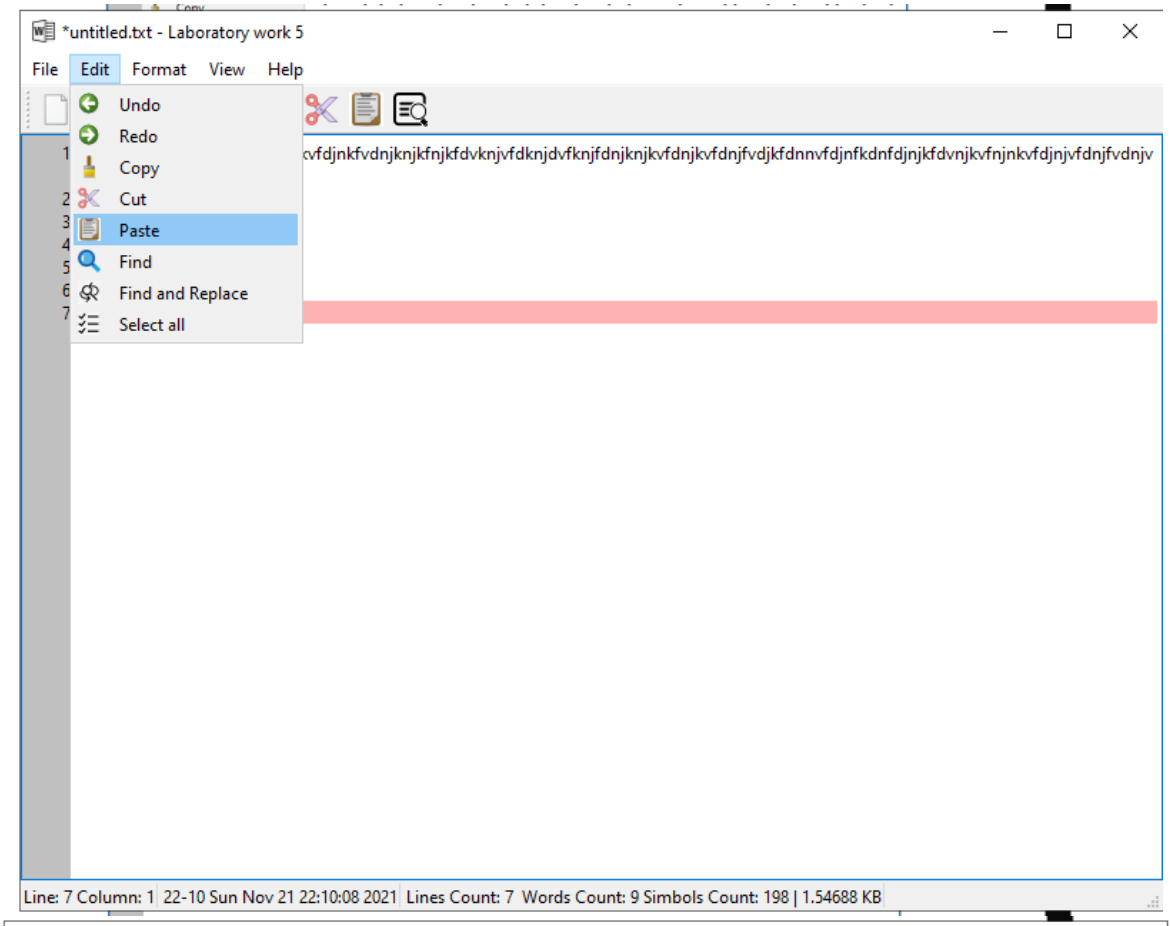

*untitled.txt - Laboratory work 5
File Edit Format View Help
Undo
Redo
Copy
Cut
Paste
Find
Find and Replace
Select all
cvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
Line: 7 Column: 1 22-06 Sun Nov 21 22:06:43 2021 Lines Count: 7 Words Count: 8 Simbols Count: 194 | 1.51563 KB

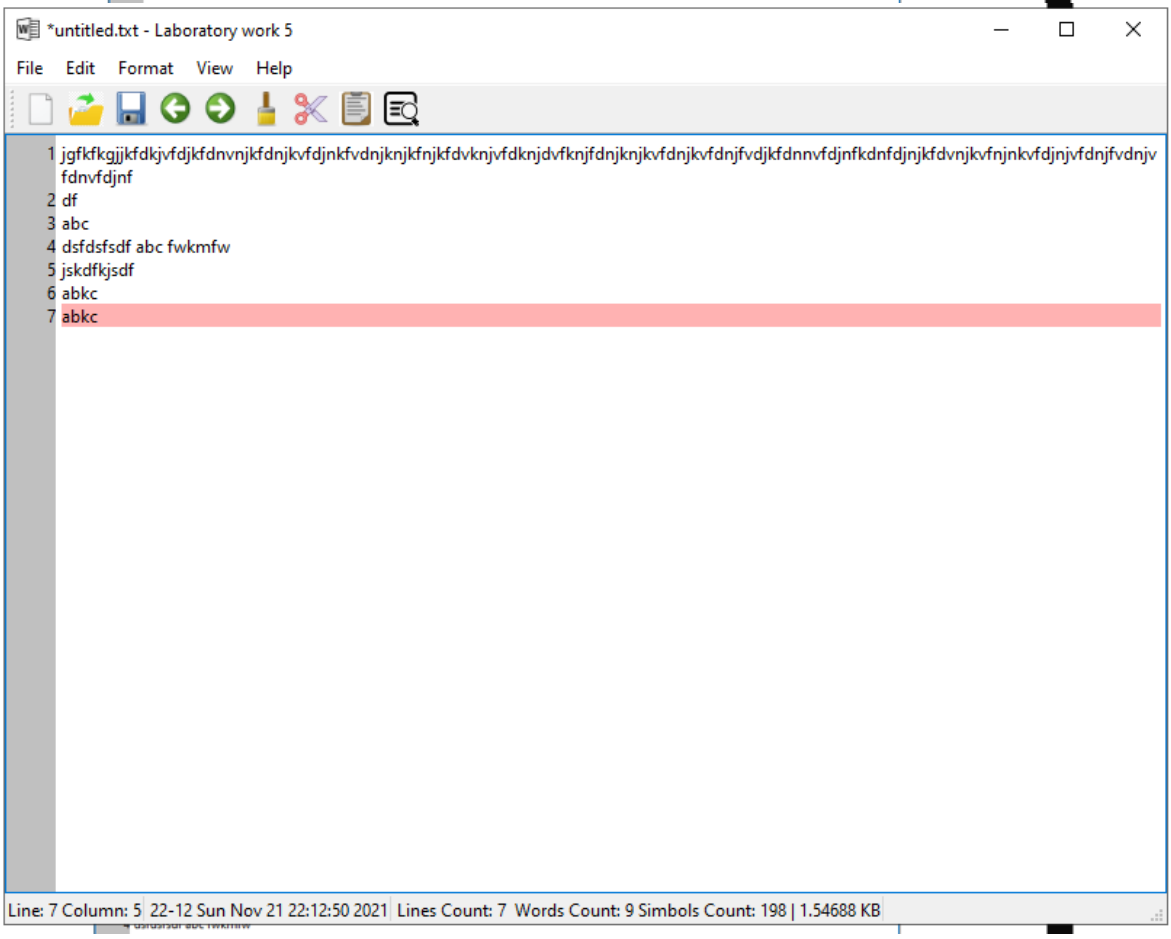

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkc
Line: 7 Column: 5 22-08 Sun Nov 21 22:08:39 2021 Lines Count: 7 Words Count: 9 Simbols Count: 198 | 1.54688 KB

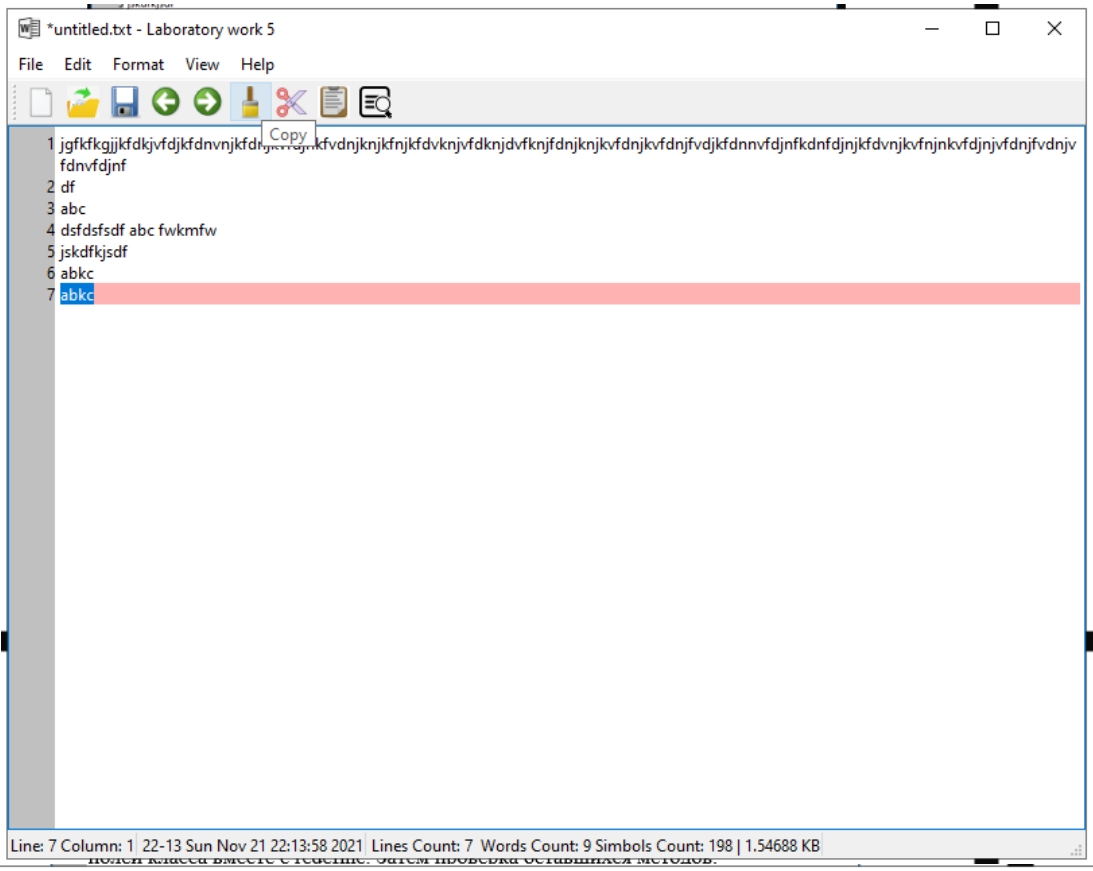

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkc
Line: 7 Column: 1 22-10 Sun Nov 21 22:10:08 2021 Lines Count: 7 Words Count: 9 Simbols Count: 198 | 1.54688 KB

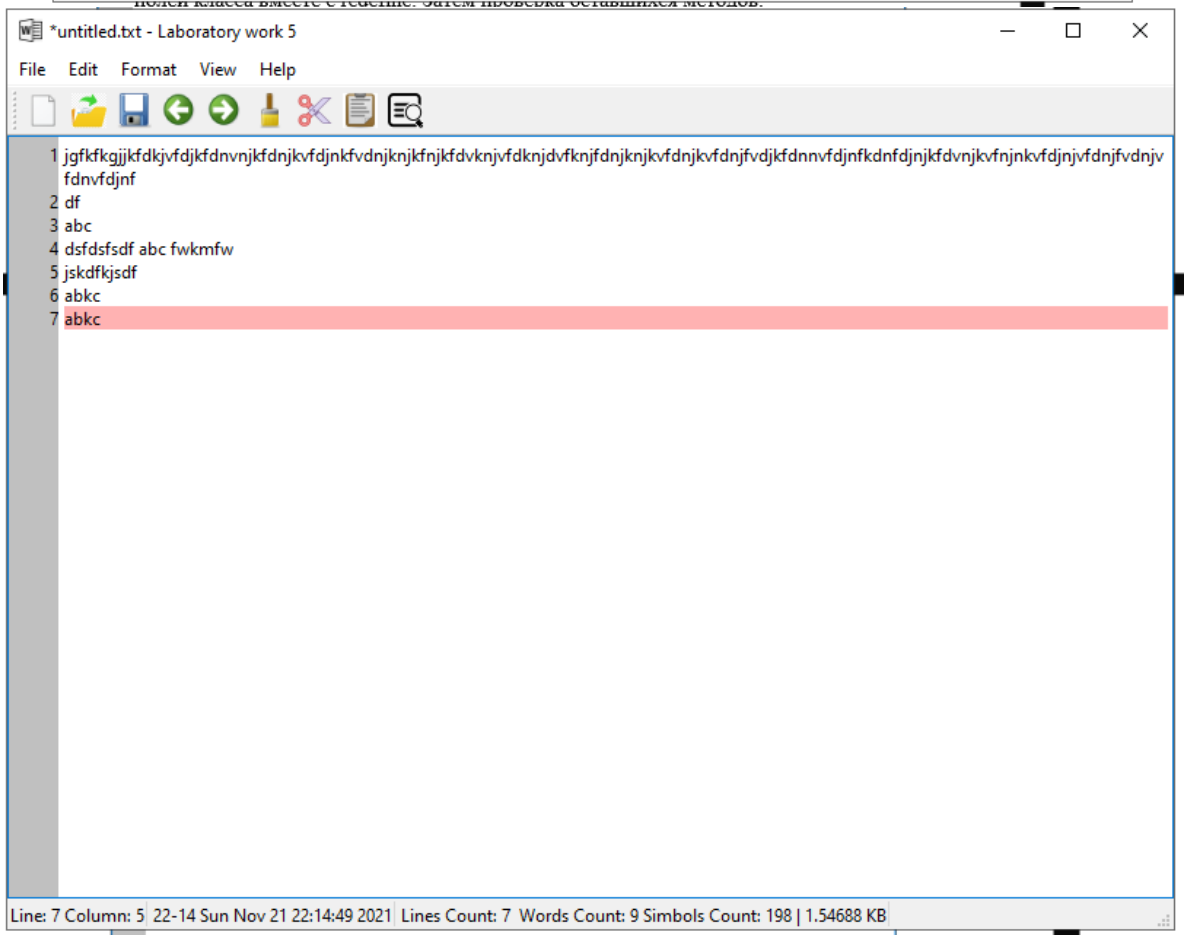

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
Undo
Redo
Copy
Cut
Paste
Find
Find and Replace
Select all
cvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
Line: 7 Column: 1 22-10 Sun Nov 21 22:10:08 2021 Lines Count: 7 Words Count: 9 Simbols Count: 198 | 1.54688 KB

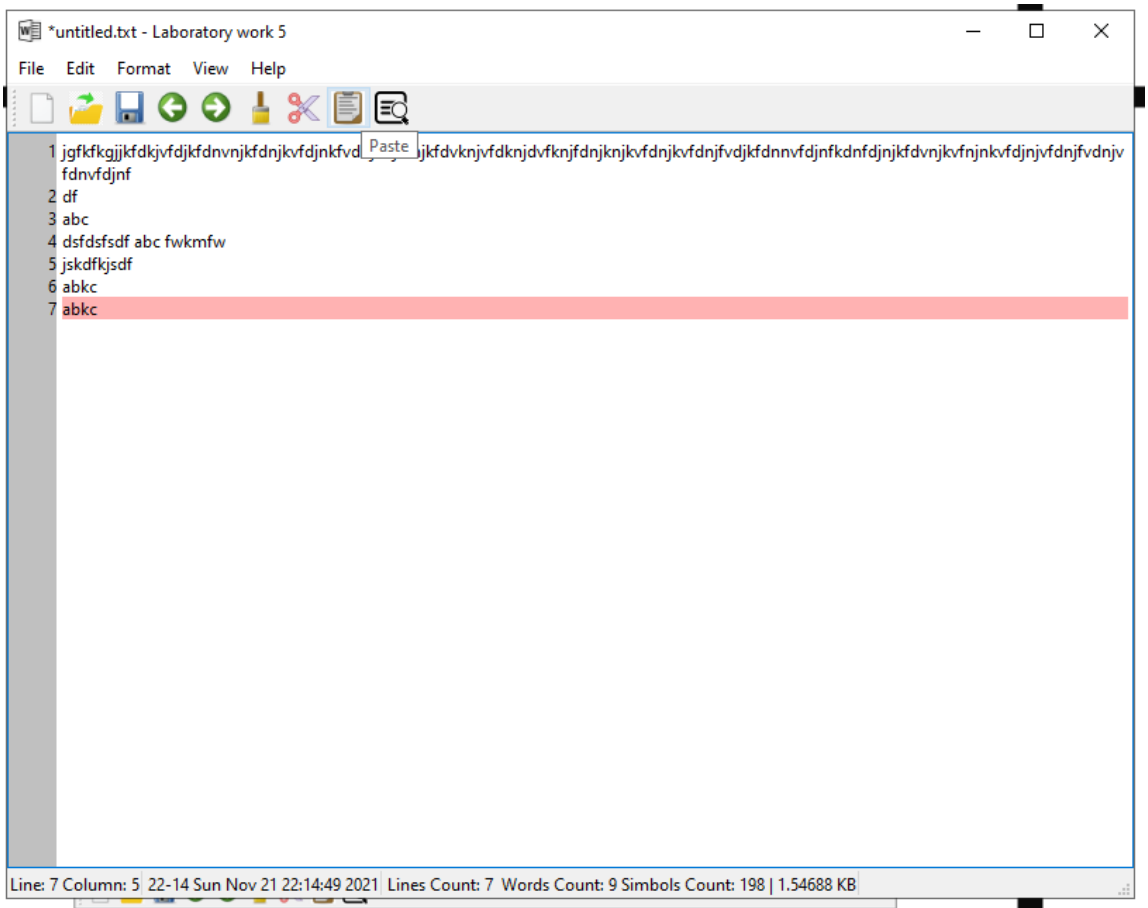

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7
Line: 7 Column: 1 22-10 Sun Nov 21 22:10:08 2021 Lines Count: 7 Words Count: 9 Simbols Count: 198 | 1.54688 KB

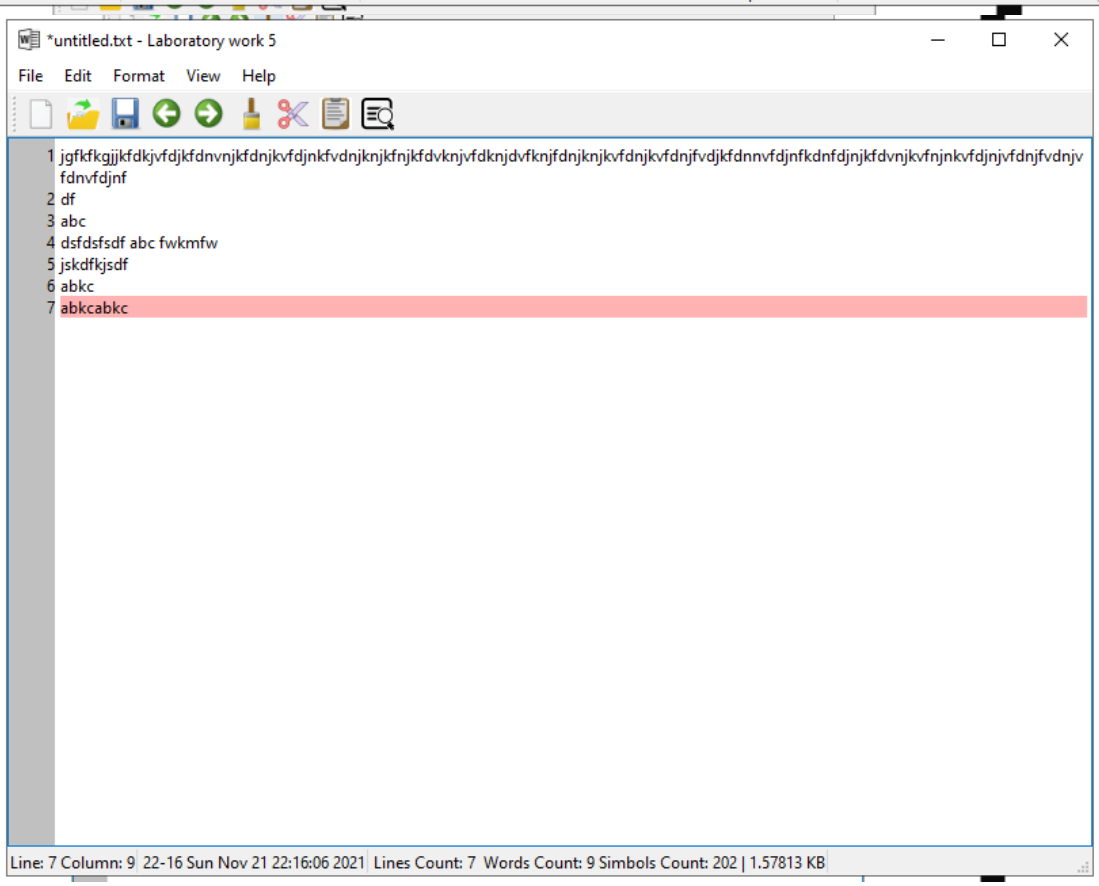

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
Undo
Redo
Copy
Cut
Paste
Find
Find and Replace
Select all
cvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
Line: 7 Column: 1
22-10 Sun Nov 21 22:10:08 2021
Lines Count: 7
Words Count: 9
Simbols Count: 198 | 1.54688 KB

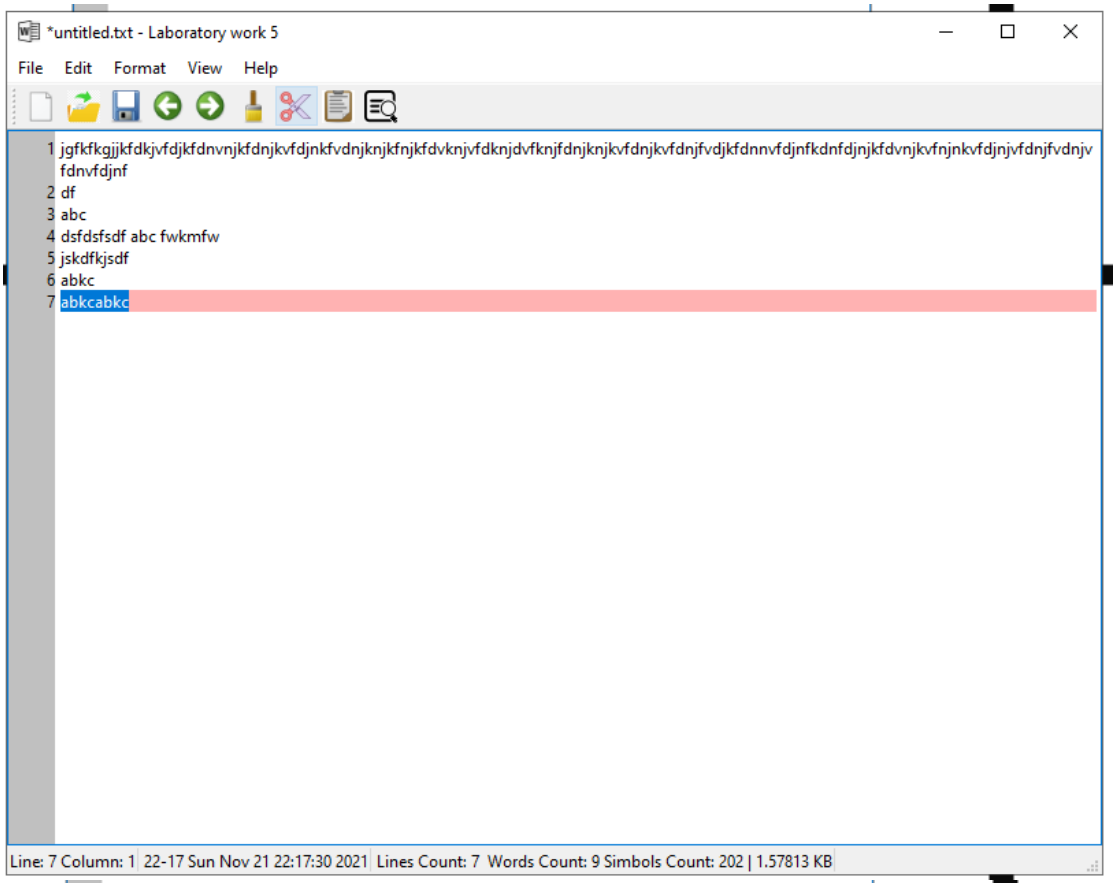

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkc
Line: 7 Column: 5
22-12 Sun Nov 21 22:12:50 2021
Lines Count: 7
Words Count: 9
Simbols Count: 198 | 1.54688 KB

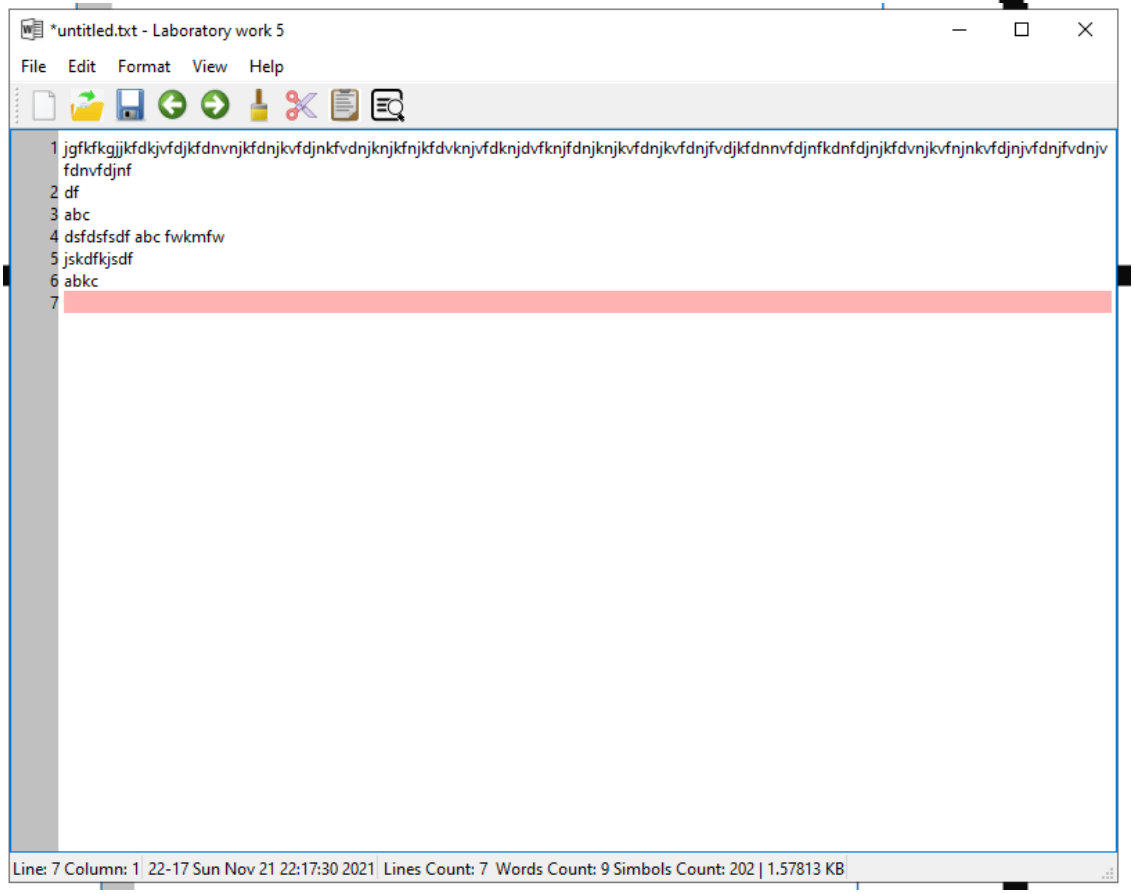

*untitled.txt - Laboratory work 5
File Edit Format View Help
Copy
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkc
Line: 7 Column: 1 22-13 Sun Nov 21 22:13:58 2021 Lines Count: 7 Words Count: 9 Simbols Count: 198 | 1.54688 KB

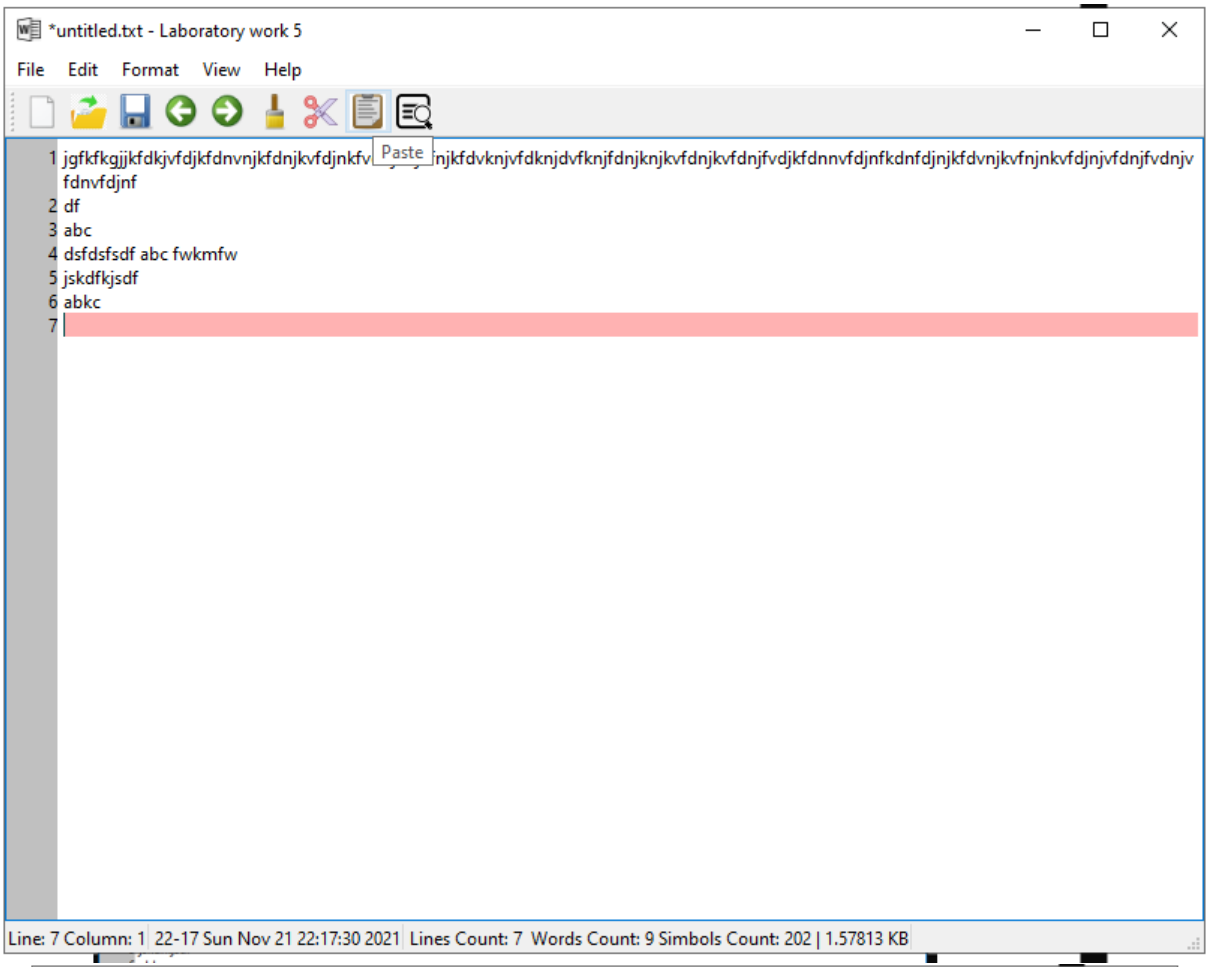

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkc
Line: 7 Column: 5 22-14 Sun Nov 21 22:14:49 2021 Lines Count: 7 Words Count: 9 Simbols Count: 198 | 1.54688 KB

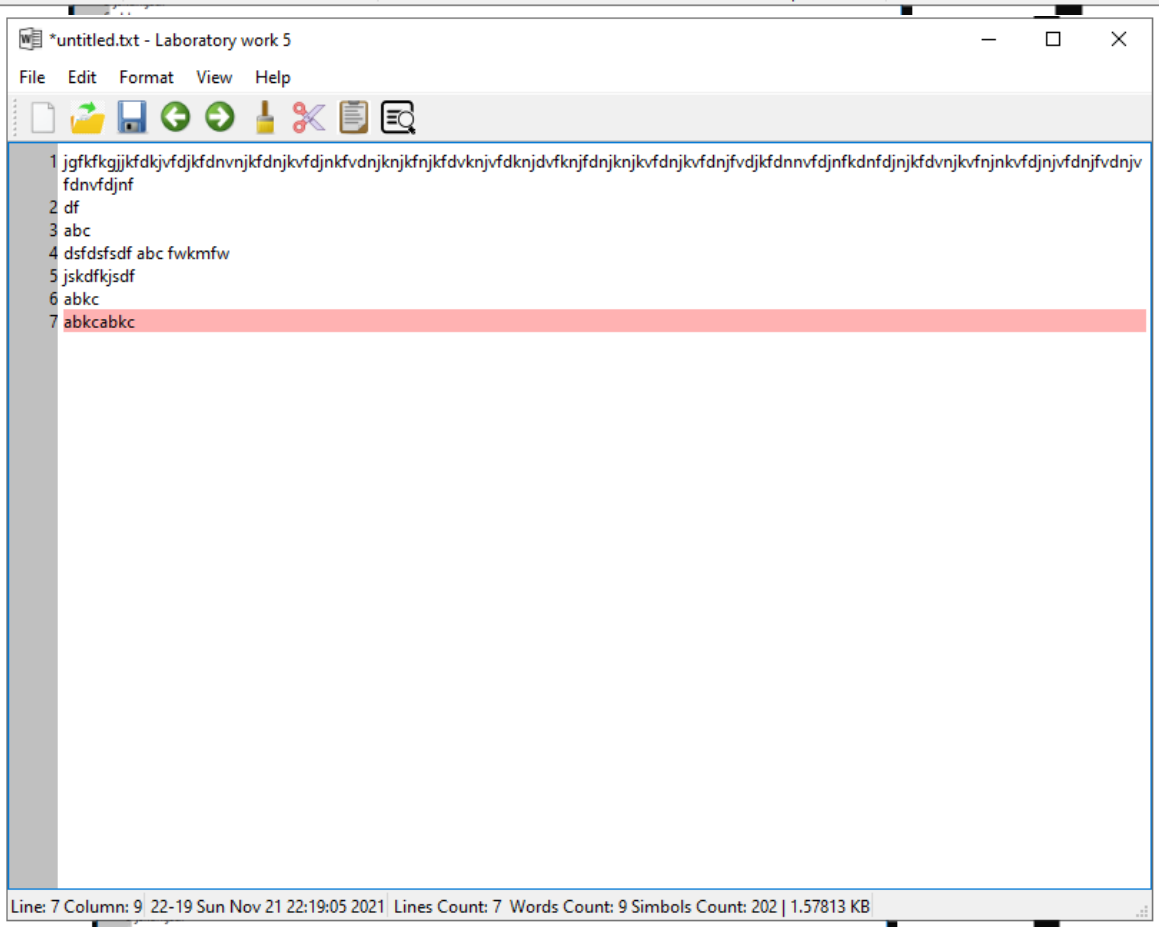

*untitled.txt - Laboratory work 5
File Edit Format View Help
Paste
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkc
Line: 7 Column: 5 22-14 Sun Nov 21 22:14:49 2021 Lines Count: 7 Words Count: 9 Simbols Count: 198 | 1.54688 KB

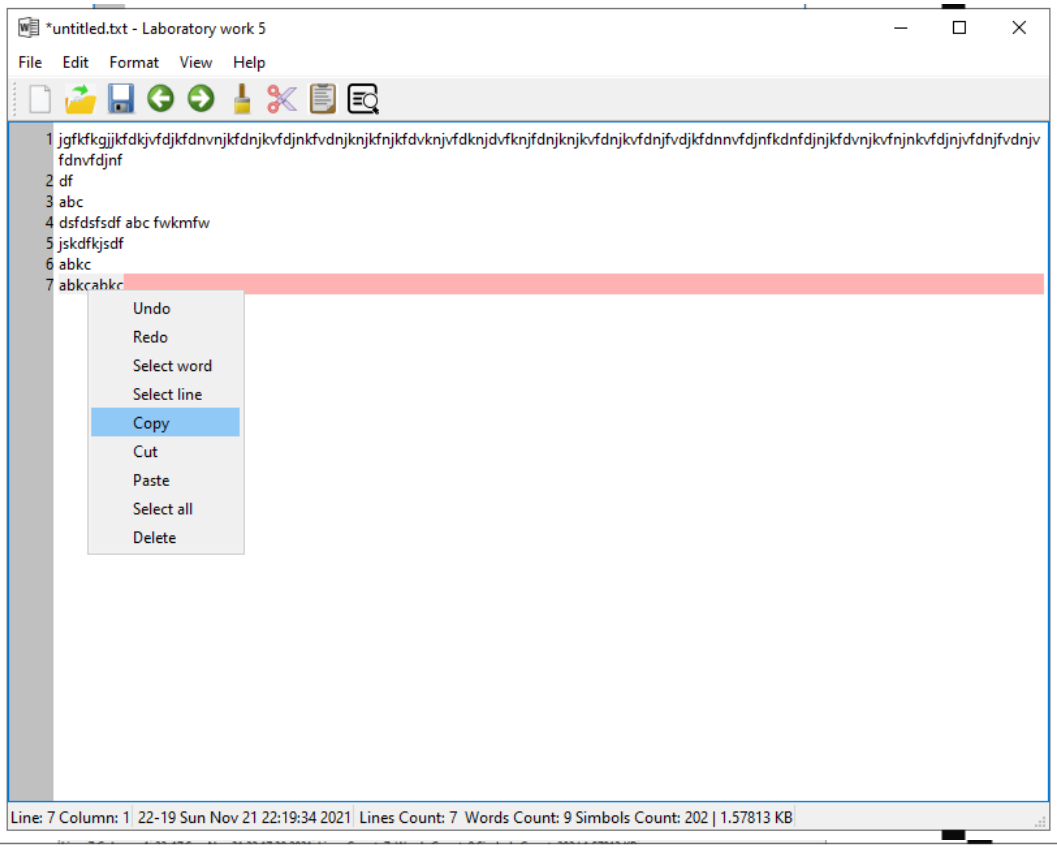

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkc
Line: 7 Column: 9 22-16 Sun Nov 21 22:16:06 2021 Lines Count: 7 Words Count: 9 Simbols Count: 202 | 1.57813 KB

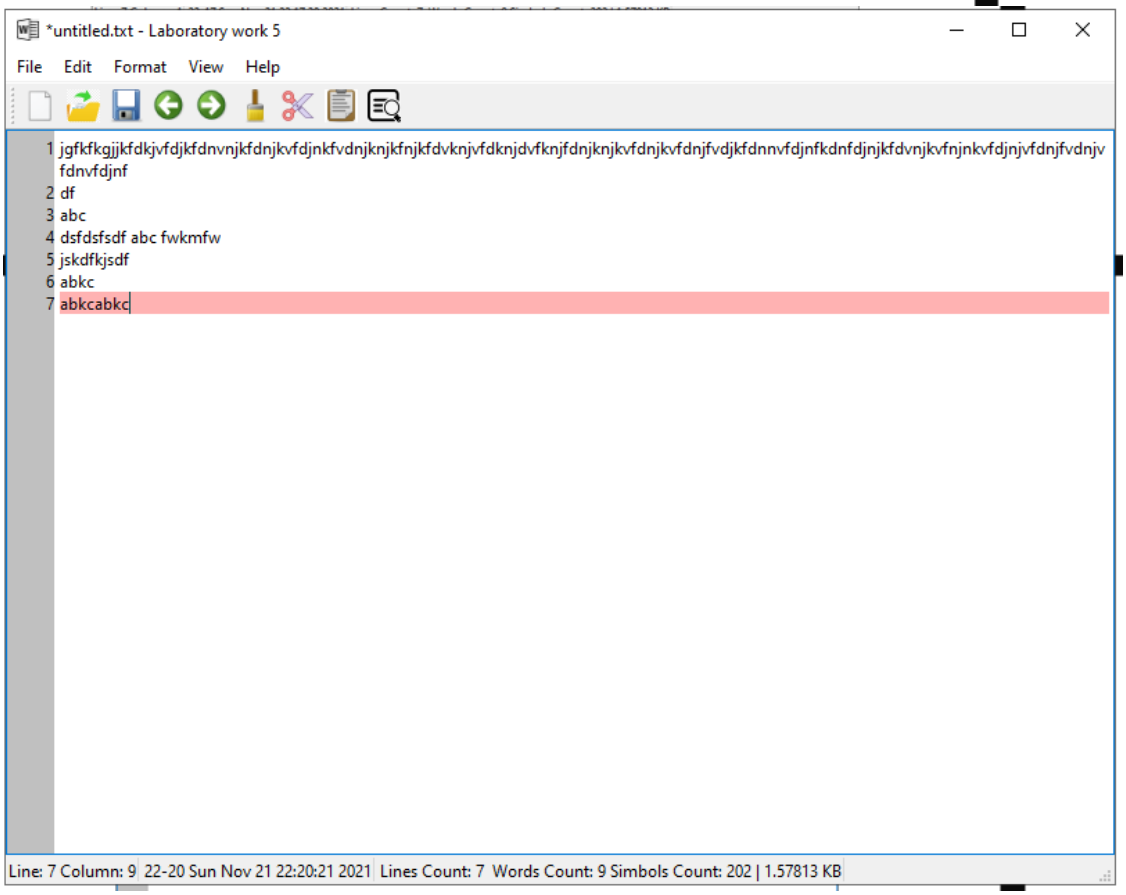

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkc
Line: 7 Column: 1 22-17 Sun Nov 21 22:17:30 2021 Lines Count: 7 Words Count: 9 Simbols Count: 202 | 1.57813 KB

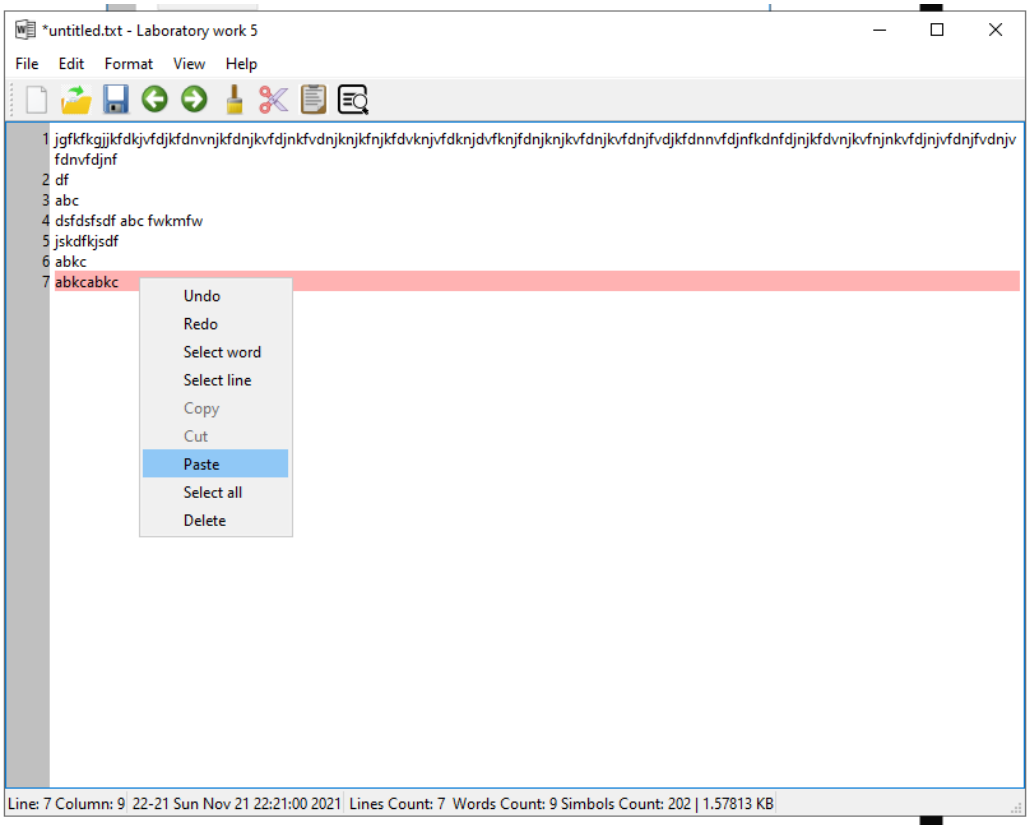

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7
Line: 7 Column: 1 22-17 Sun Nov 21 22:17:30 2021 Lines Count: 7 Words Count: 9 Simbols Count: 202 | 1.57813 KB

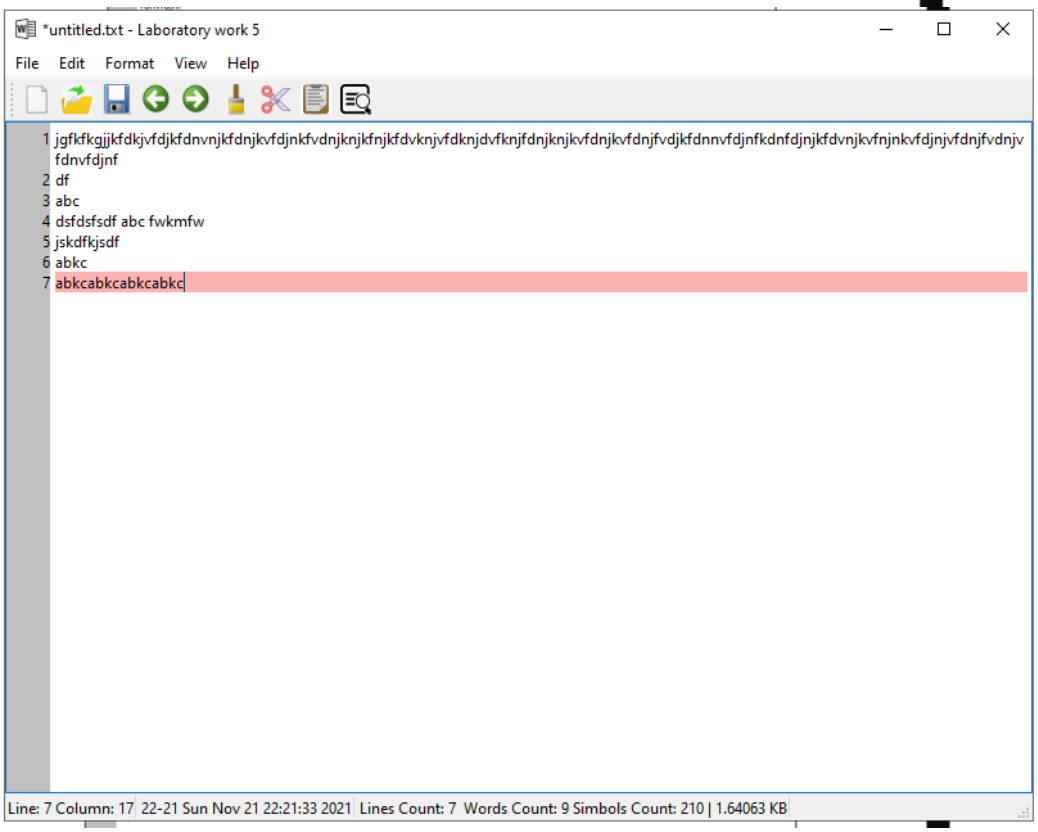

*untitled.txt - Laboratory work 5
File Edit Format View Help
Paste
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfv
njkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7
Line: 7 Column: 1 22-17 Sun Nov 21 22:17:30 2021 Lines Count: 7 Words Count: 9 Simbols Count: 202 | 1.57813 KB

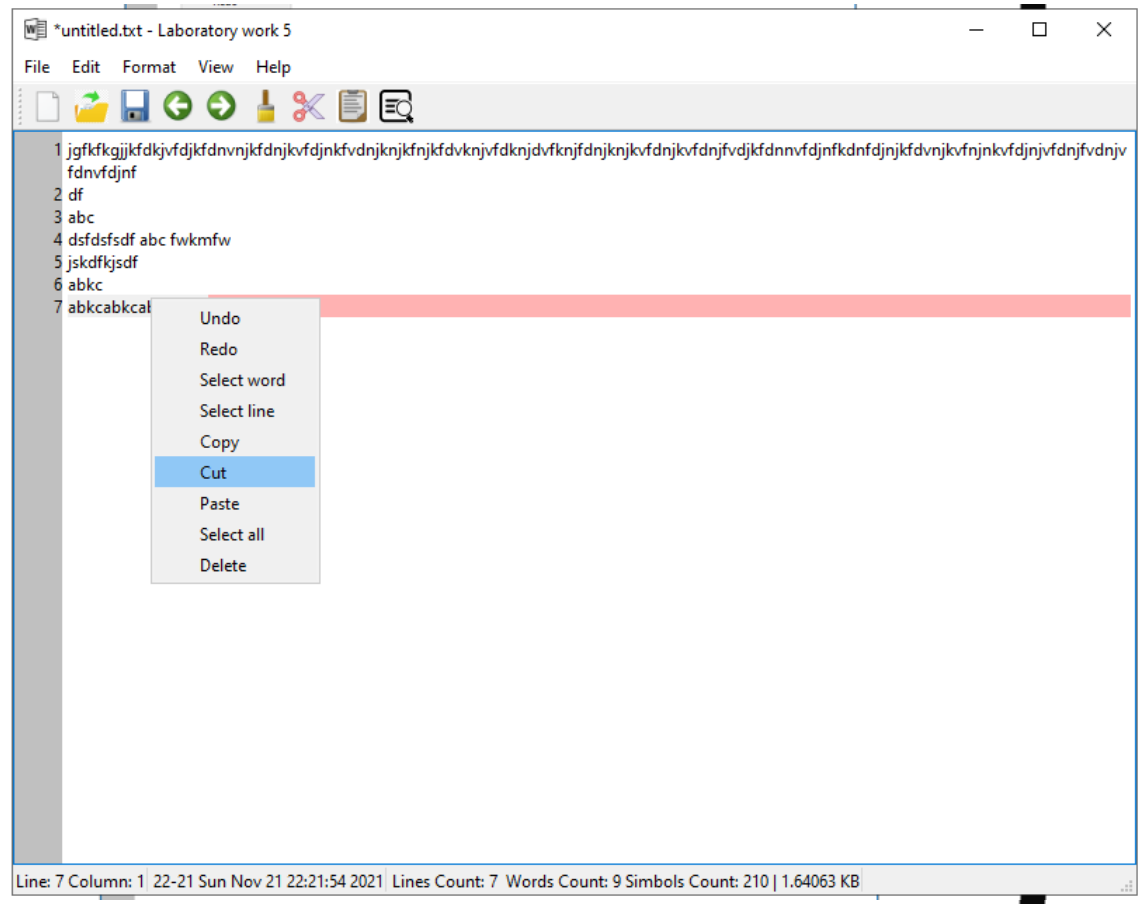

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkc
Line: 7 Column: 9 22-19 Sun Nov 21 22:19:05 2021 Lines Count: 7 Words Count: 9 Simbols Count: 202 | 1.57813 KB

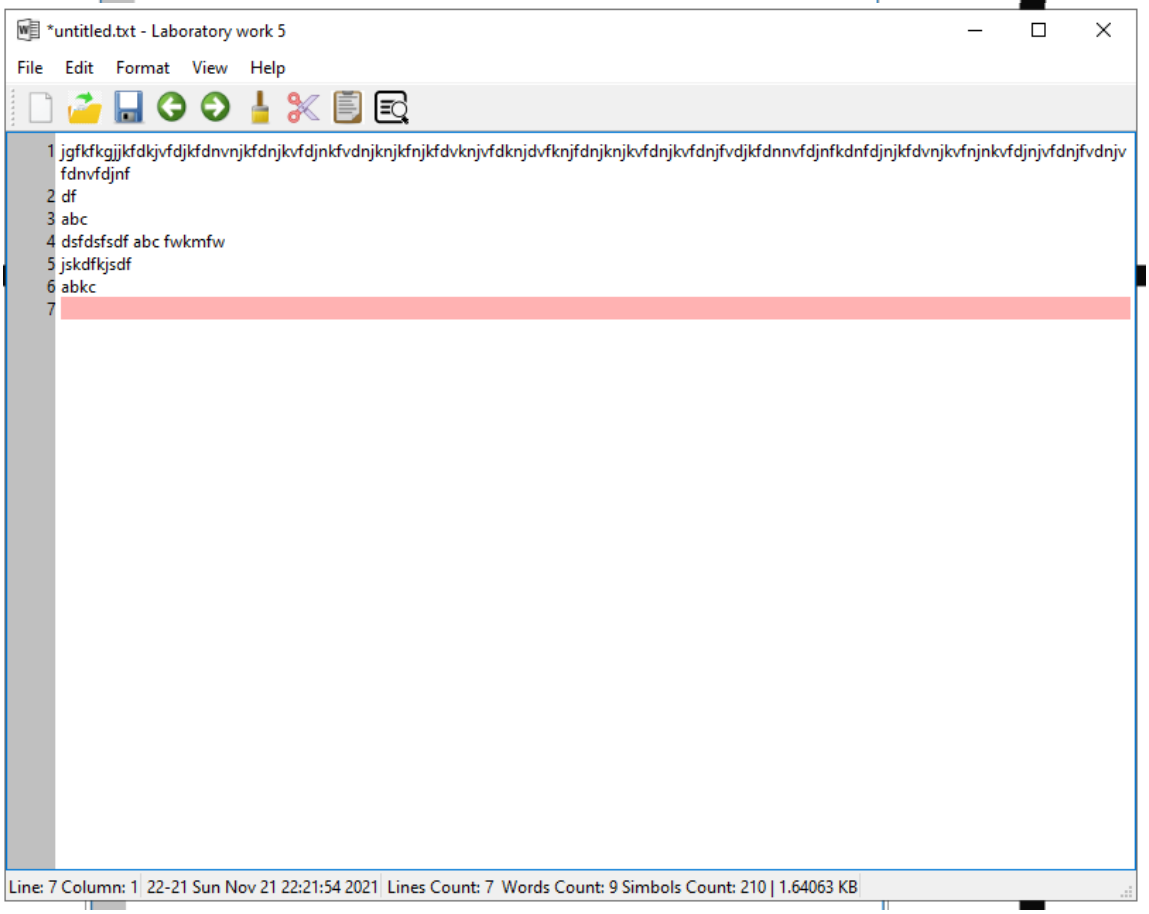

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkc
Undo
Redo
Select word
Select line
Copy
Cut
Paste
Select all
Delete
Line: 7 Column: 1 22-19 Sun Nov 21 22:19:34 2021 Lines Count: 7 Words Count: 9 Simbols Count: 202 | 1.57813 KB

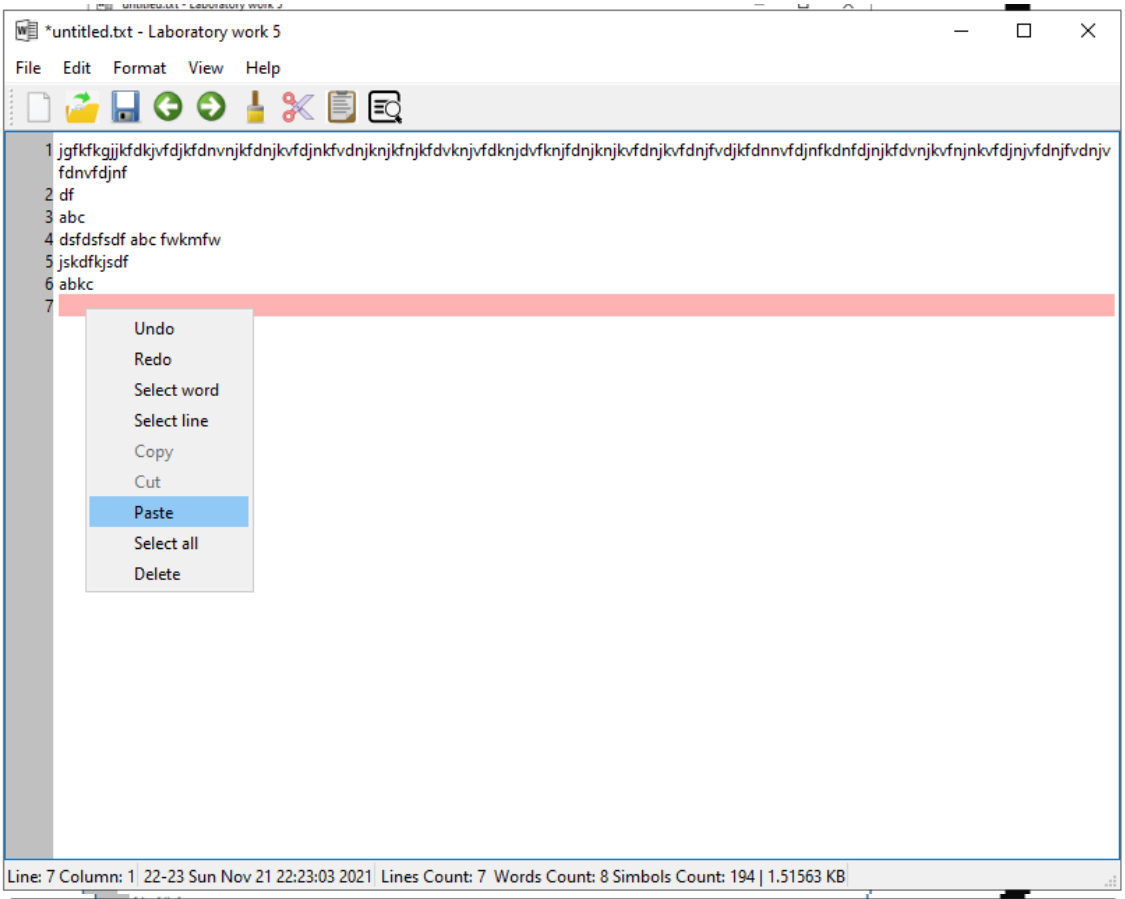

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkc
Line: 7 Column: 9 22-20 Sun Nov 21 22:20:21 2021 Lines Count: 7 Words Count: 9 Simbols Count: 202 | 1.57813 KB

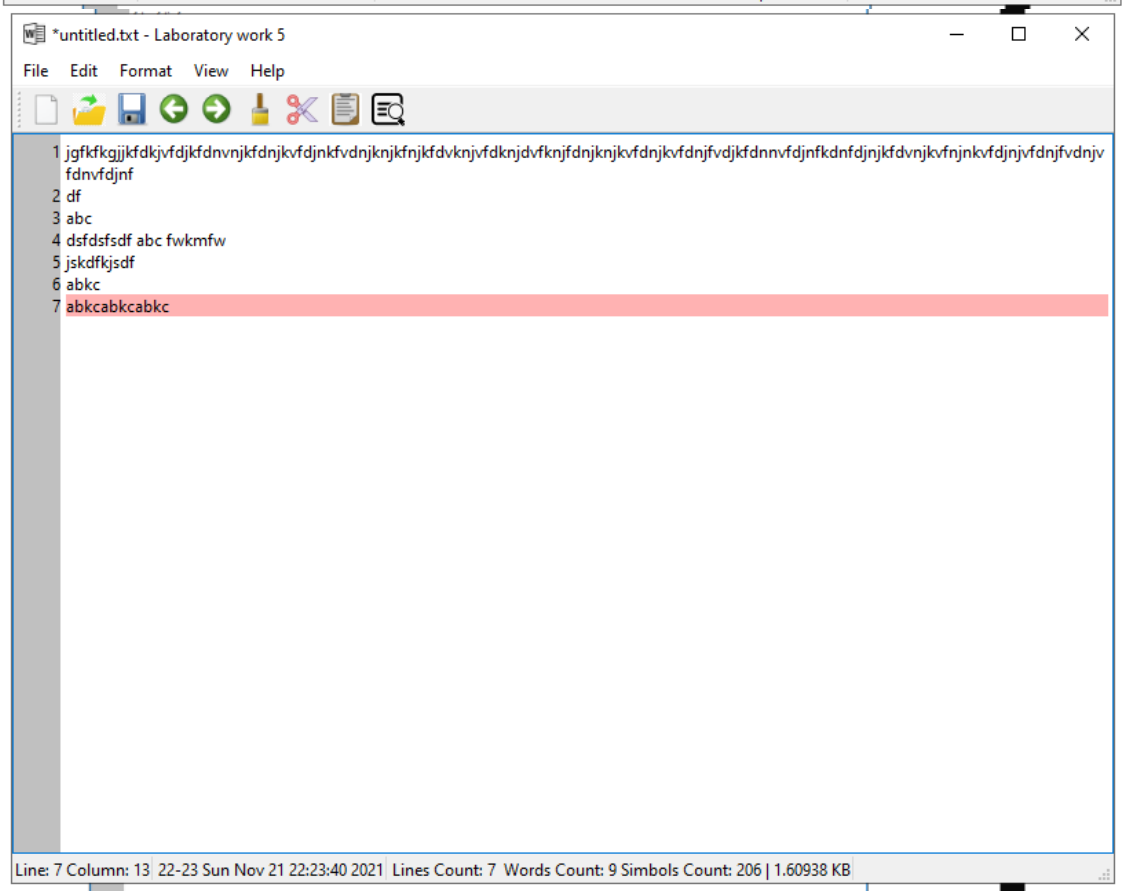

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkc
Undo
Redo
Select word
Select line
Copy
Cut
Paste
Select all
Delete
Line: 7 Column: 9 22-21 Sun Nov 21 22:21:00 2021 Lines Count: 7 Words Count: 9 Simbols Count: 202 | 1.57813 KB

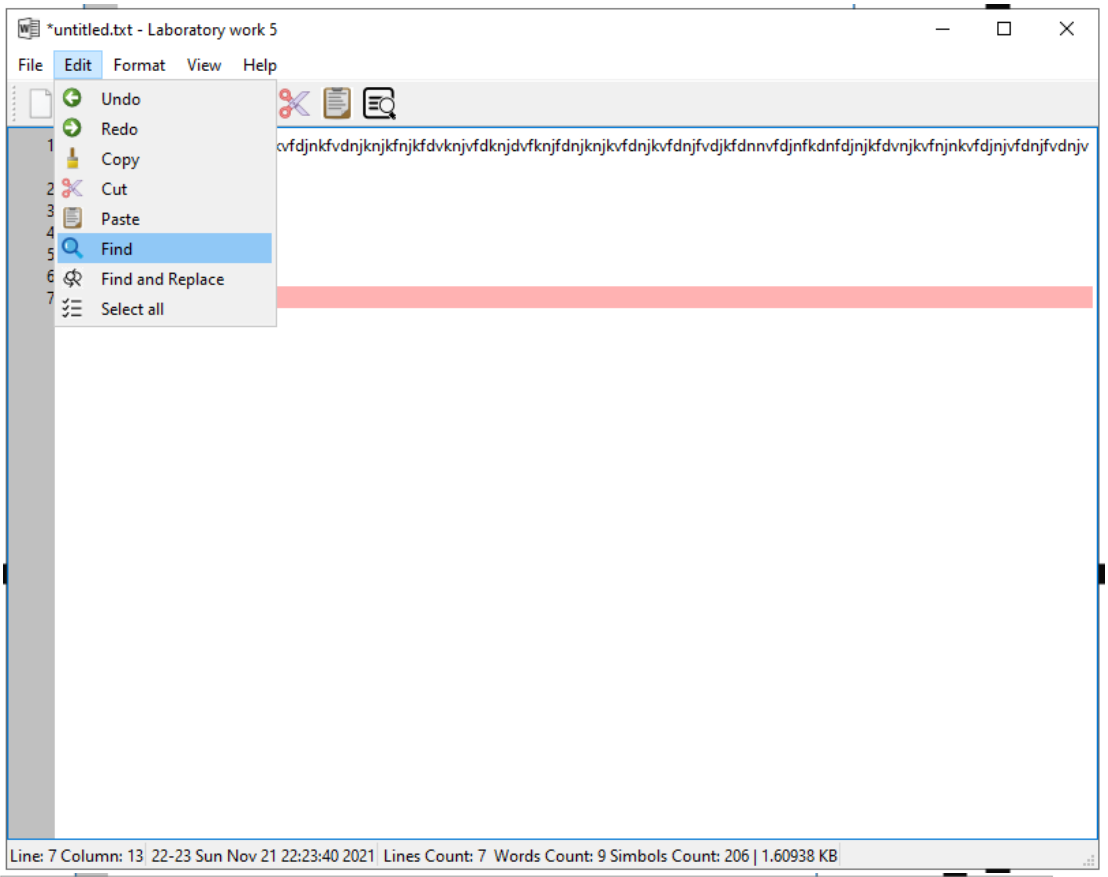

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkcabkc
Line: 7 Column: 17 22-21 Sun Nov 21 22:21:33 2021 Lines Count: 7 Words Count: 9 Simbols Count: 210 | 1.64063 KB

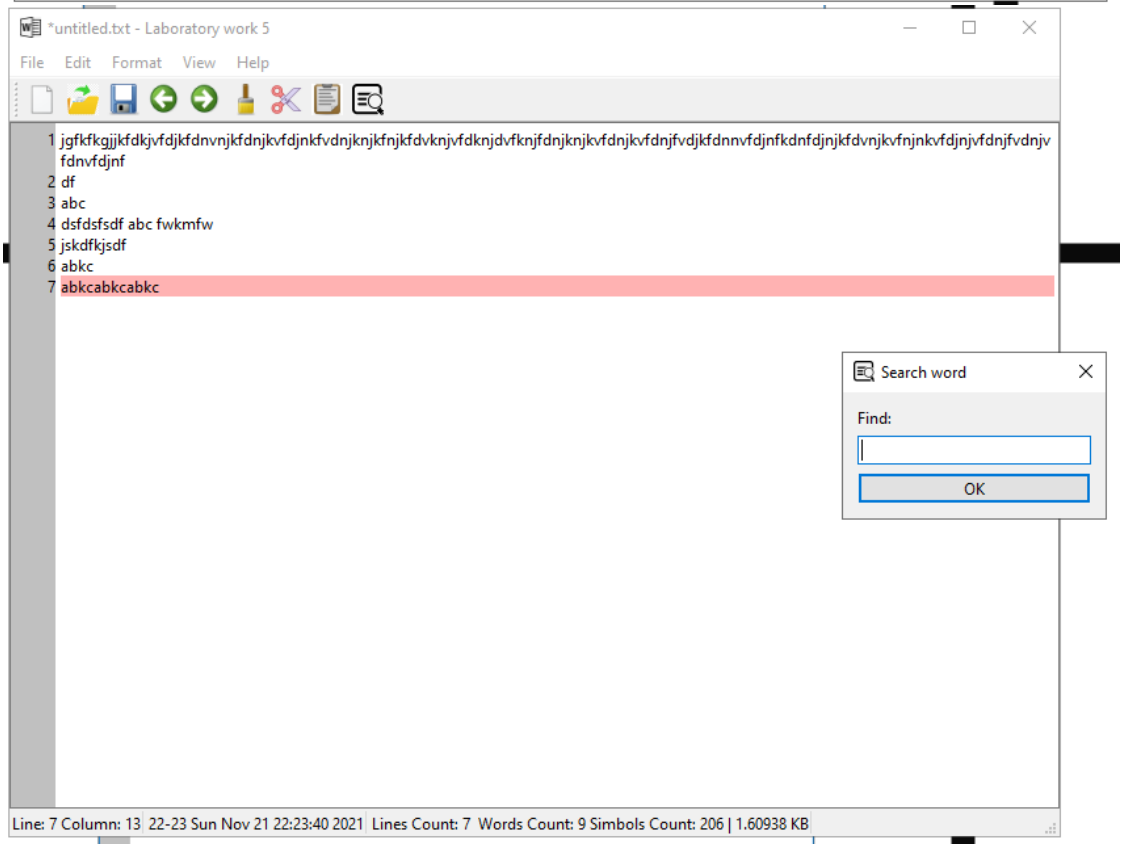

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcab
Undo
Redo
Select word
Select line
Copy
Cut
Paste
Select all
Delete
Line: 7 Column: 1 22-21 Sun Nov 21 22:21:54 2021 Lines Count: 7 Words Count: 9 Simbols Count: 210 | 1.64063 KB

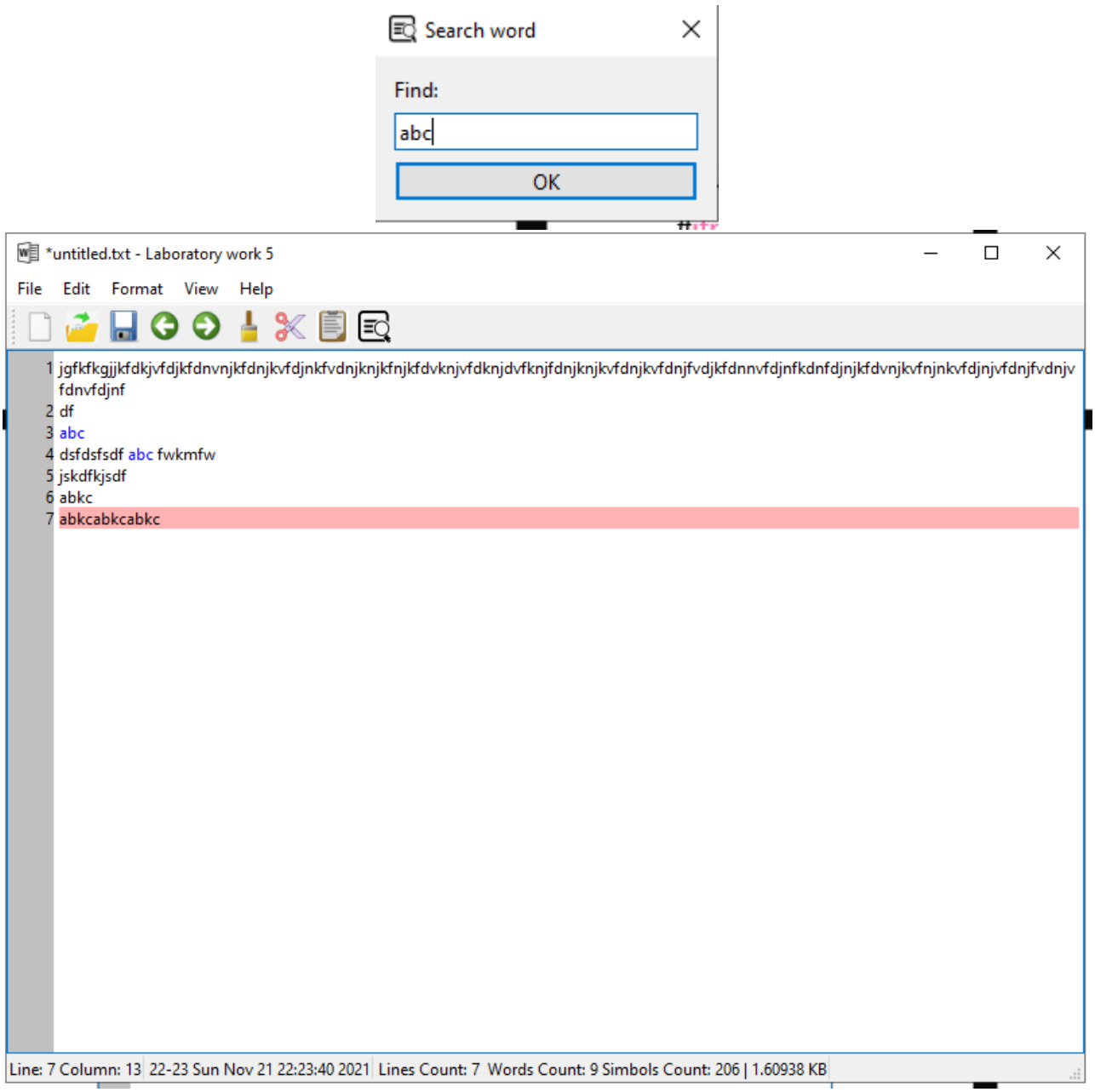

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7
Line: 7 Column: 1 22-21 Sun Nov 21 22:21:54 2021 Lines Count: 7 Words Count: 9 Simbols Count: 210 | 1.64063 KB

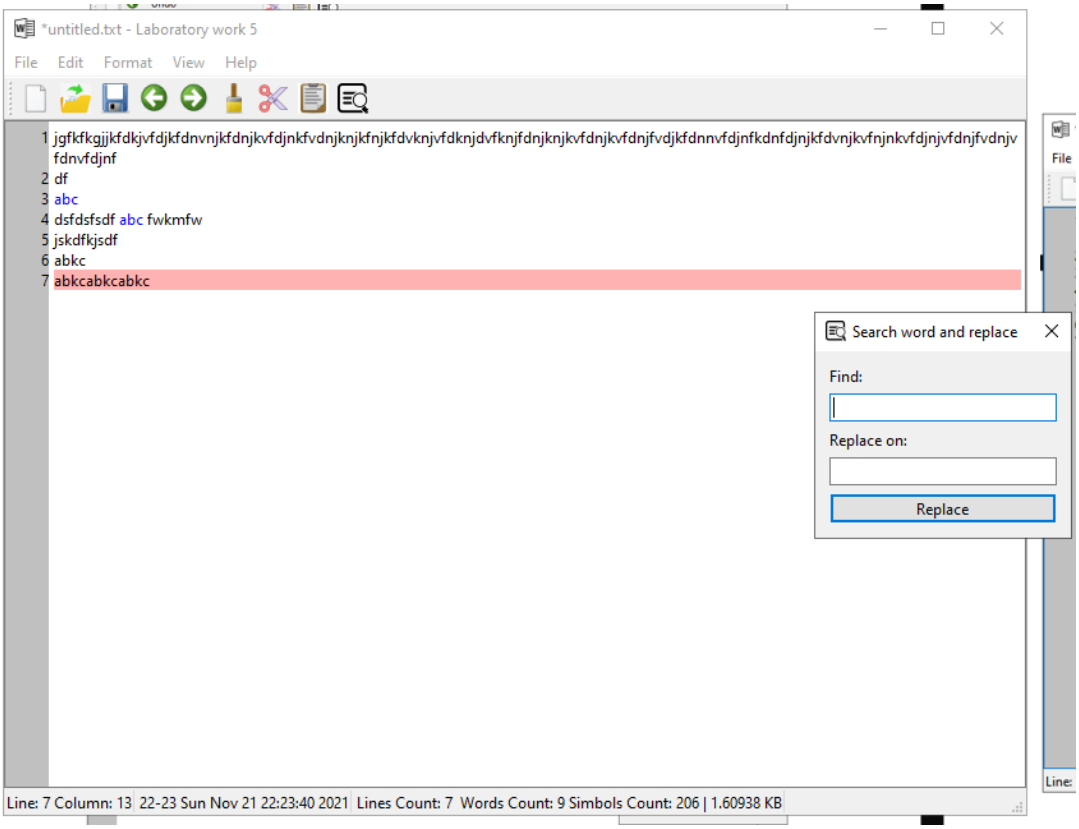

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7
Undo
Redo
Select word
Select line
Copy
Cut
Paste
Select all
Delete
Line: 7 Column: 1 22-23 Sun Nov 21 22:23:03 2021 Lines Count: 7 Words Count: 8 Simbols Count: 194 | 1.51563 KB

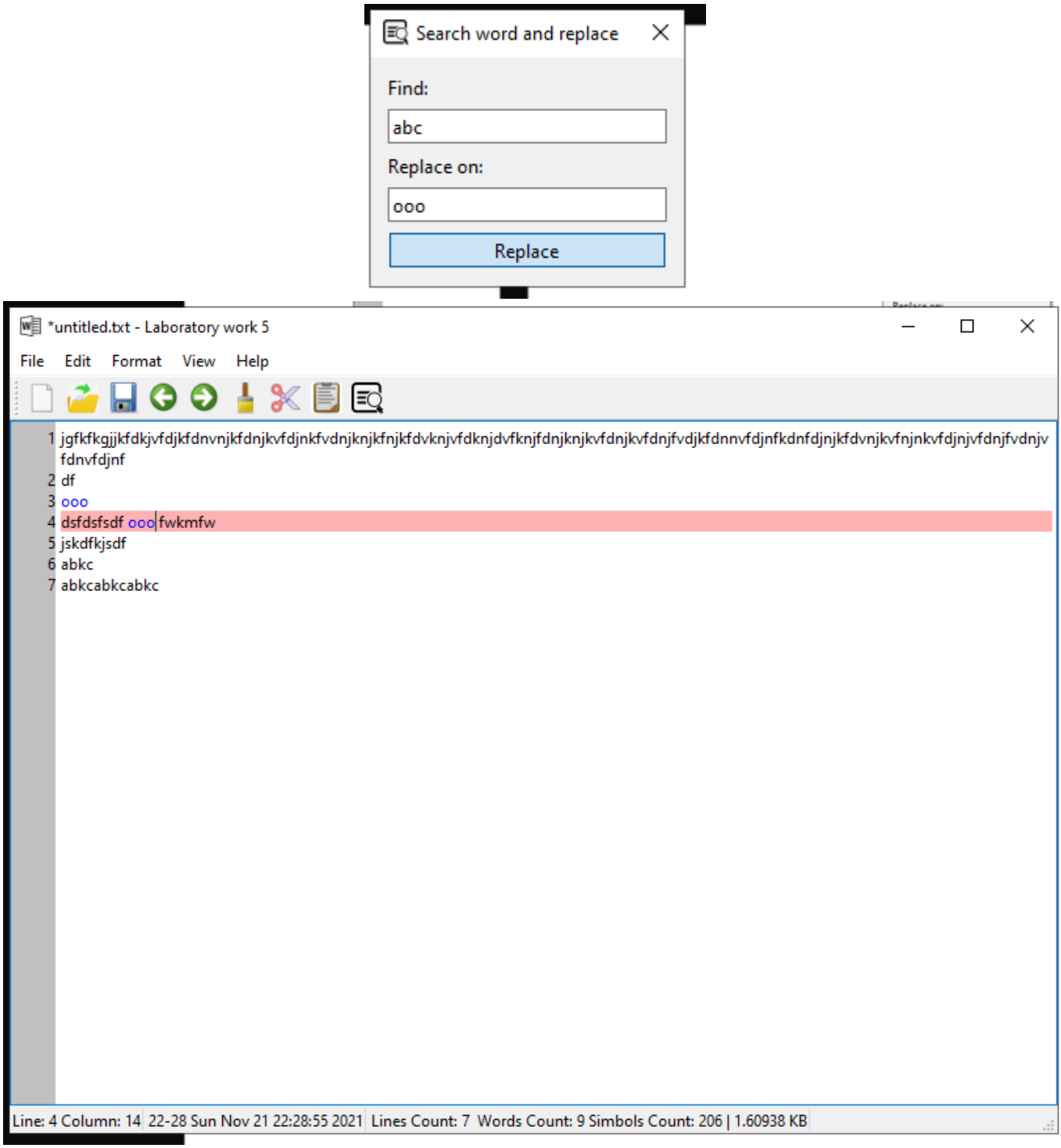

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 7 Column: 13 22-23 Sun Nov 21 22:23:40 2021 Lines Count: 7 Words Count: 9 Simbols Count: 206 | 1.60938 KB

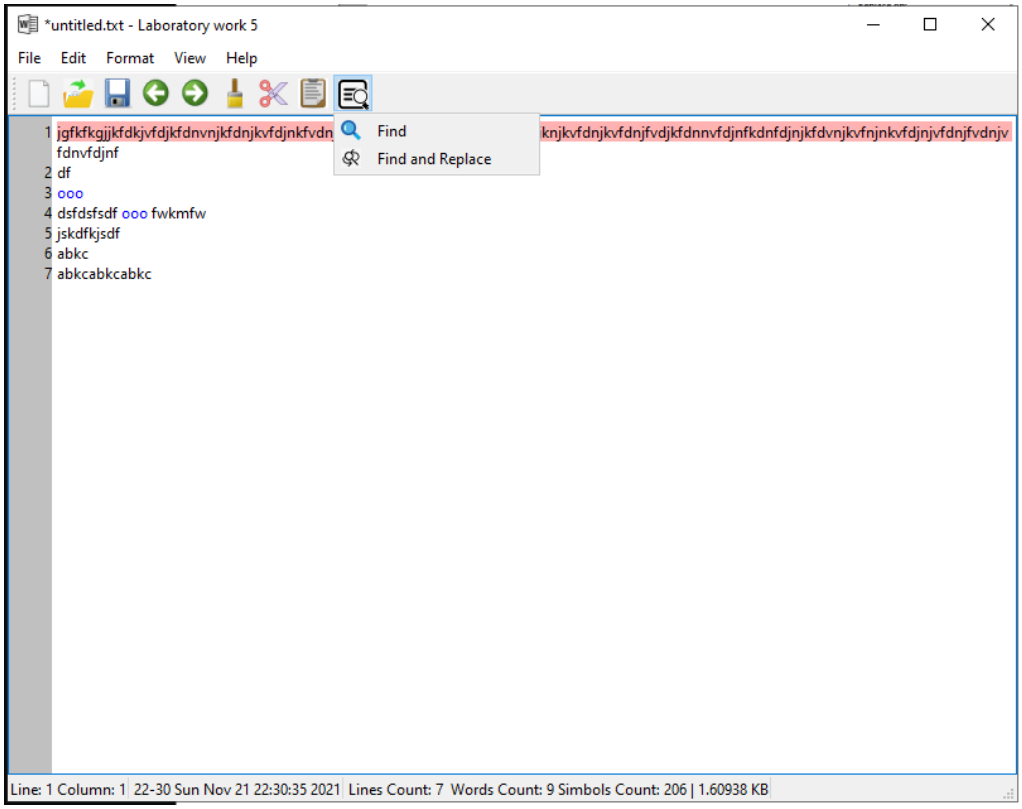

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
Undo
Redo
Copy
Cut
Paste
Find
Find and Replace
Select all
cvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
Line: 7 Column: 13
22-23 Sun Nov 21 22:23:40 2021
Lines Count: 7
Words Count: 9
Simbols Count: 206 | 1.60938 KB

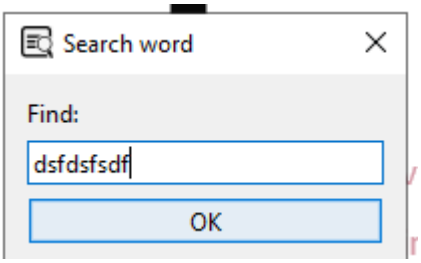

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Search word
Find:
OK
Line: 7 Column: 13
22-23 Sun Nov 21 22:23:40 2021
Lines Count: 7
Words Count: 9
Simbols Count: 206 | 1.60938 KB

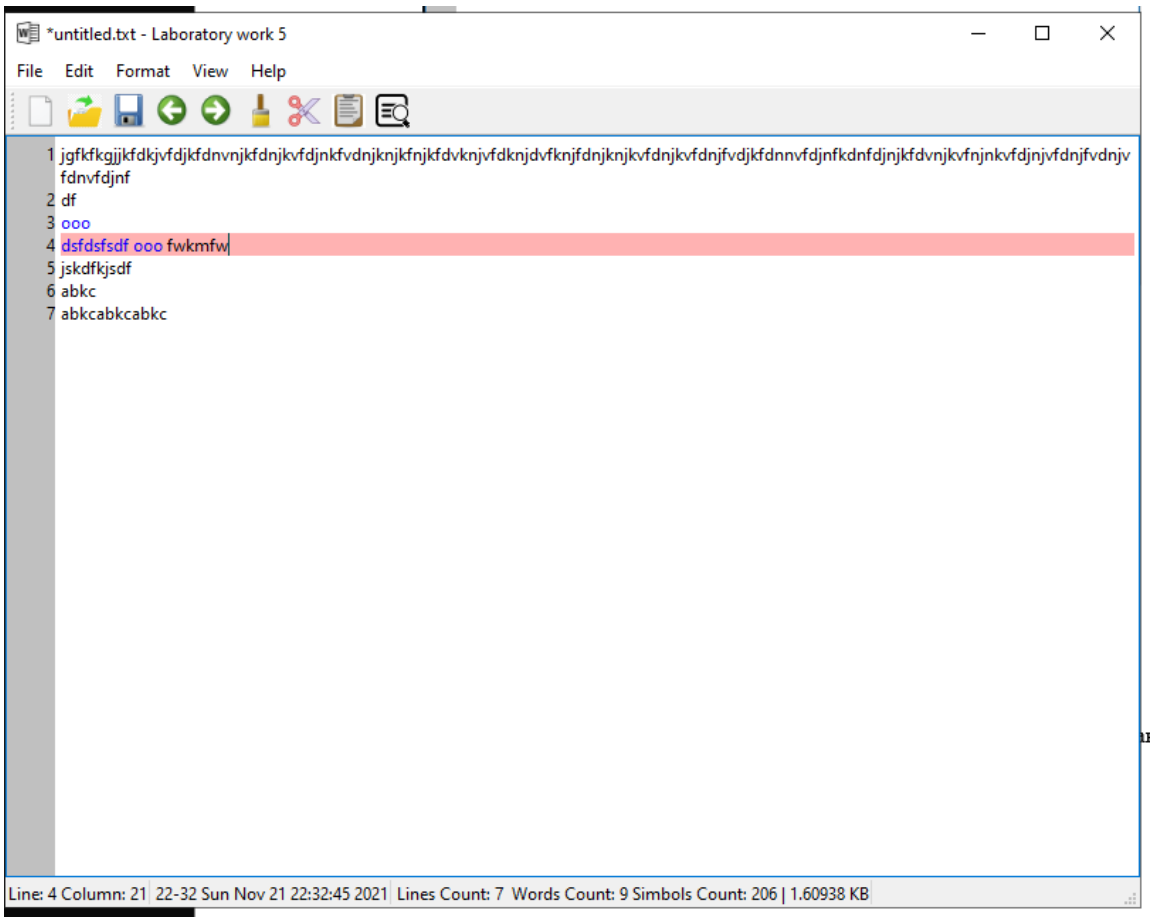

Search word
Find:
abc
OK
*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjvfdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 7 Column: 13 22-23 Sun Nov 21 22:23:40 2021 Lines Count: 7 Words Count: 9 Simbols Count: 206 | 1.60938 KB

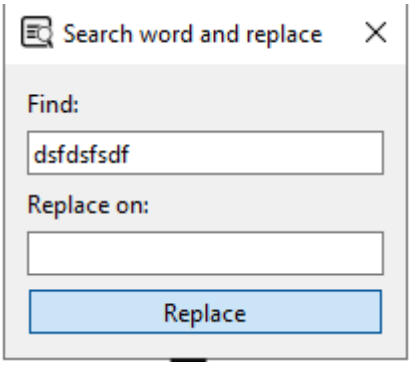

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 abc
4 dsfdsfsdf abc fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Search word and replace
Find:
Replace on:
Replace
Line: 7 Column: 13 22-23 Sun Nov 21 22:23:40 2021 Lines Count: 7 Words Count: 9 Simbols Count: 206 | 1.60938 KB

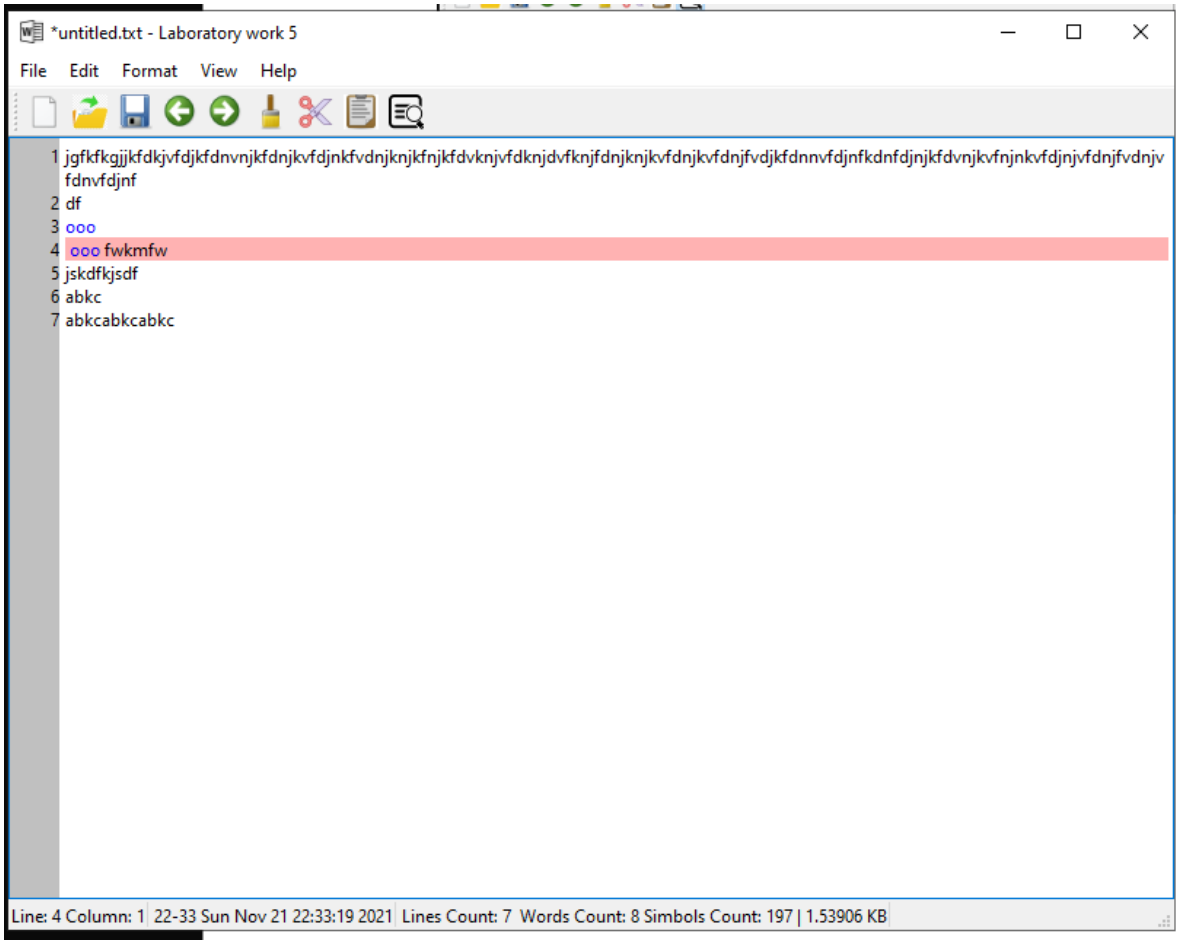

Search word and replace
Find:
abc
Replace on:
ooo
Replace
*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjvfdnvfdjnf
2 df
3 ooo
4 dsfdsfsdf ooo fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 14 22-28 Sun Nov 21 22:28:55 2021 Lines Count: 7 Words Count: 9 Simbols Count: 206 | 1.60938 KB

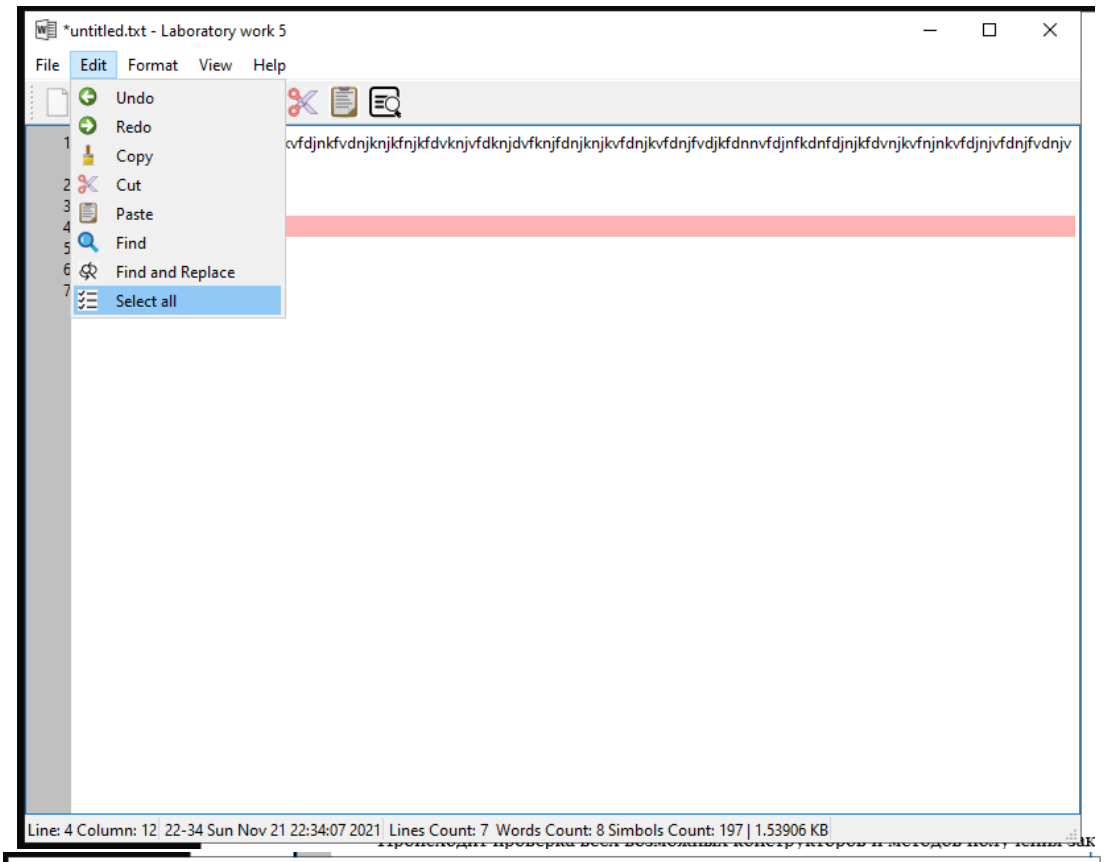

*untitled.txt - Laboratory work 5
File Edit Format View Help
Find
Find and Replace
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdn knjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 ooo
4 dsfdsfsdf ooo fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 1 Column: 1 22-30 Sun Nov 21 22:30:35 2021 Lines Count: 7 Words Count: 9 Simbols Count: 206 | 1.60938 KB

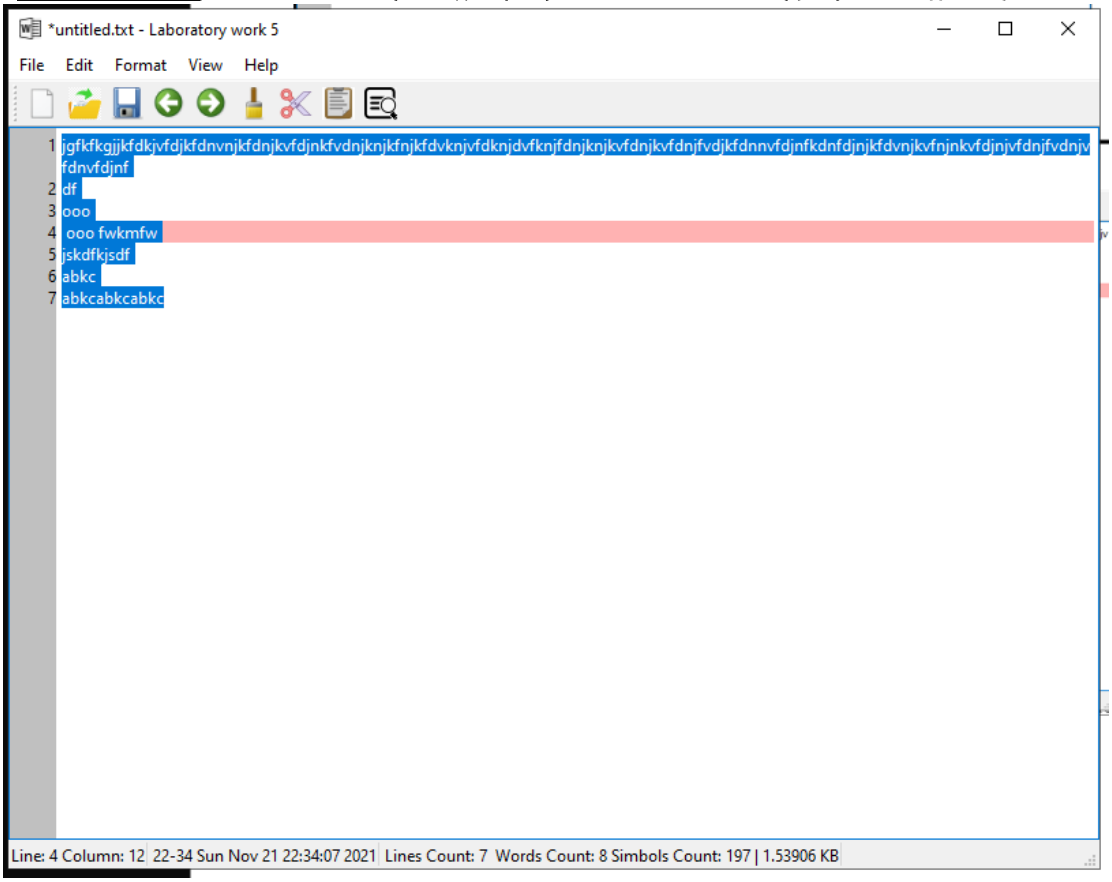

Search word
Find:
dsfdsfsdf
OK

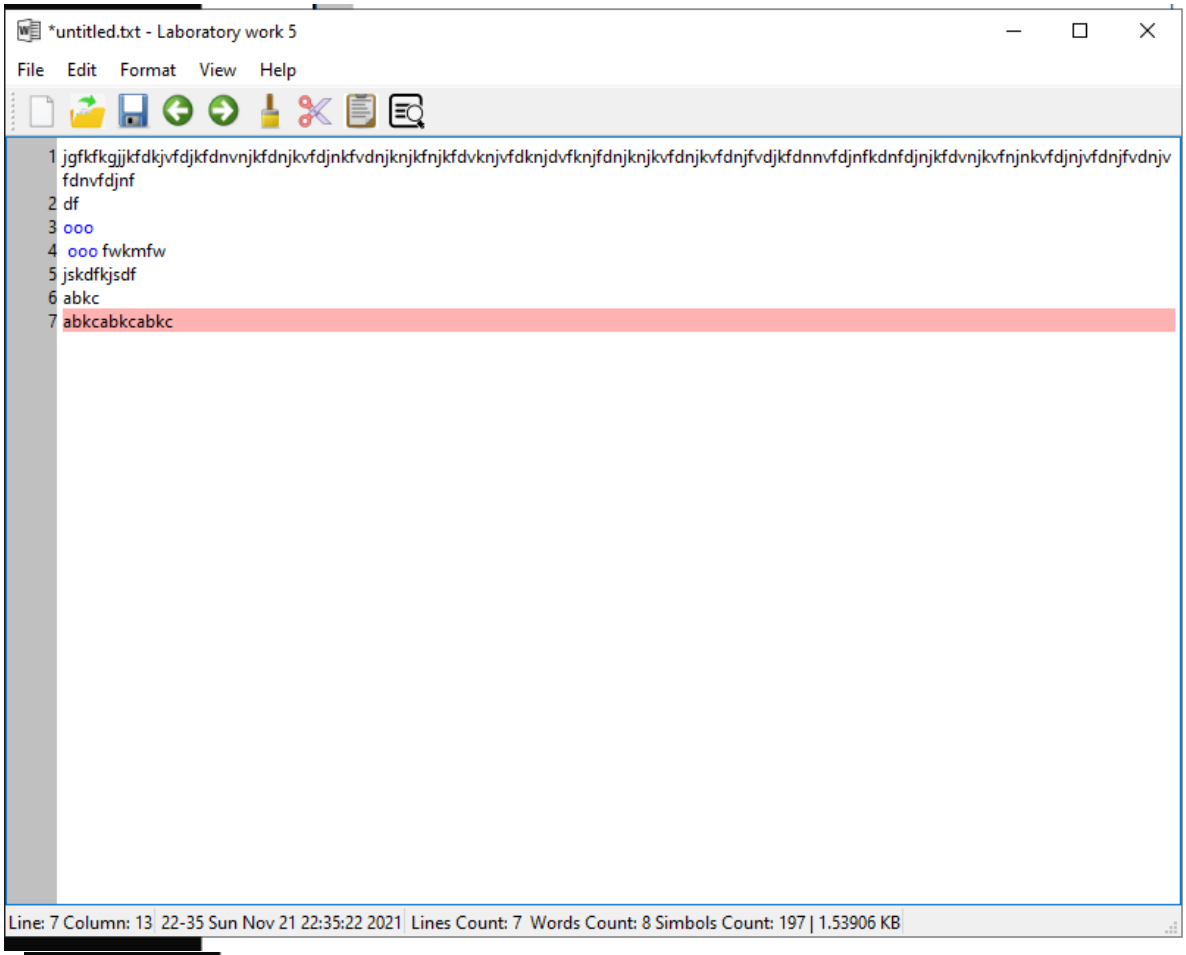

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 ooo
4 dsfdsfsdf ooo fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 21 22-32 Sun Nov 21 22:32:45 2021 Lines Count: 7 Words Count: 9 Simbols Count: 206 | 1.60938 KB

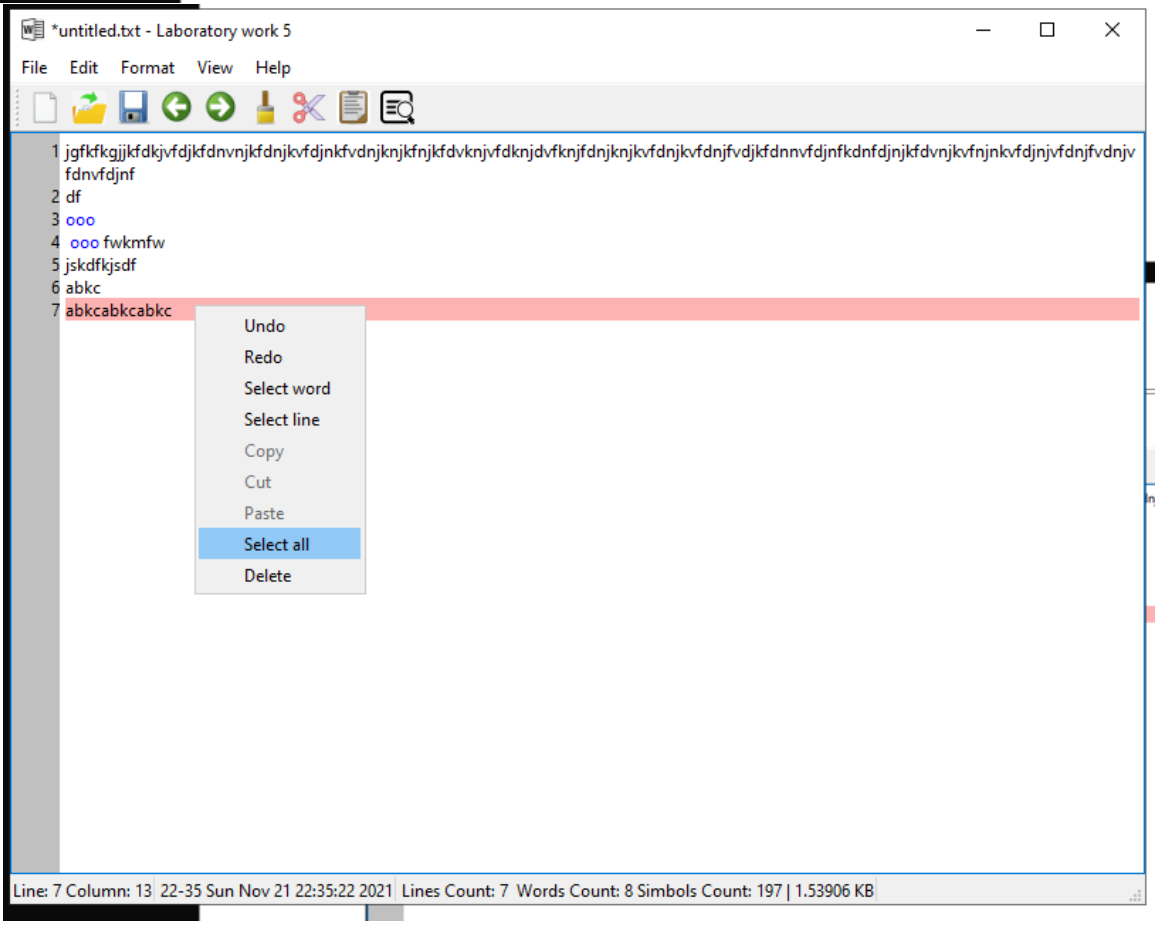

Search word and replace
Find:
dsfdsfsdf
Replace on:
Replace

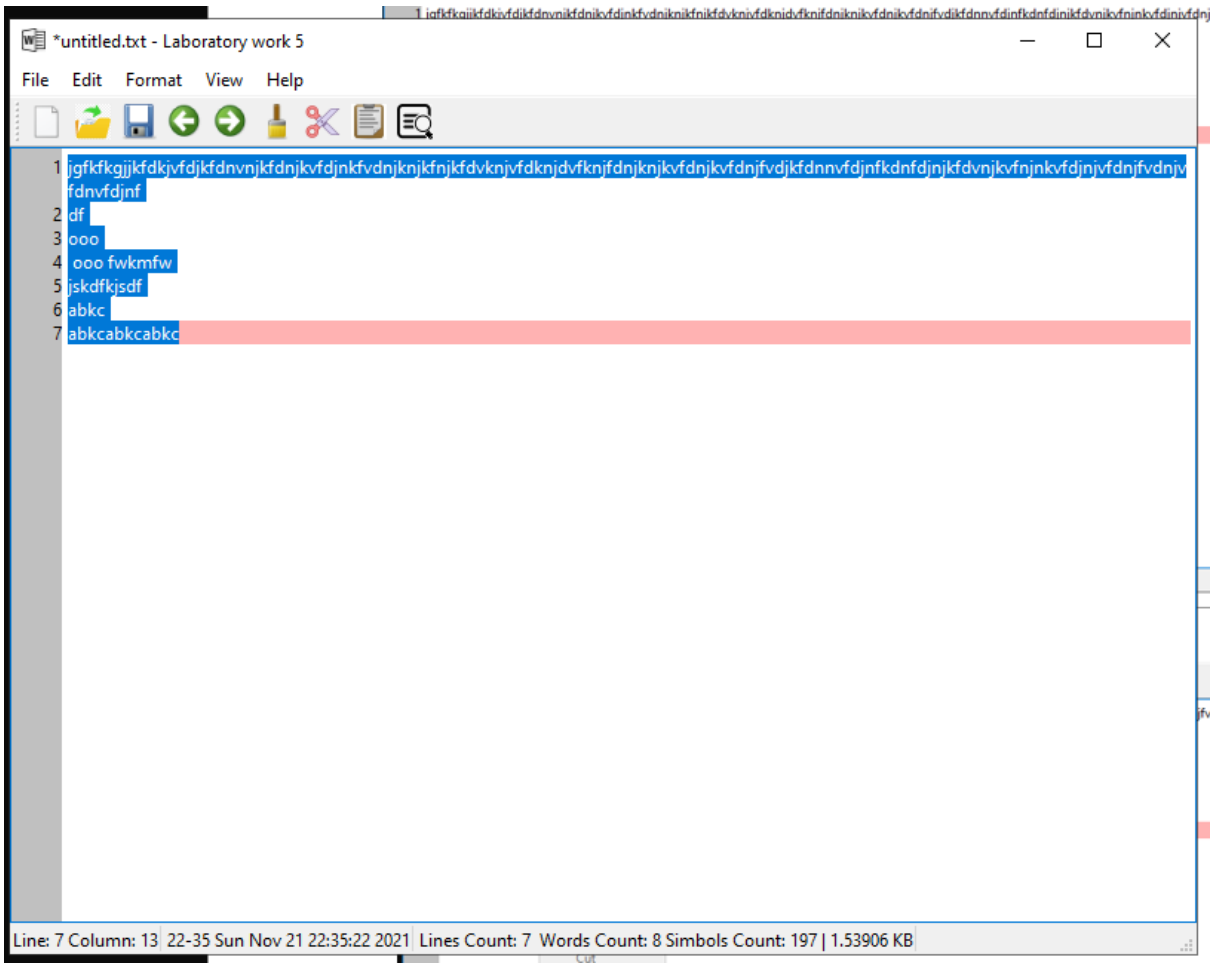

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjvfdnvfdjnf
2 df
3 ooo
4 ooo fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 1 22-33 Sun Nov 21 22:33:19 2021 Lines Count: 7 Words Count: 8 Simbols Count: 197 | 1.53906 KB

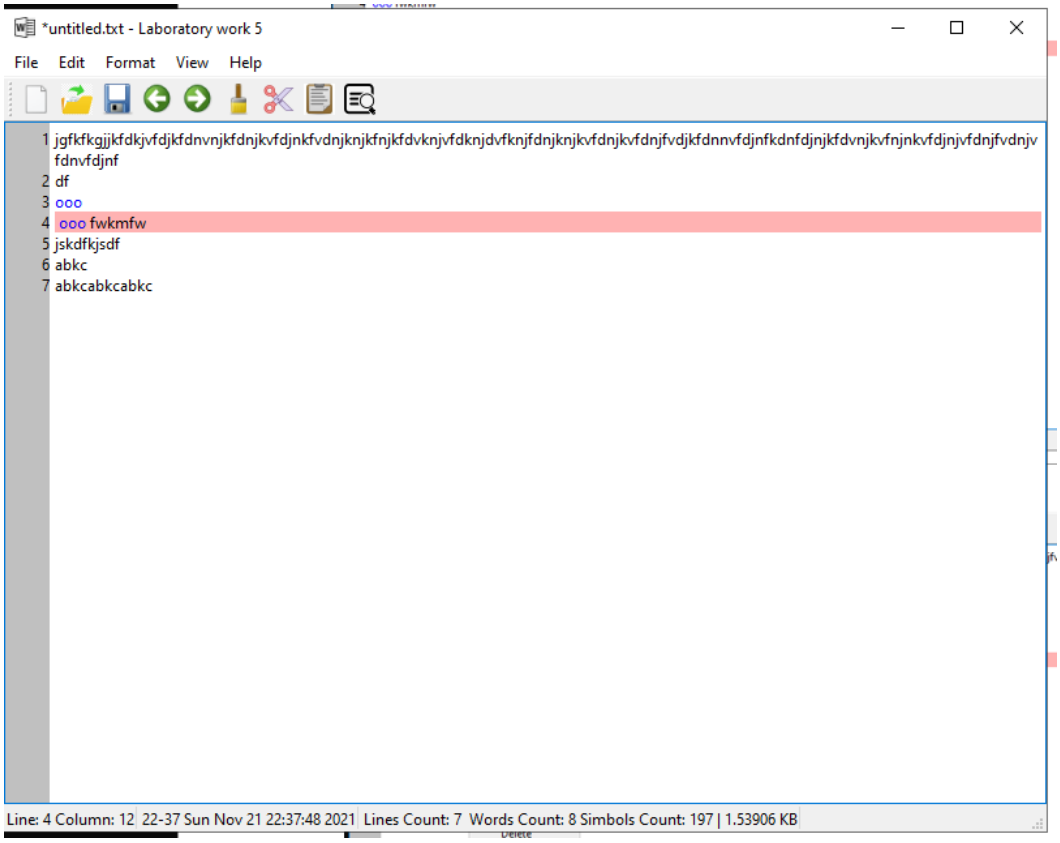

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
Undo
Redo
Copy
Cut
Paste
Find
Find and Replace
Select all
Line: 4 Column: 12
22-34 Sun Nov 21 22:34:07 2021
Lines Count: 7
Words Count: 8
Simbols Count: 197 | 1.53906 KB

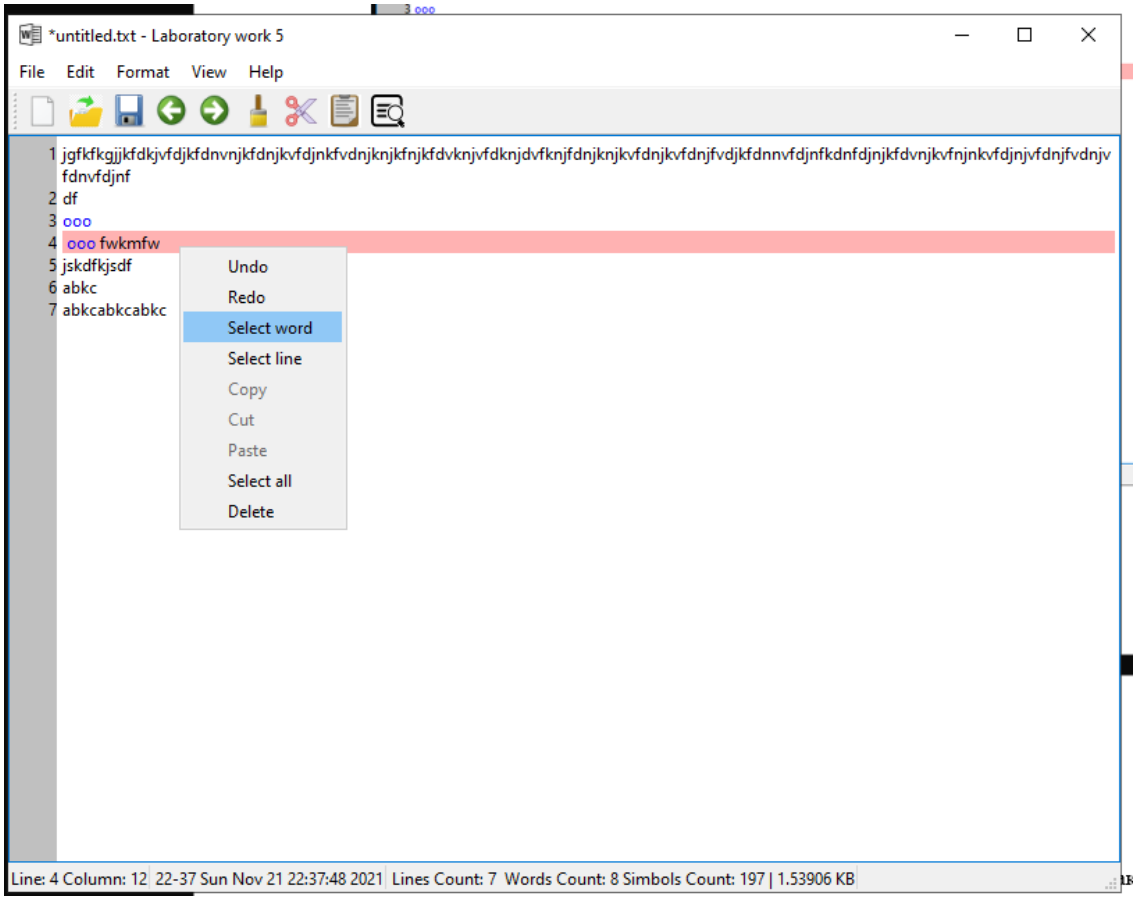

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
df
ooo
ooo fwkmfw
jskdfkjsdf
abkc
abkcabkcabkc
Line: 4 Column: 12
22-34 Sun Nov 21 22:34:07 2021
Lines Count: 7
Words Count: 8
Simbols Count: 197 | 1.53906 KB

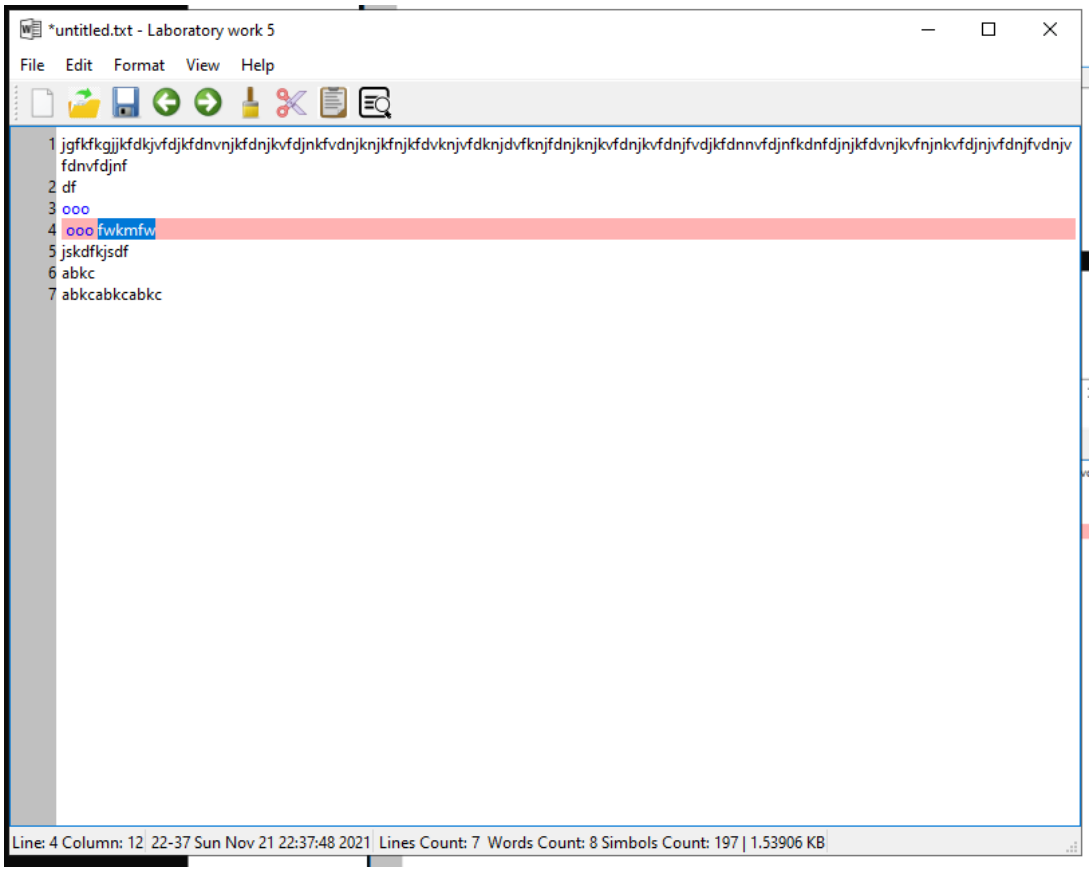

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 ooo
4 ooo fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 7 Column: 13 22-35 Sun Nov 21 22:35:22 2021 Lines Count: 7 Words Count: 8 Simbols Count: 197 | 1.53906 KB

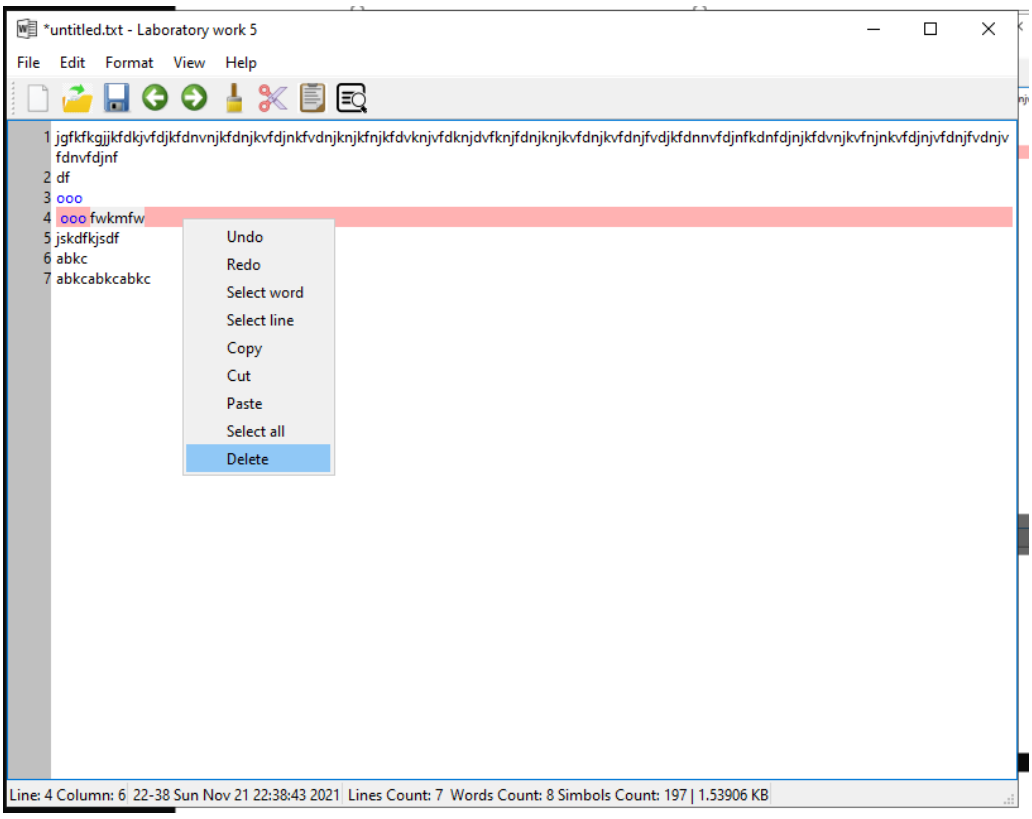

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 ooo
4 ooo fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Undo
Redo
Select word
Select line
Copy
Cut
Paste
Select all
Delete
Line: 7 Column: 13 22-35 Sun Nov 21 22:35:22 2021 Lines Count: 7 Words Count: 8 Simbols Count: 197 | 1.53906 KB

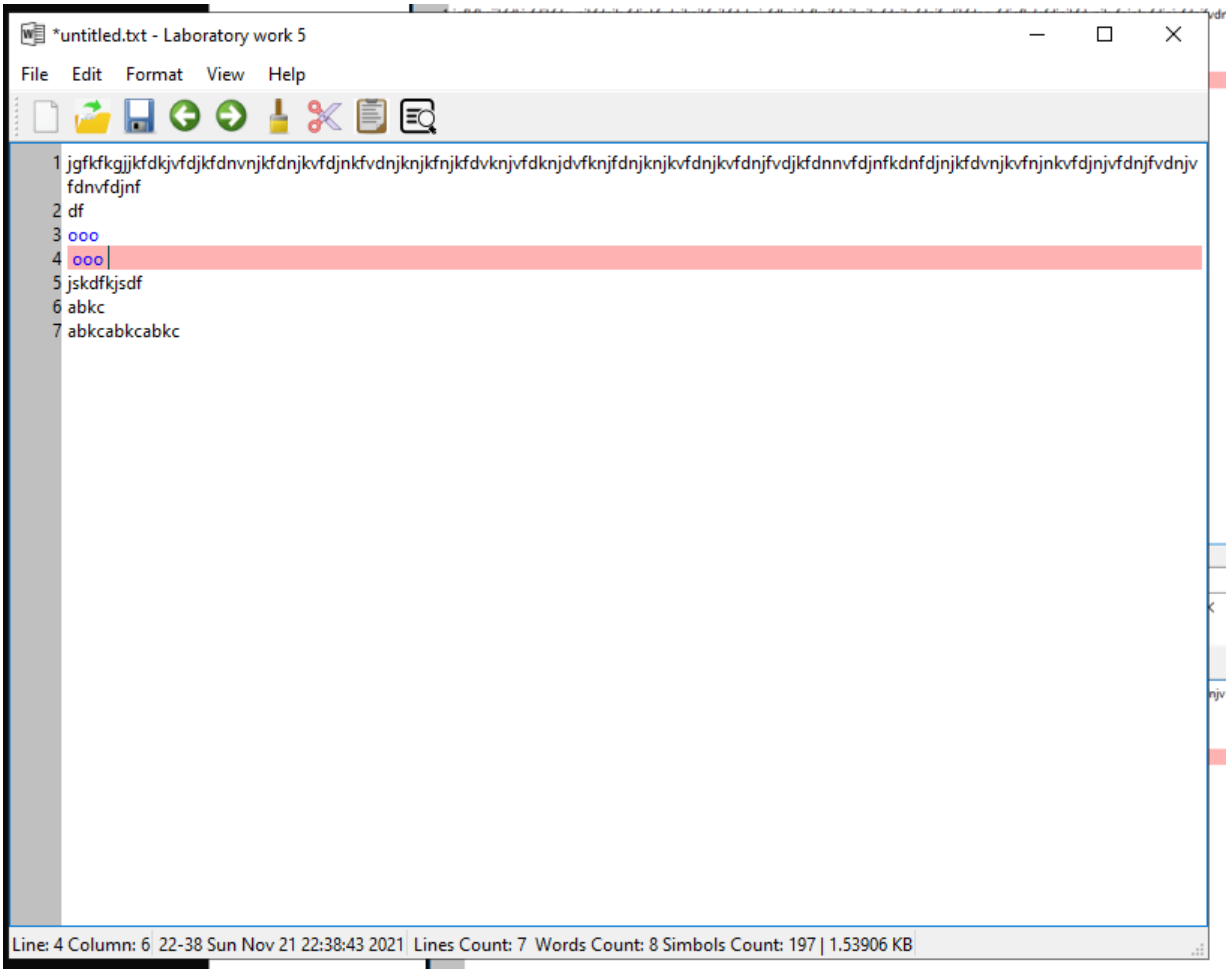

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 ooo
4 ooo fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 7 Column: 13 22-35 Sun Nov 21 22:35:22 2021 Lines Count: 7 Words Count: 8 Simbols Count: 197 | 1.53906 KB

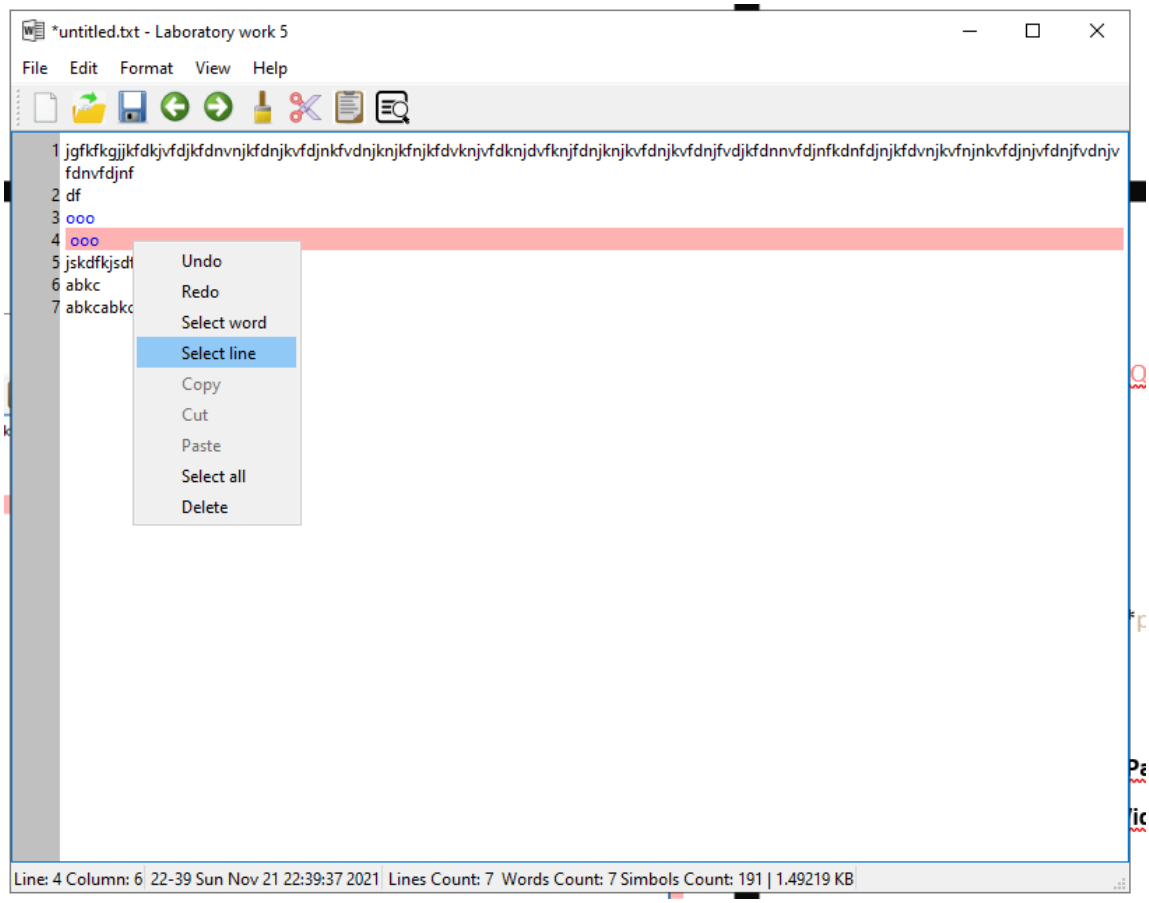

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 ooo
4 ooo fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 12 22-37 Sun Nov 21 22:37:48 2021 Lines Count: 7 Words Count: 8 Simbols Count: 197 | 1.53906 KB

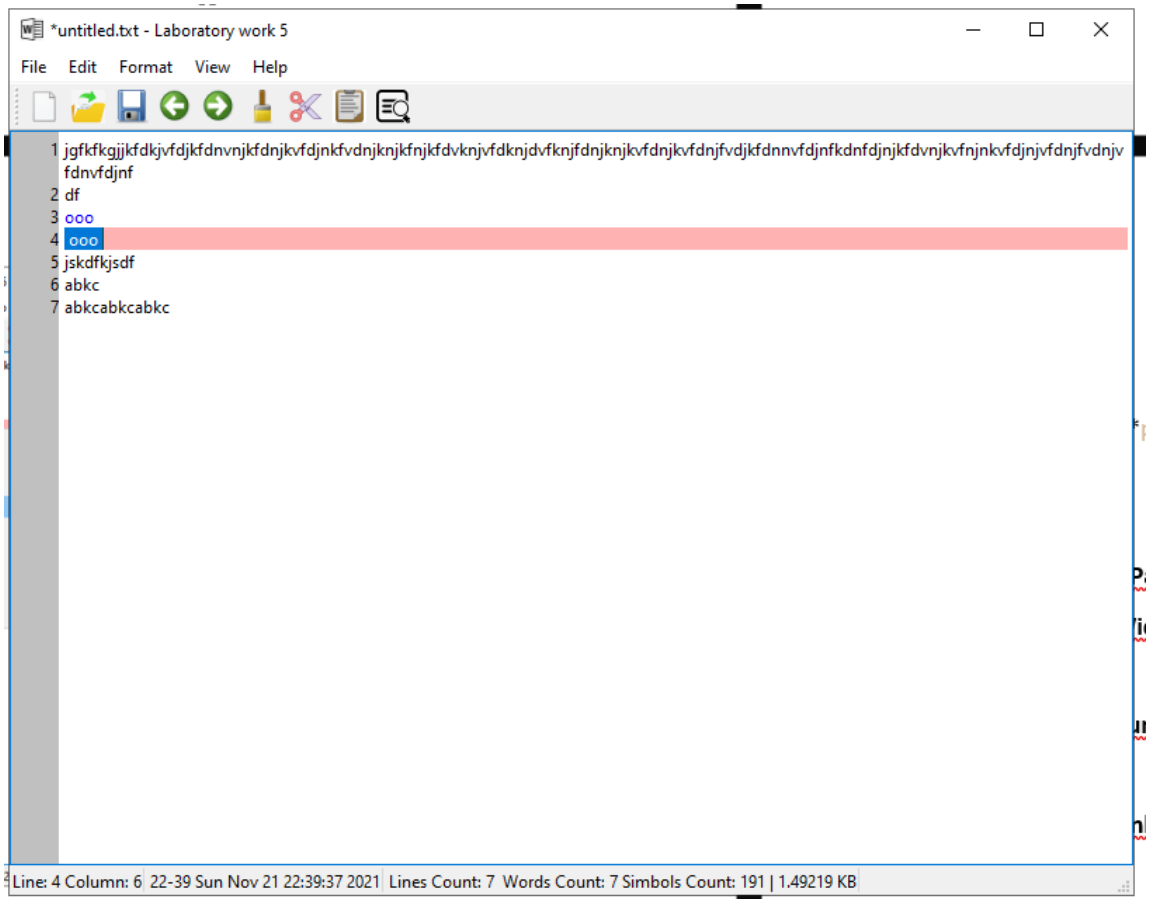

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 ooo
4 ooo fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Undo
Redo
Select word
Select line
Copy
Cut
Paste
Select all
Delete
Line: 4 Column: 12 22-37 Sun Nov 21 22:37:48 2021 Lines Count: 7 Words Count: 8 Simbols Count: 197 | 1.53906 KB

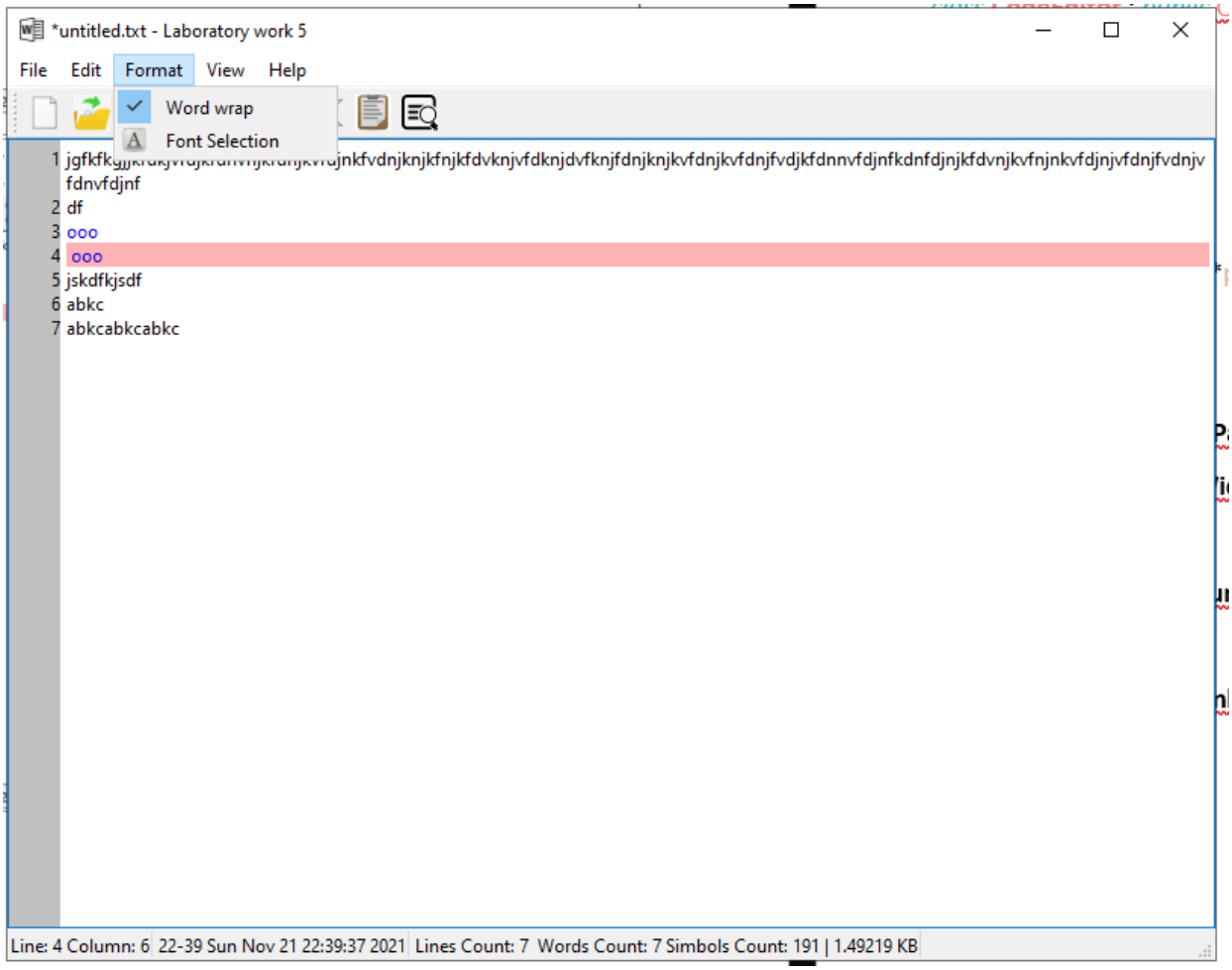

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 ooo
4 ooo fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 12 22-37 Sun Nov 21 22:37:48 2021 Lines Count: 7 Words Count: 8 Simbols Count: 197 | 1.53906 KB

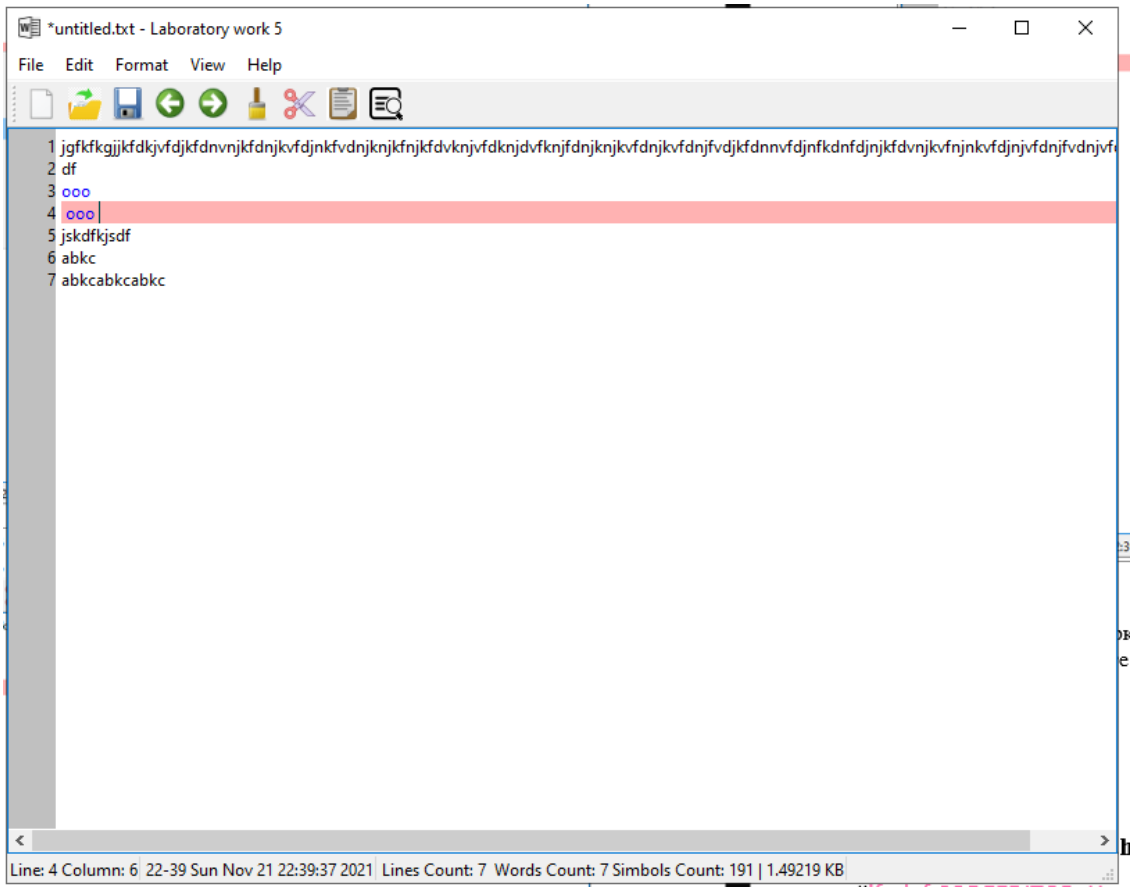

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 ooo
4 ooo fwkmfw
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Undo
Redo
Select word
Select line
Copy
Cut
Paste
Select all
Delete
Line: 4 Column: 6 22-38 Sun Nov 21 22:38:43 2021 Lines Count: 7 Words Count: 8 Simbols Count: 197 | 1.53906 KB

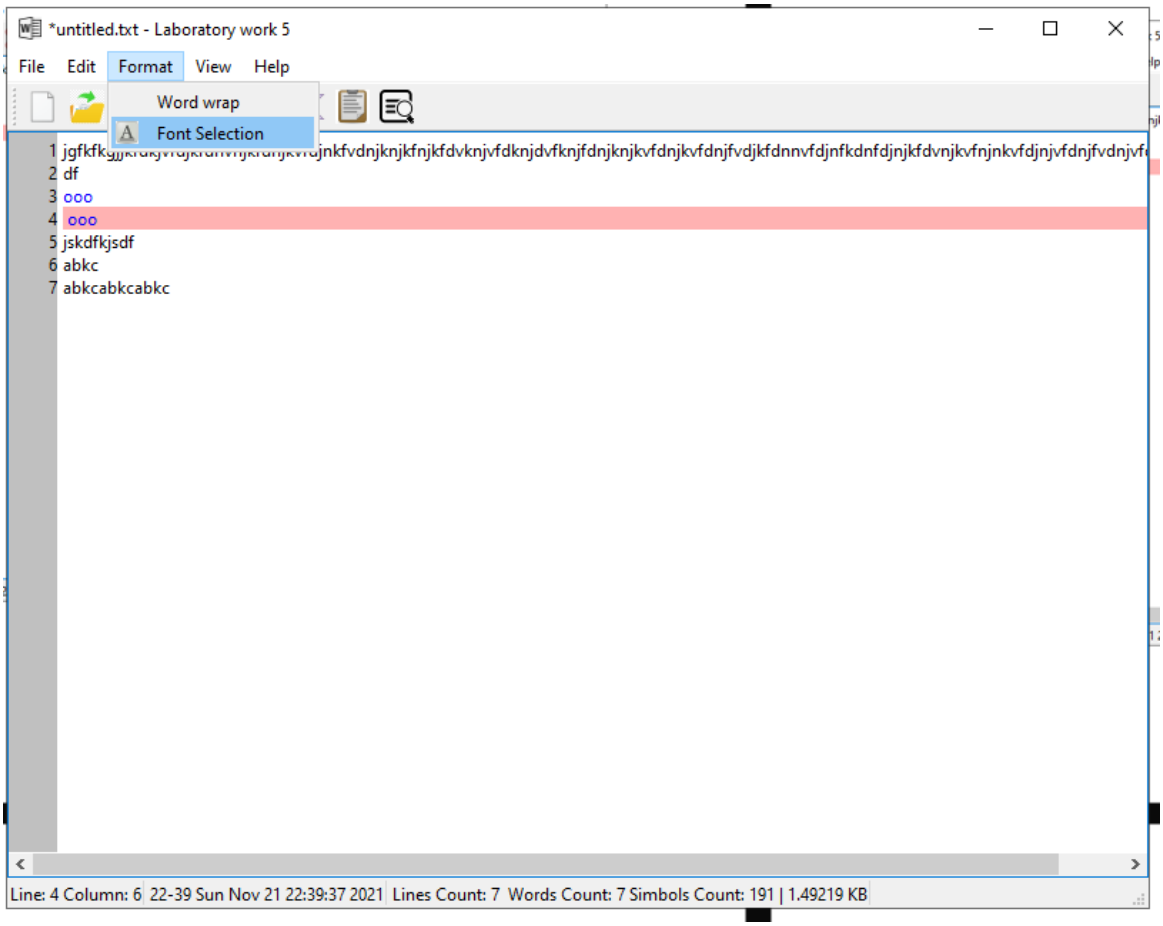

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 ooo
4 ooo
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 6 22-38 Sun Nov 21 22:38:43 2021 Lines Count: 7 Words Count: 8 Simbols Count: 197 | 1.53906 KB

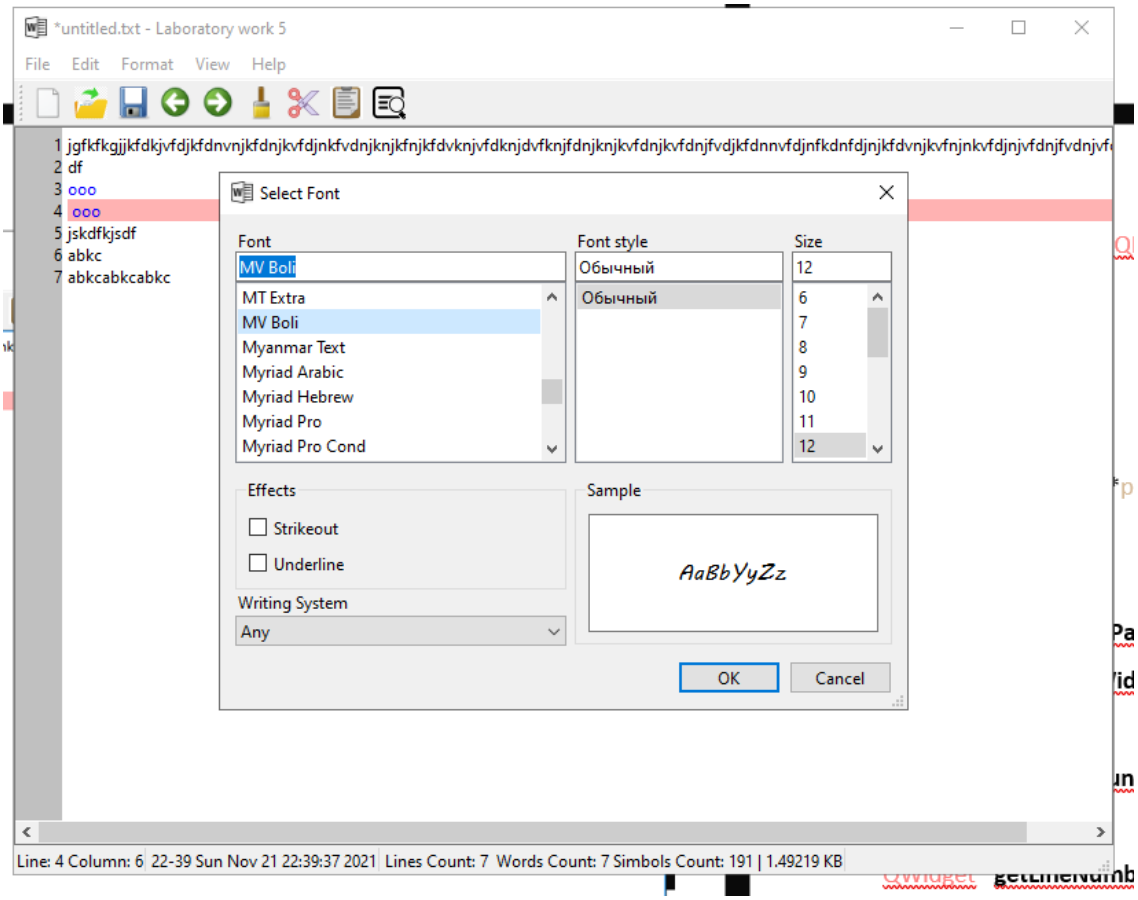

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 ooo
4 ooo
5 jskdfkjsdf
6 abkc
7 abkcabkc
Undo
Redo
Select word
Select line
Copy
Cut
Paste
Select all
Delete
Line: 4 Column: 6 22-39 Sun Nov 21 22:39:37 2021 Lines Count: 7 Words Count: 7 Simbols Count: 191 | 1.49219 KB

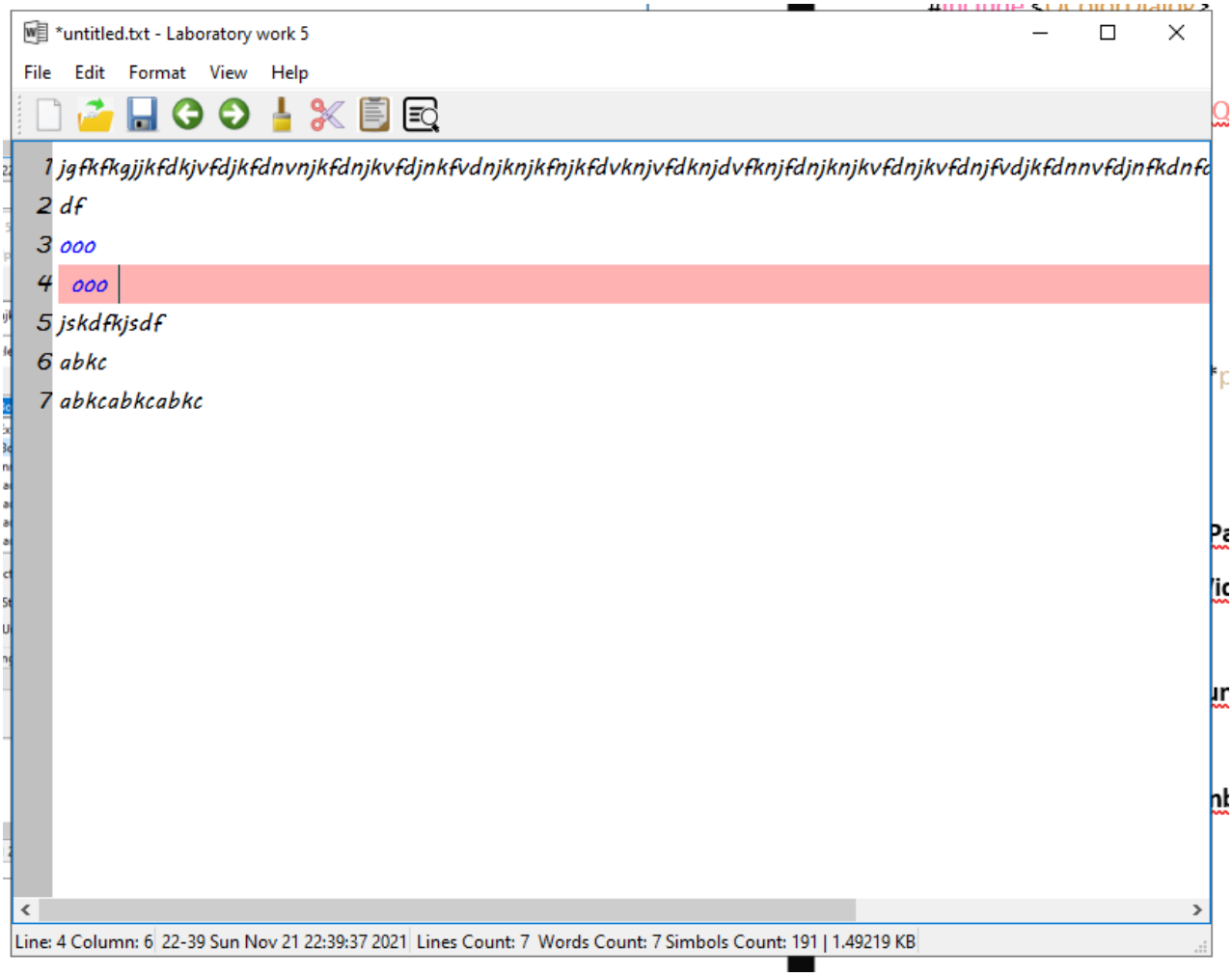

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 ooo
4 ooo
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 6 22-39 Sun Nov 21 22:39:37 2021 Lines Count: 7 Words Count: 7 Simbols Count: 191 | 1.49219 KB

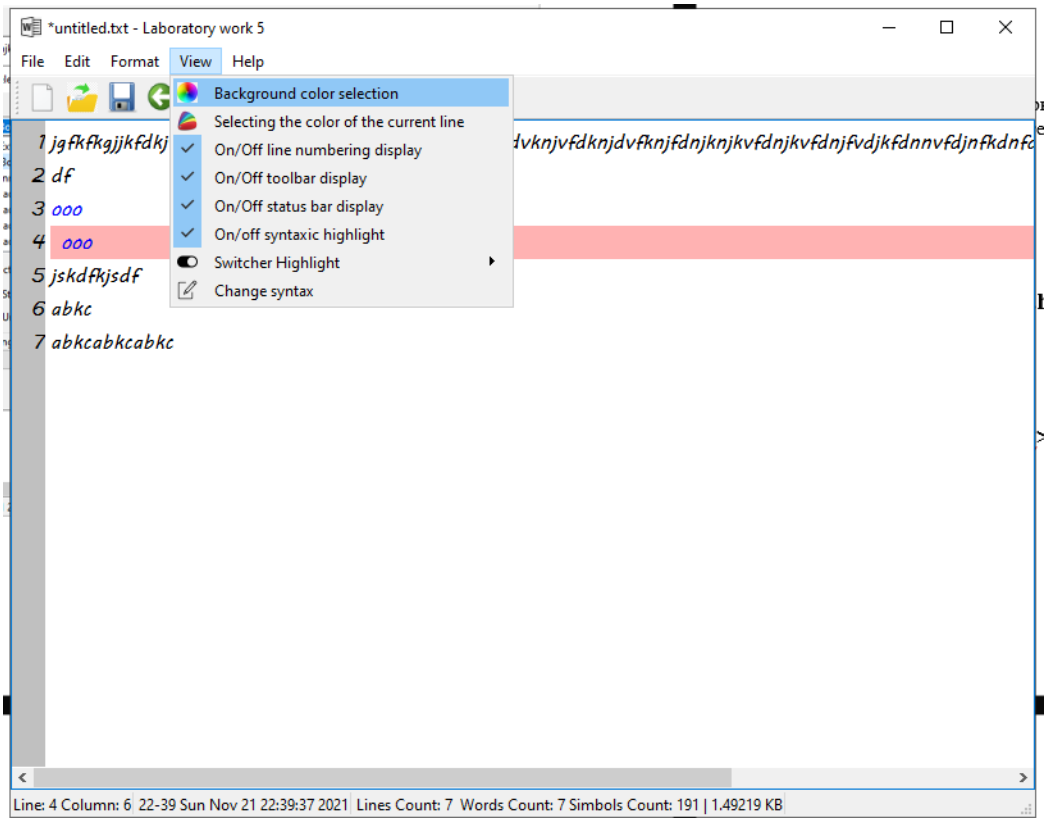

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
Word wrap
Font Selection
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
fdnvfdjnf
2 df
3 ooo
4 ooo
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 6 22-39 Sun Nov 21 22:39:37 2021 Lines Count: 7 Words Count: 7 Simbols Count: 191 | 1.49219 KB

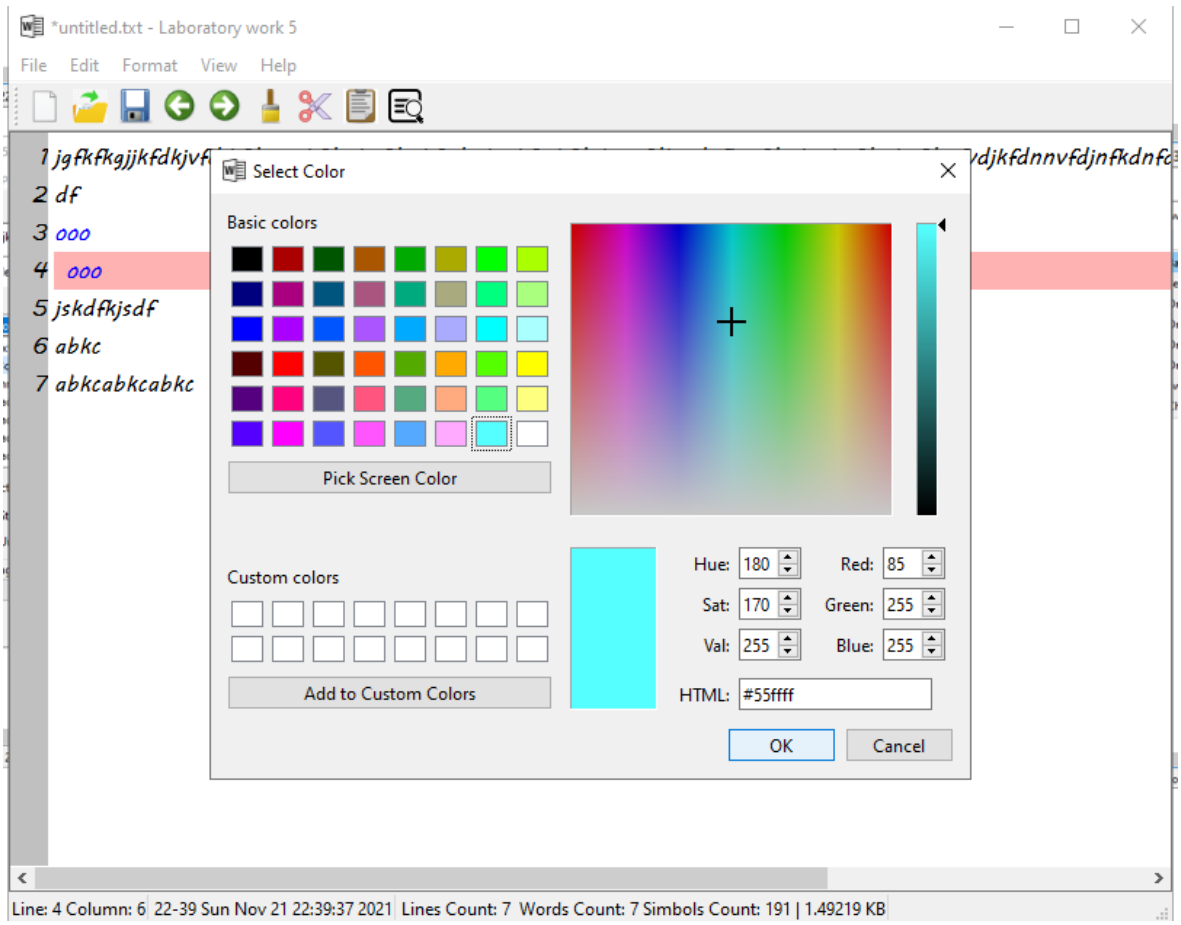

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjv
2 df
3 ooo
4 ooo
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 6 22-39 Sun Nov 21 22:39:37 2021 Lines Count: 7 Words Count: 7 Simbols Count: 191 | 1.49219 KB

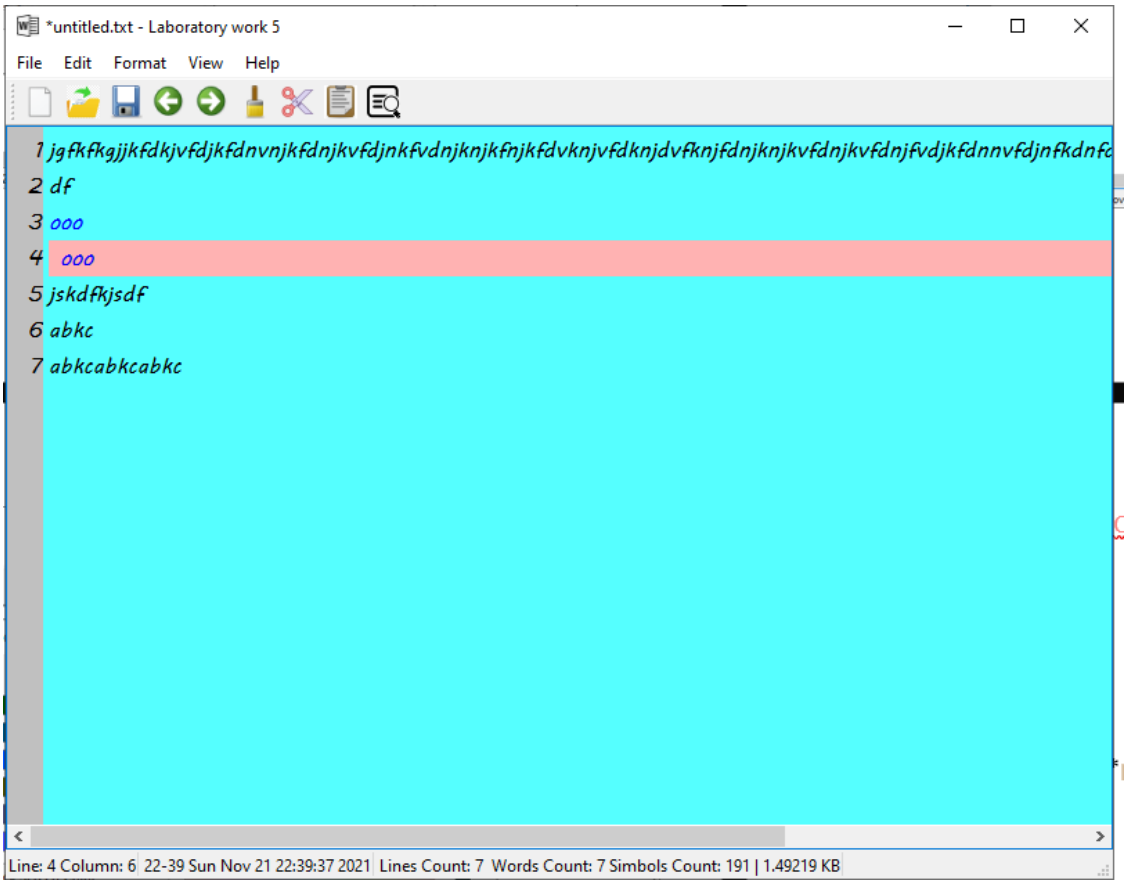

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
Word wrap
Font Selection
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjvf
2 df
3 ooo
4 ooo
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 6 22-39 Sun Nov 21 22:39:37 2021 Lines Count: 7 Words Count: 7 Simbols Count: 191 | 1.49219 KB

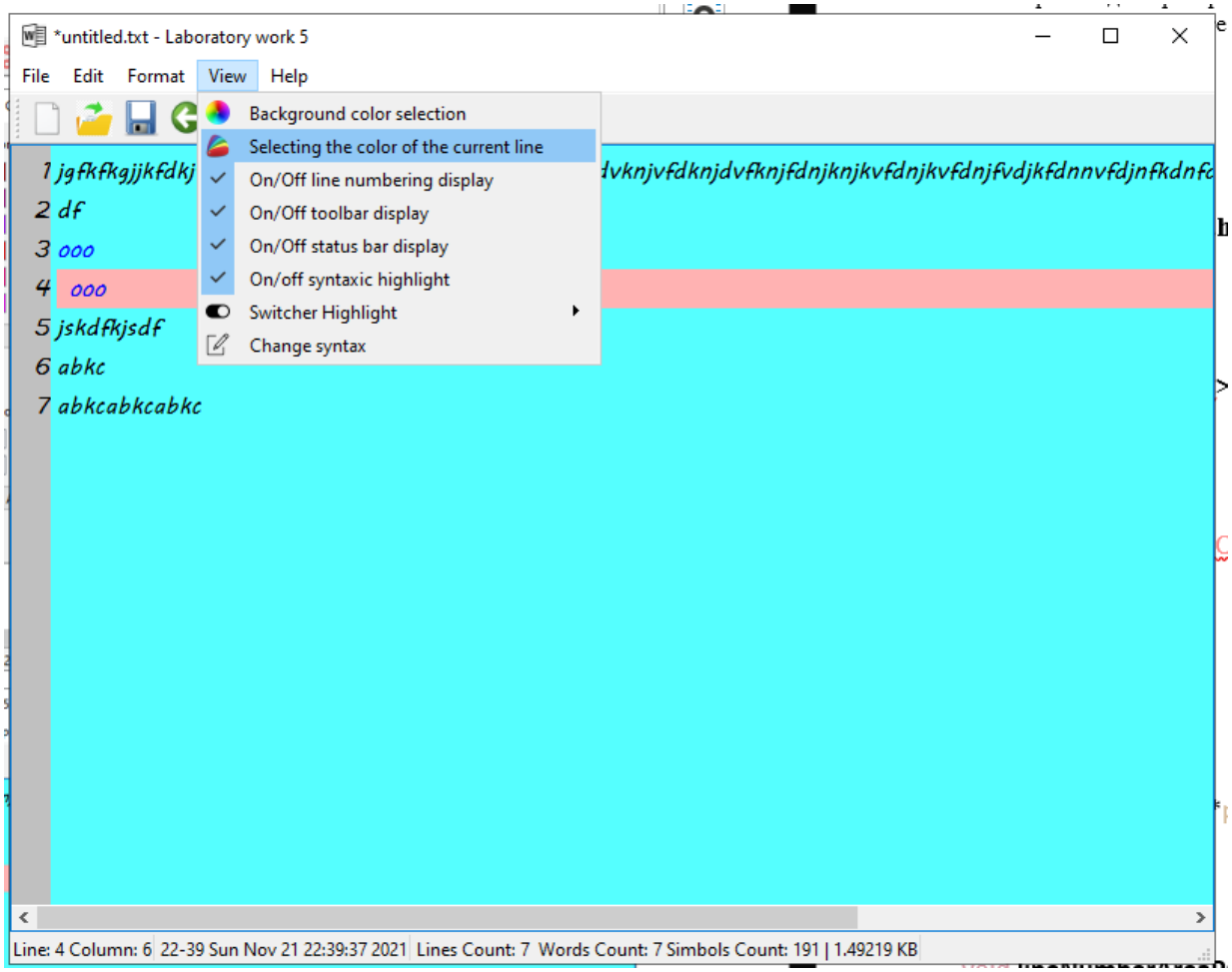

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfdjnjkfdvnjkvfnjnkvfdjnjvfdnjfvdnjvf
2 df
3 ooo
4 ooo
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Select Font
Font
MV Boli
MT Extra
MV Boli
Myanmar Text
Myriad Arabic
Myriad Hebrew
Myriad Pro
Myriad Pro Cond
Font style
Обычный
Обычный
Size
12
6
7
8
9
10
11
12
Effects
Strikeout
Underline
Writing System
Any
Sample
AaBbYyZz
OK
Cancel
Line: 4 Column: 6 22-39 Sun Nov 21 22:39:37 2021 Lines Count: 7 Words Count: 7 Simbols Count: 191 | 1.49219 KB

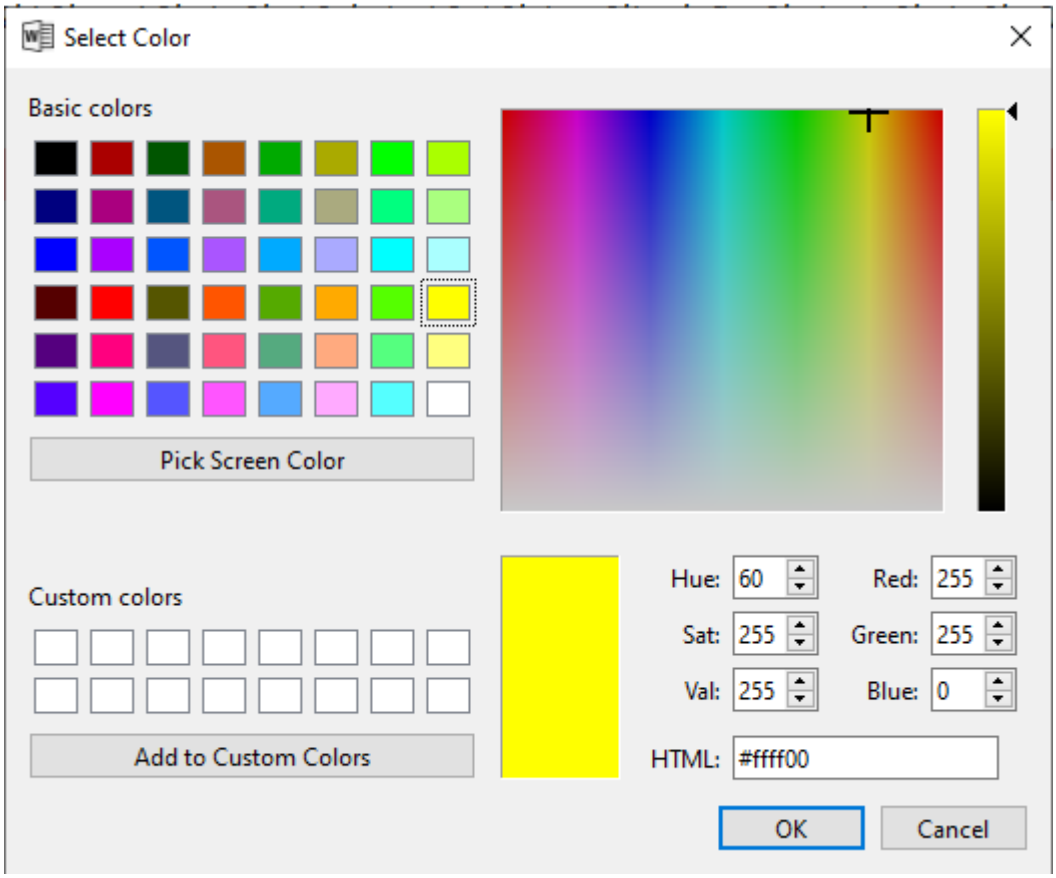

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfc
2 df
3 ooo
4 ooo
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 6 22-39 Sun Nov 21 22:39:37 2021 Lines Count: 7 Words Count: 7 Simbols Count: 191 | 1.49219 KB

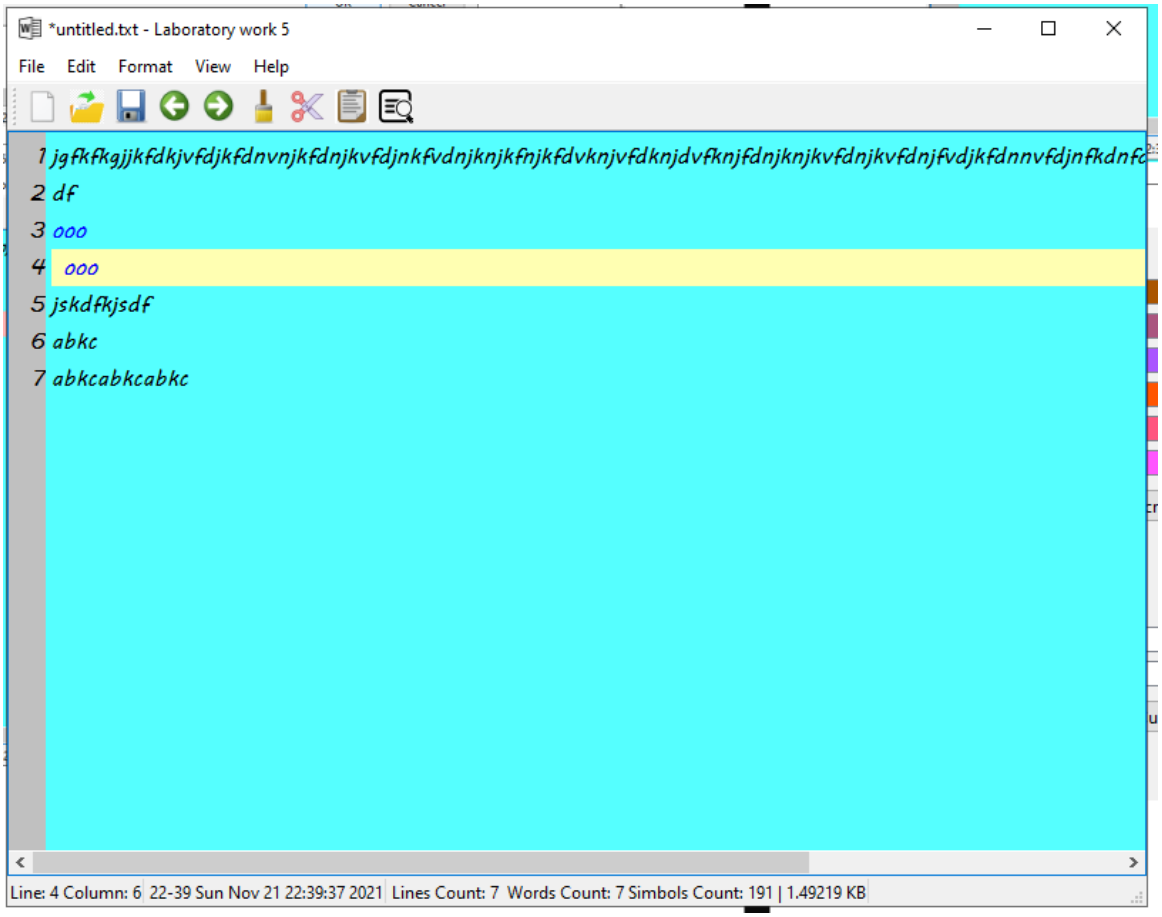

*untitled.txt - Laboratory work 5
File Edit Format View Help
Background color selection
Selecting the color of the current line
On/Off line numbering display
On/Off toolbar display
On/Off status bar display
On/off syntaxic highlight
Switcher Highlight
Change syntax
1 jgfkfkgjjkfdkj
2 df
3 ooo
4 ooo
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 6 22-39 Sun Nov 21 22:39:37 2021 Lines Count: 7 Words Count: 7 Simbols Count: 191 | 1.49219 KB

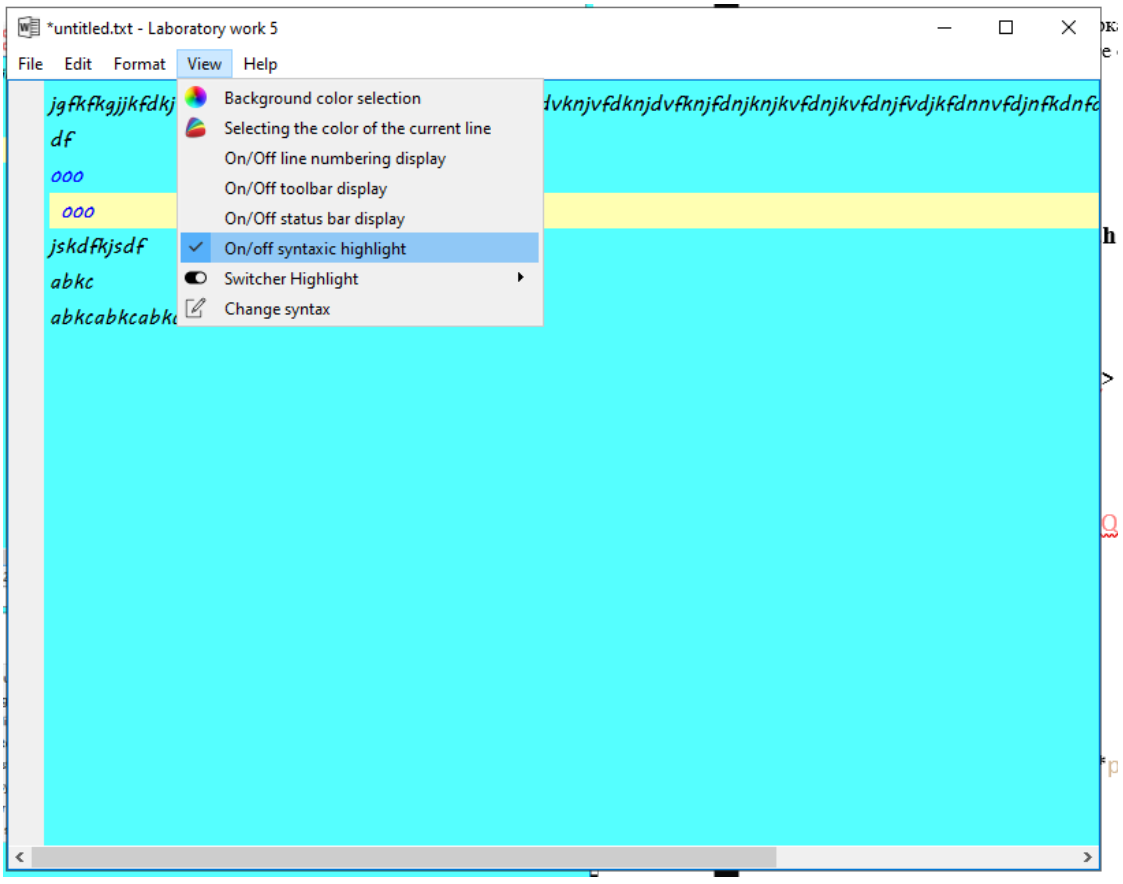

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvf
2 df
3 ooo
4 ooo
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Select Color
Basic colors
Pick Screen Color
Custom colors
Add to Custom Colors
Hue: 180
Sat: 170
Val: 255
Red: 85
Green: 255
Blue: 255
HTML: #55ffff
OK
Cancel
Line: 4 Column: 6 22-39 Sun Nov 21 22:39:37 2021 Lines Count: 7 Words Count: 7 Simbols Count: 191 | 1.49219 KB

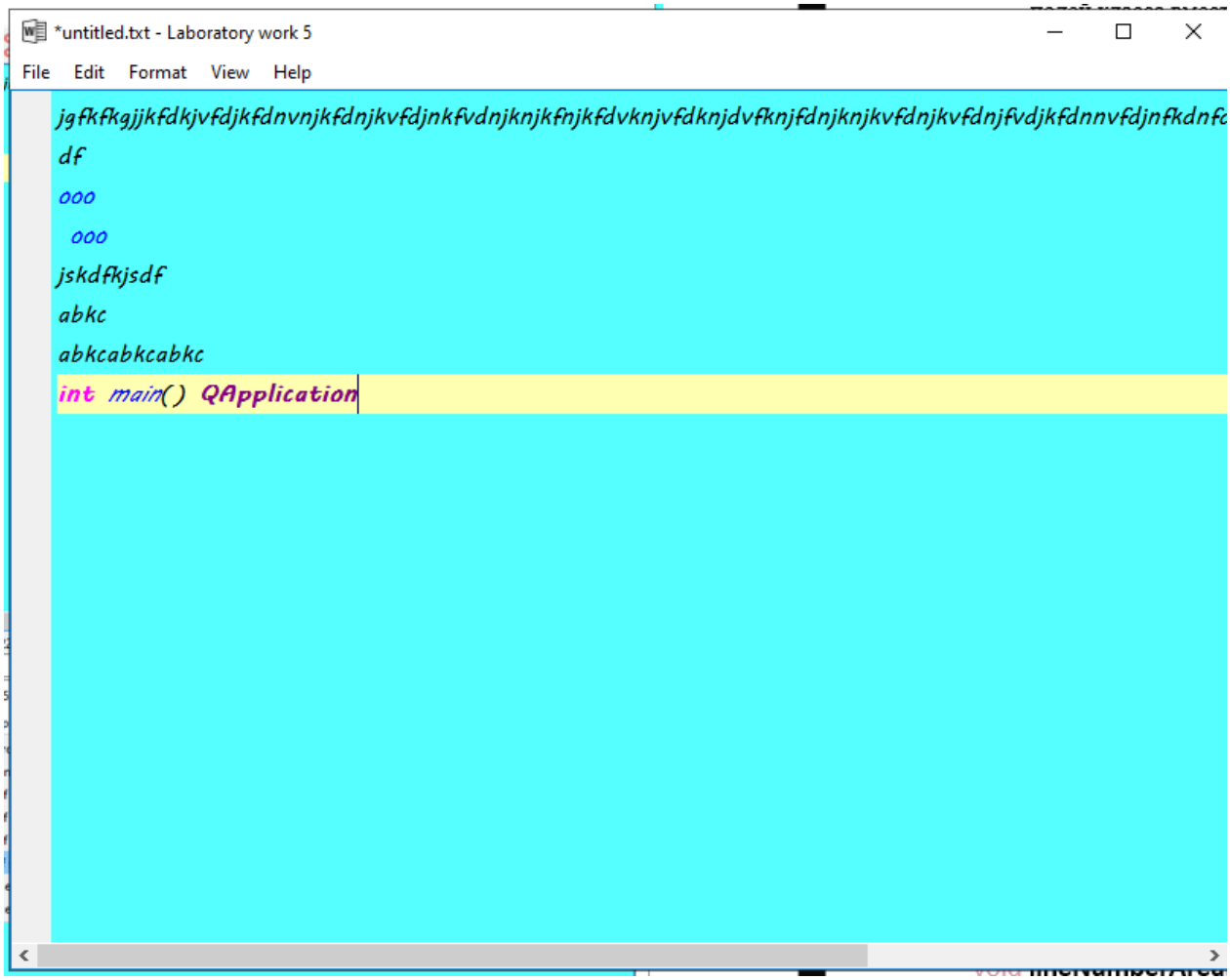

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfc
2 df
3 ooo
4 ooo
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 6 22-39 Sun Nov 21 22:39:37 2021 Lines Count: 7 Words Count: 7 Simbols Count: 191 | 1.49219 KB


*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
Background color selection
Selecting the color of the current line
On/Off line numbering display
On/Off toolbar display
On/Off status bar display
On/off syntaxic highlight
Switcher Highlight
Change syntax
1 jgfkfkgjjkfdkj
2 df
3 ooo
4 ooo
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 6
22-39 Sun Nov 21 22:39:37 2021
Lines Count: 7
Words Count: 7
Simbols Count: 191 | 1.49219 KB


Select Color
Basic colors
Pick Screen Color
Custom colors
Add to Custom Colors
Hue: 60
Sat: 255
Val: 255
Red: 255
Green: 255
Blue: 0
HTML: #ffff00
OK
Cancel

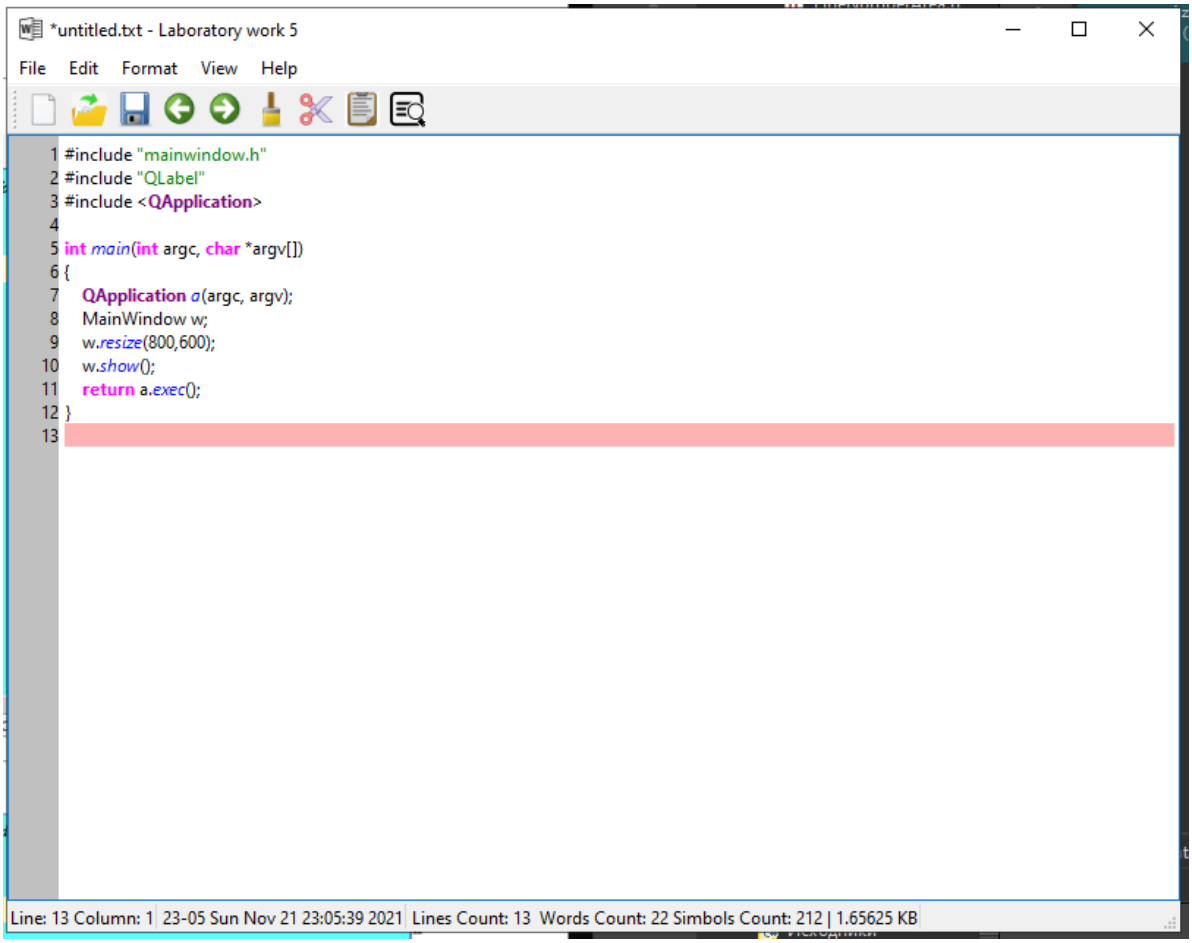

*untitled.txt - Laboratory work 5
File Edit Format View Help
1 jgfkfkgjjkfdkjvfdjkfdnvnjkfdnjkvfdjnkfvdnjknjkfnjkfdvknjvfdknjdvfknjfdnjknjkvfdnjkvfdnjfvdjkfdnnvfdjnfkdnfd
2 df
3 ooo
4 ooo
5 jskdfkjsdf
6 abkc
7 abkcabkcabkc
Line: 4 Column: 6 22-39 Sun Nov 21 22:39:37 2021 Lines Count: 7 Words Count: 7 Simbols Count: 191 | 1.49219 KB

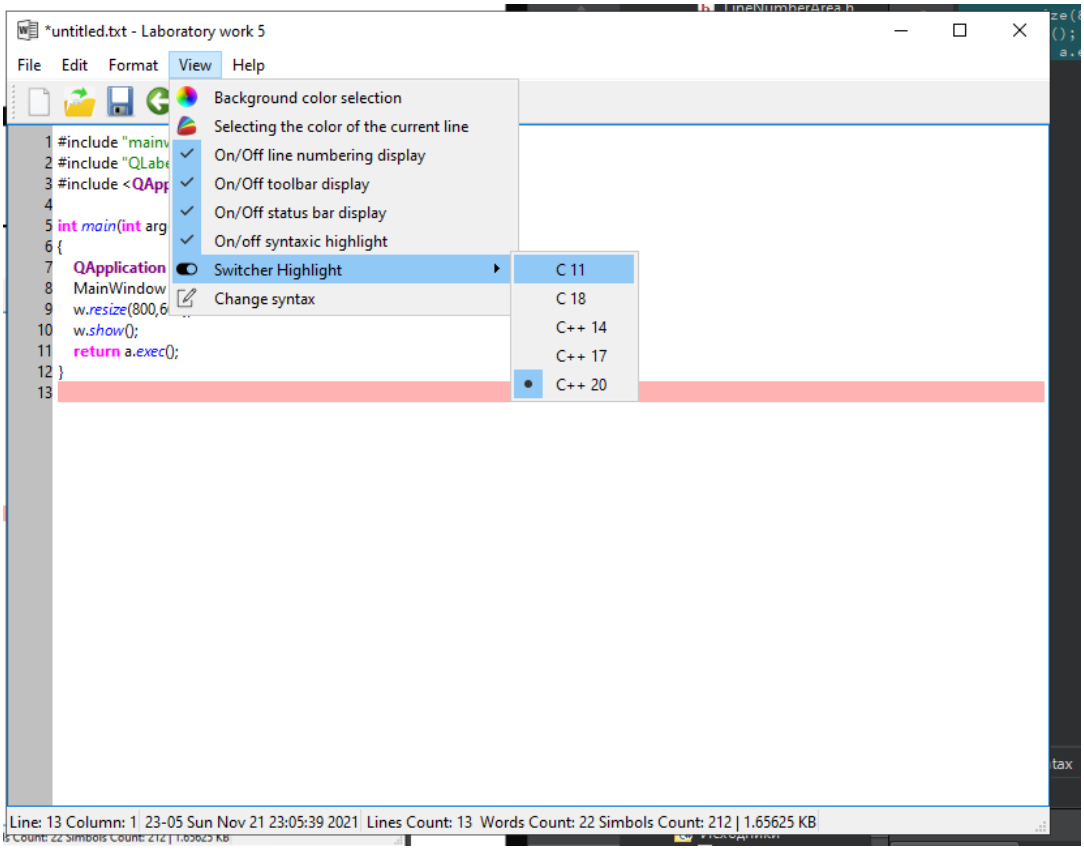

*untitled.txt - Laboratory work 5
File Edit Format View Help
Background color selection
Selecting the color of the current line
On/Off line numbering display
On/Off toolbar display
On/Off status bar display
On/off syntaxic highlight
Switcher Highlight
Change syntax

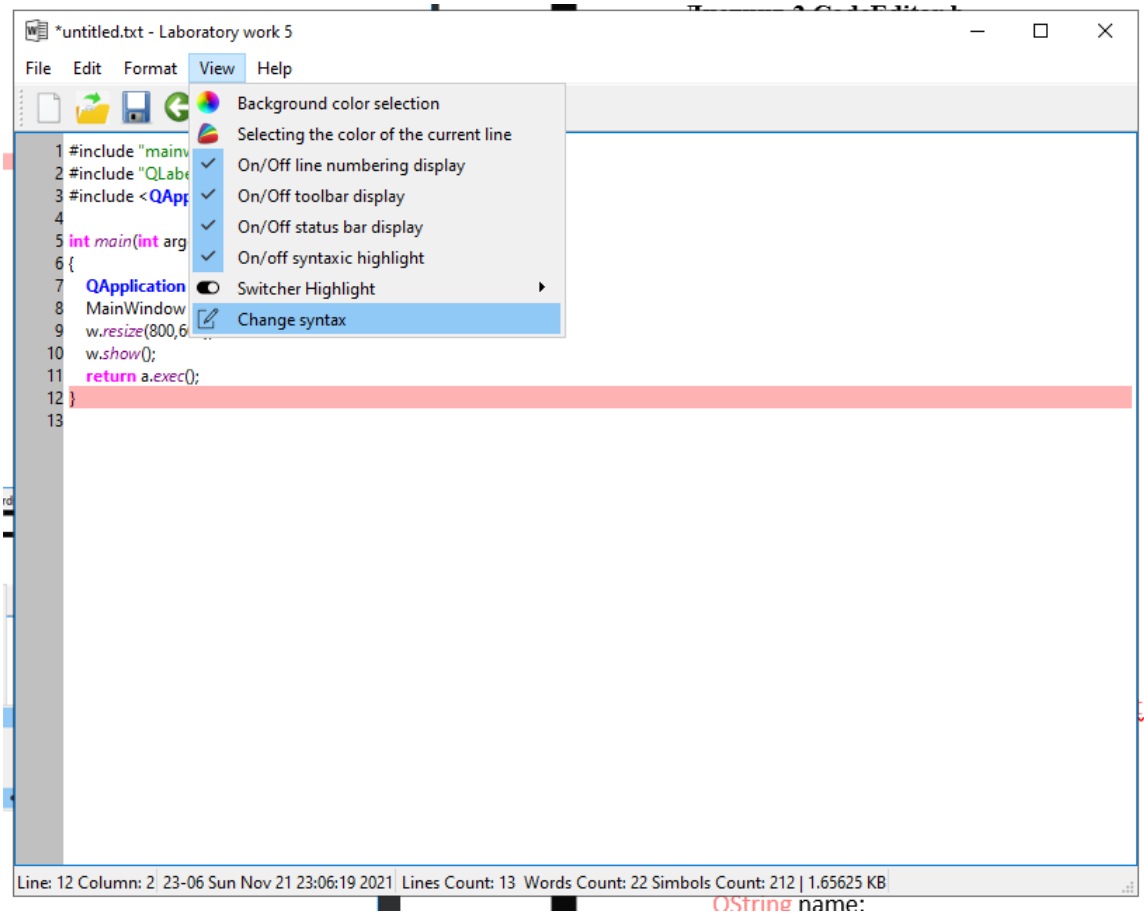


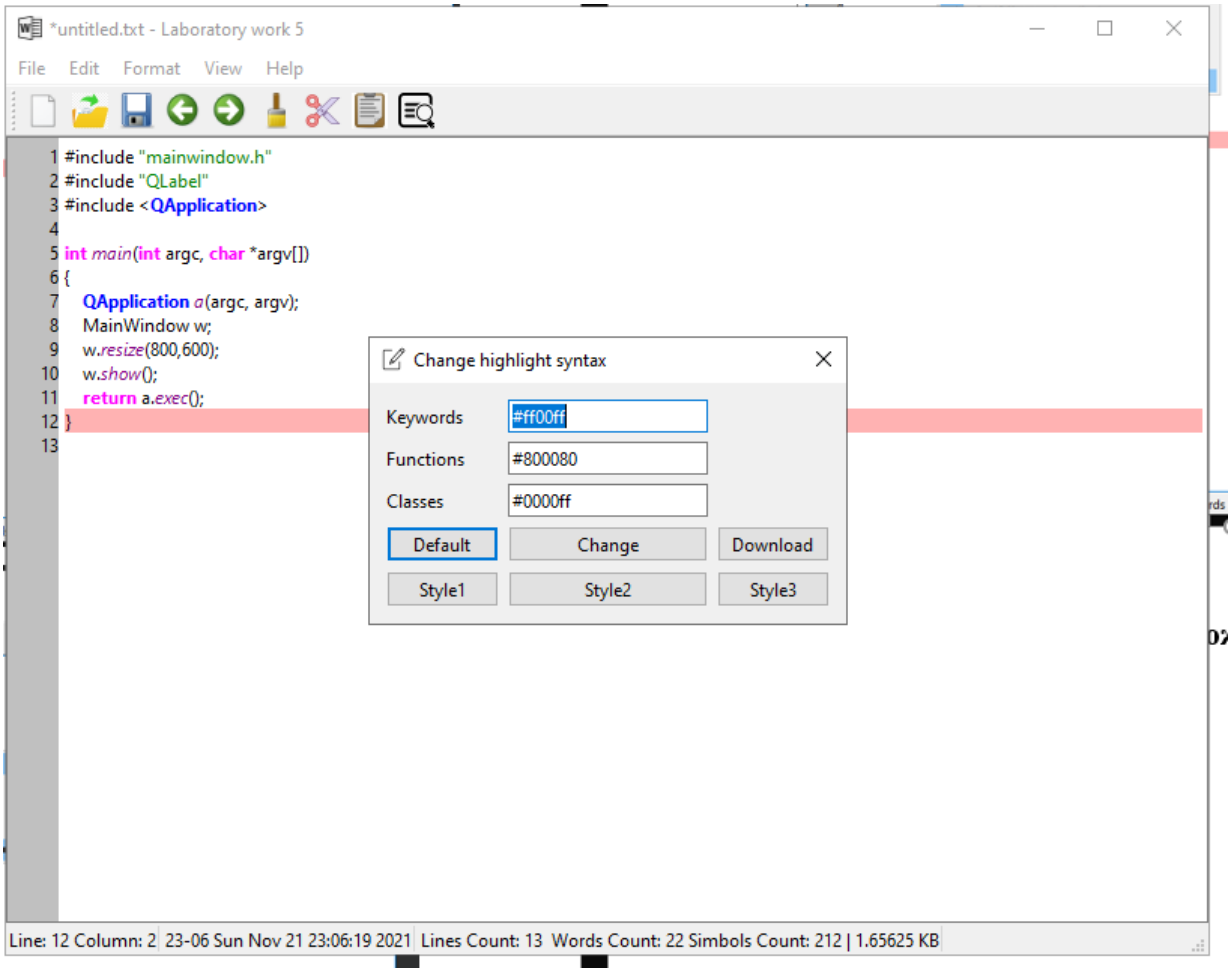
Test №2

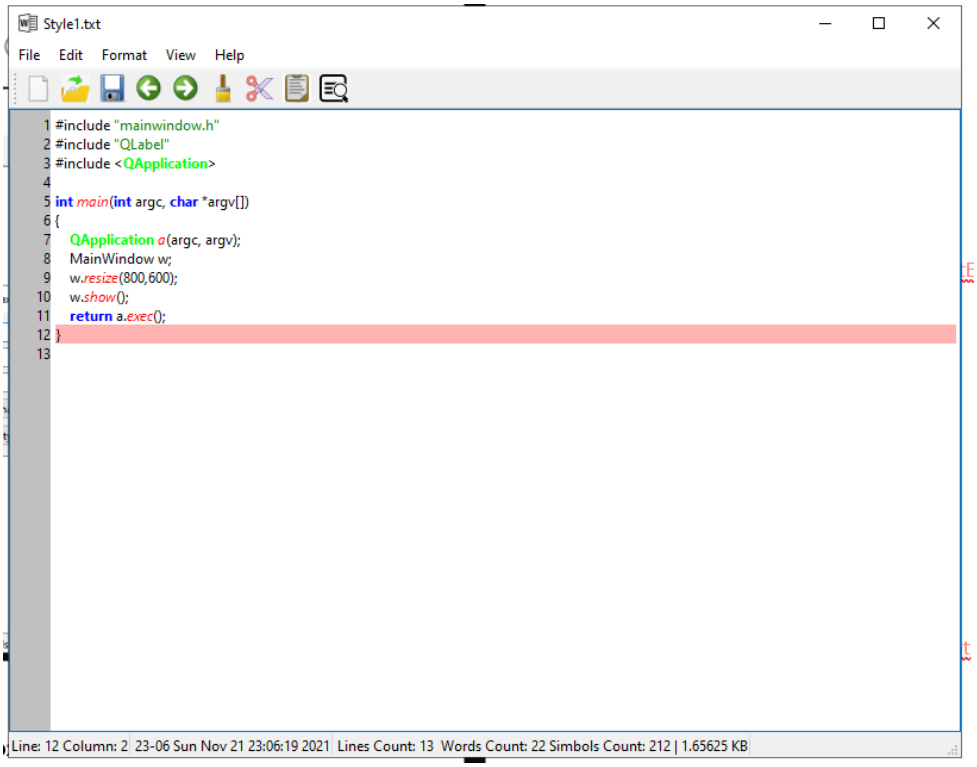
Проверим оставшиеся пункты меню View и Help.

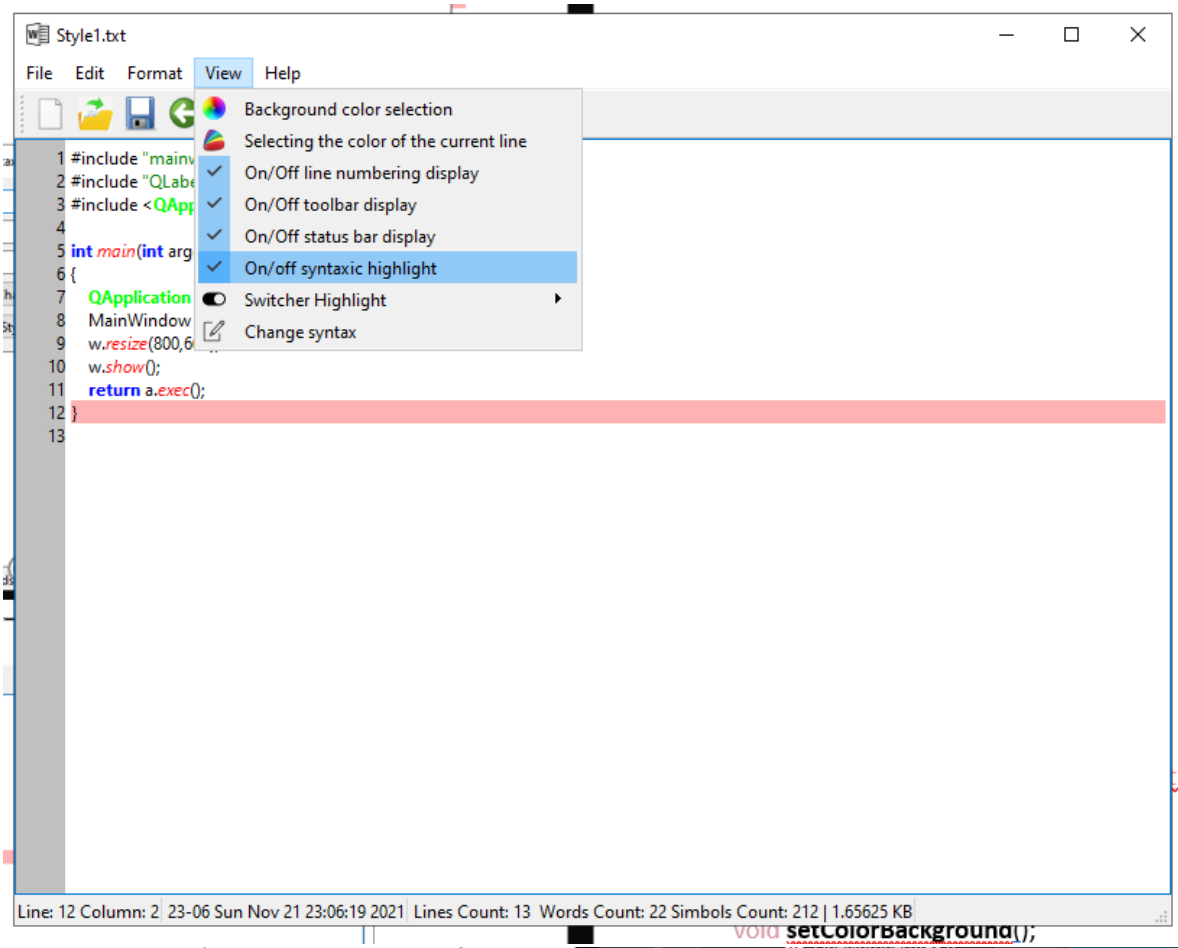

*untitled.txt - Laboratory work 5
File Edit Format View Help
#include "mainwindow.h"
#include "QLabel"
#include <QApplication>
int main(int argc, char *argv[])
{
QApplication a(argc, argv);
MainWindow w;
w.resize(800,600);
w.show();
return a.exec();
}
Line: 13 Column: 1 23-05 Sun Nov 21 23:05:39 2021 Lines Count: 13 Words Count: 22 Simbols Count: 212 | 1.65625 KB

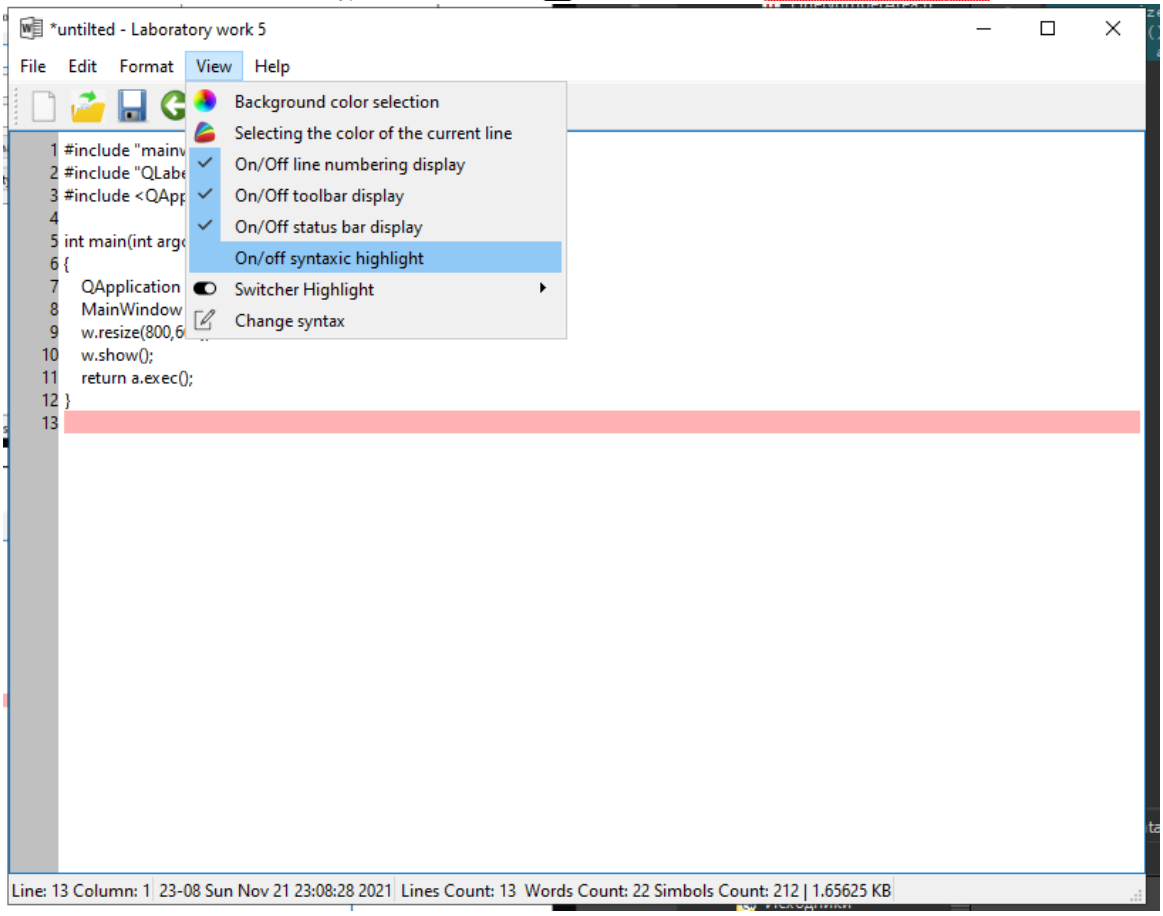

*untitled.txt - Laboratory work 5
File Edit Format View Help
Background color selection
Selecting the color of the current line
On/Off line numbering display
On/Off toolbar display
On/Off status bar display
On/off syntaxic highlight
Switcher Highlight
Change syntax
C 11
C 18
C++ 14
C++ 17
C++ 20
Line: 13 Column: 1 23-05 Sun Nov 21 23:05:39 2021 Lines Count: 13 Words Count: 22 Simbols Count: 212 | 1.65625 KB

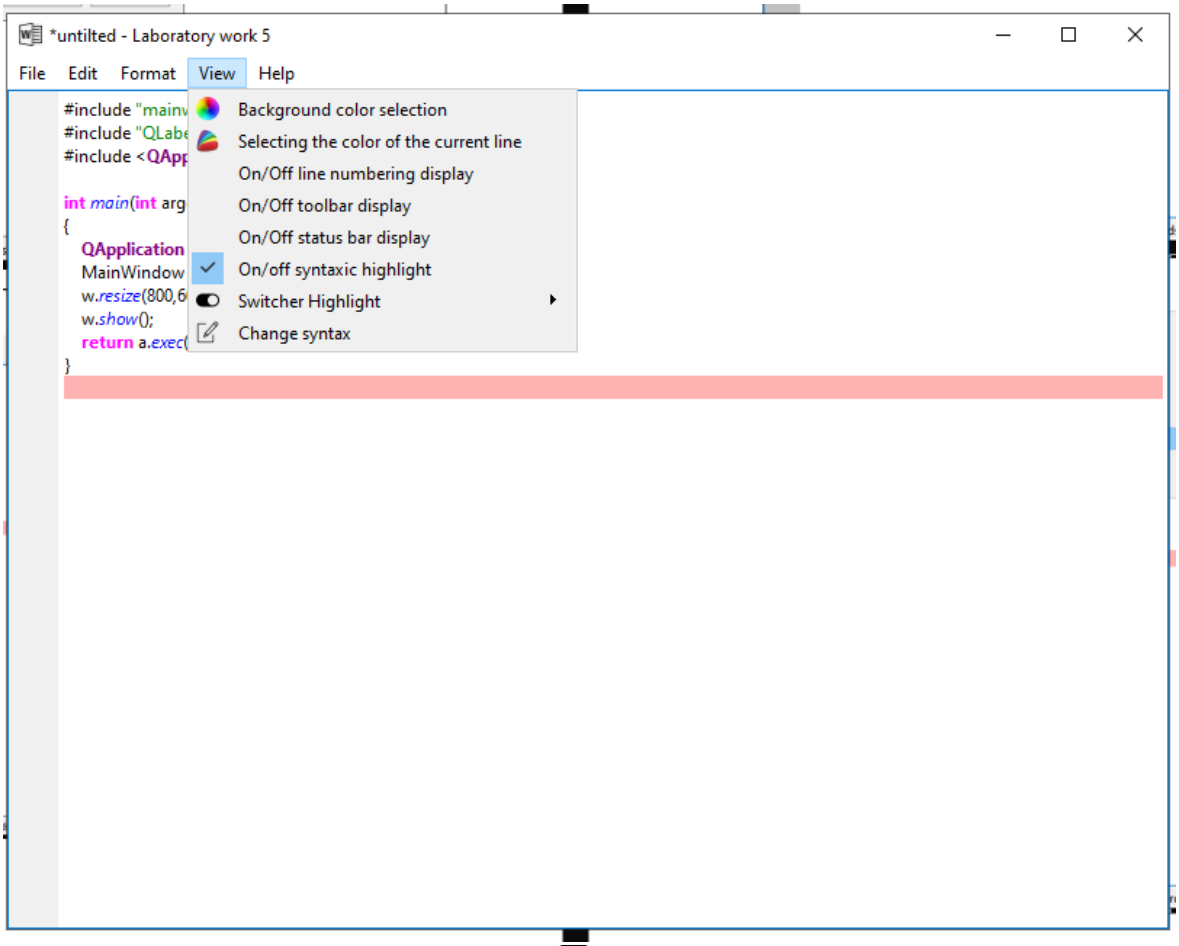

*untitled.txt - Laboratory work 5
File
Edit
Format
View
Help
Background color selection
Selecting the color of the current line
On/Off line numbering display
On/Off toolbar display
On/Off status bar display
On/off syntaxic highlight
Switcher Highlight
Change syntax
#include "mainv
#include "QLabe
#include <QApp
int main(int arg
{
QApplication
MainWindow
w.resize(800,6
w.show();
return a.exec();
}
Line: 12 Column: 2
23-06 Sun Nov 21 23:06:19 2021
Lines Count: 13
Words Count: 22 Simbols Count: 212 | 1.65625 KB

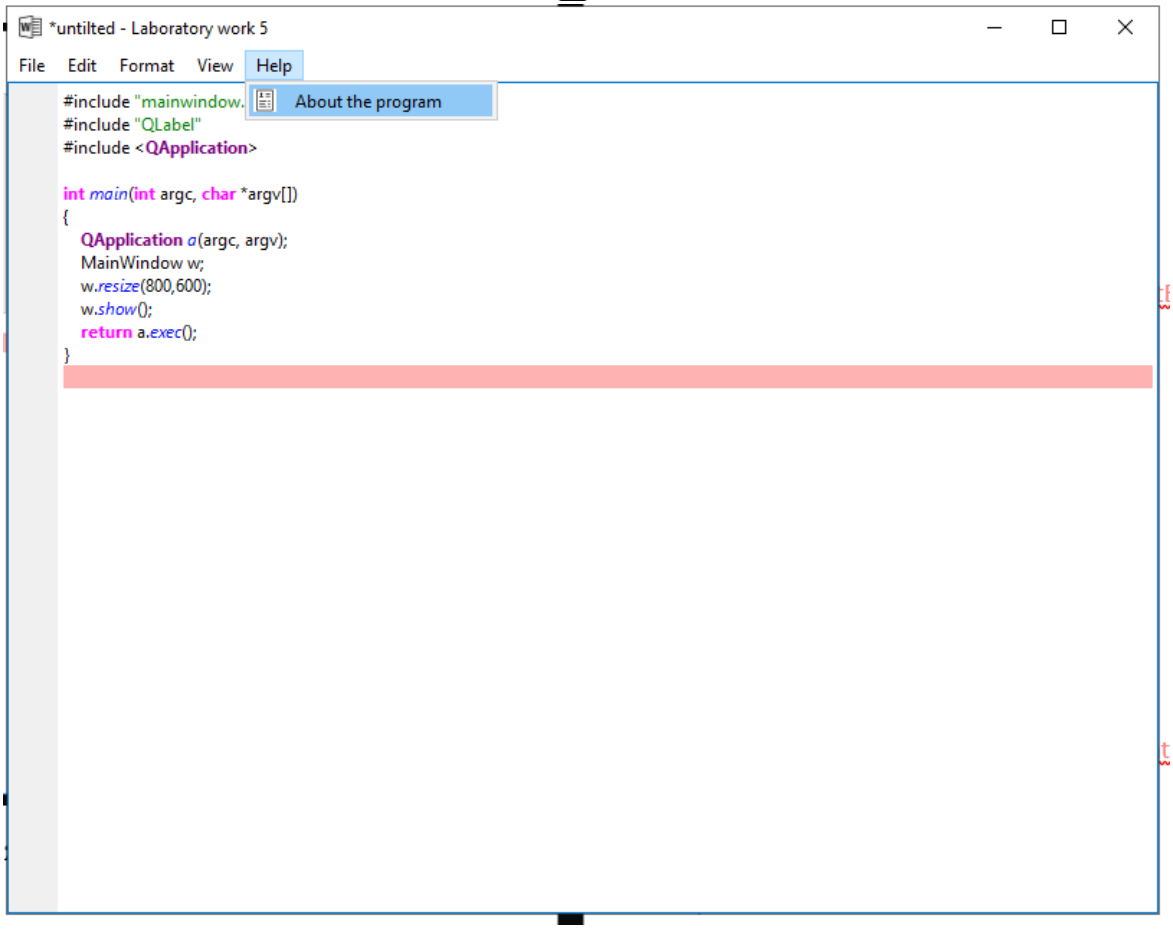

*untitled.txt - Laboratory work 5
File Edit Format View Help
#include "mainwindow.h"
#include "QLabel"
#include <QApplication>
int main(int argc, char *argv[])
{
QApplication a(argc, argv);
MainWindow w;
w.resize(800,600);
w.show();
return a.exec();
}
Change highlight syntax
Keywords
#ff00ff
Functions
#800080
Classes
#0000ff
Default
Change
Download
Style1
Style2
Style3
Line: 12 Column: 2 23-06 Sun Nov 21 23:06:19 2021 Lines Count: 13 Words Count: 22 Simbols Count: 212 | 1.65625 KB

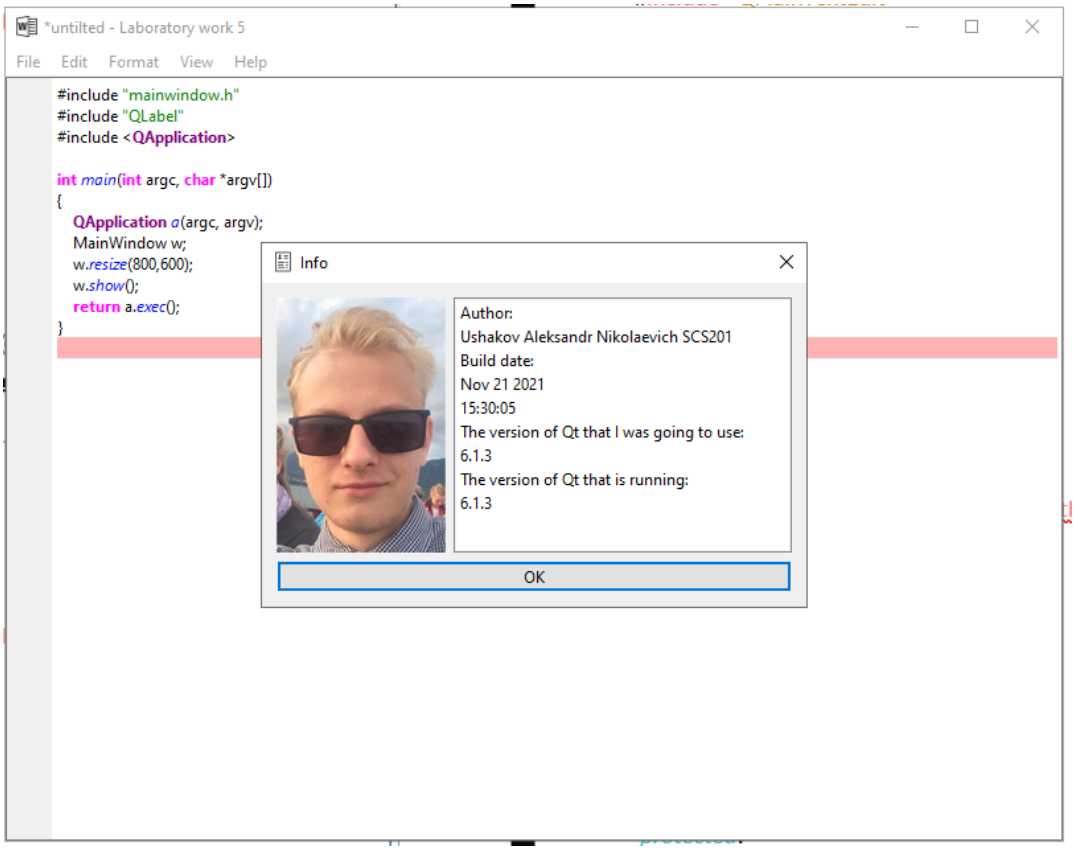

Style1.txt
File Edit Format View Help
#include "mainwindow.h"
#include "QLabel"
#include <QApplication>
int main(int argc, char *argv[])
{
QApplication a(argc, argv);
MainWindow w;
w.resize(800,600);
w.show();
return a.exec();
}
Line: 12 Column: 2 23-06 Sun Nov 21 23:06:19 2021 Lines Count: 13 Words Count: 22 Simbols Count: 212 | 1.65625 KB

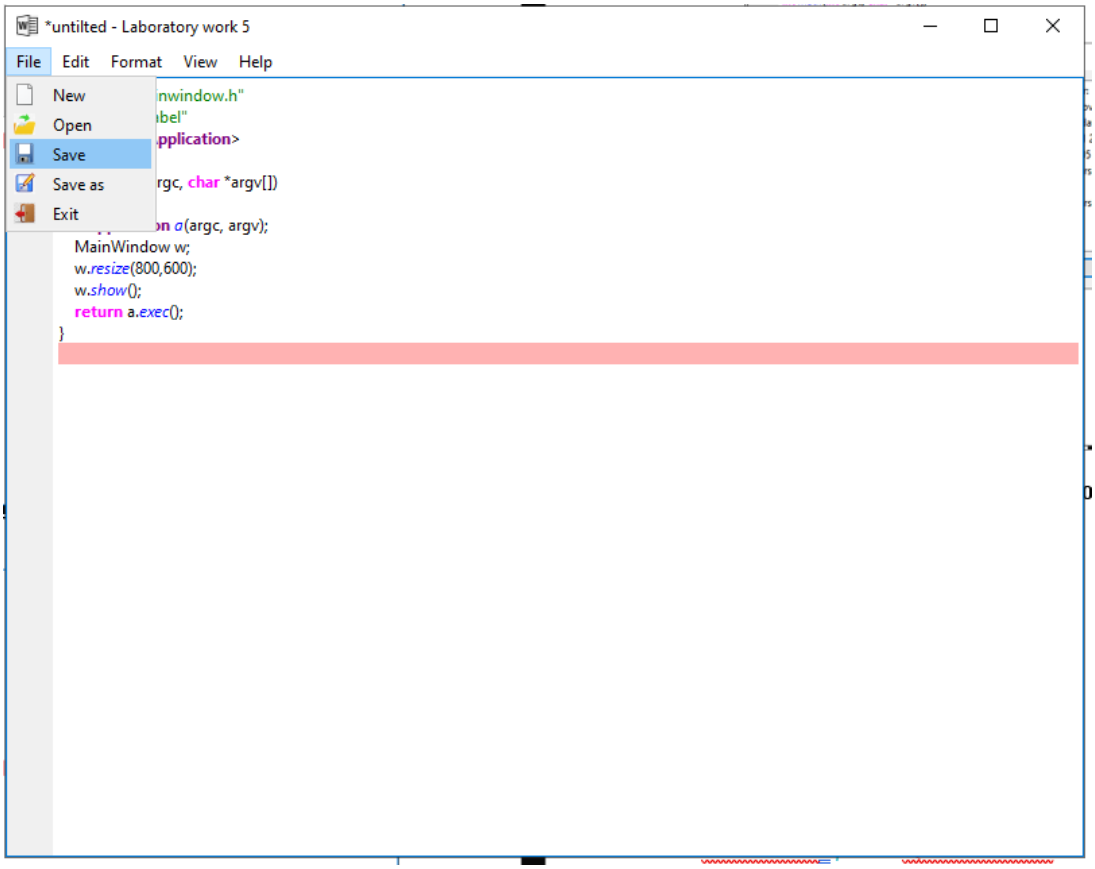

Style1.txt
File Edit Format View Help
Background color selection
Selecting the color of the current line
On/Off line numbering display
On/Off toolbar display
On/Off status bar display
On/off syntaxic highlight
Switcher Highlight
Change syntax
#include "mainv
#include "QLabe
#include <QApp
int main(int arg
{
QApplication
MainWindow
w.resize(800,6
w.show();
return a.exec();
}
Line: 12 Column: 2 23-06 Sun Nov 21 23:06:19 2021 Lines Count: 13 Words Count: 22 Simbols Count: 212 | 1.65625 KB


*untilted - Laboratory work 5
File Edit Format View Help
Background color selection
Selecting the color of the current line
On/Off line numbering display
On/Off toolbar display
On/Off status bar display
On/off syntaxic highlight
Switcher Highlight
Change syntax
#include "mainv
#include "QLabe
#include <QApp
int main(int arg
{
QApplication
MainWindow
w.resize(800,6
w.show();
return a.exec();
}
Line: 13 Column: 1 23-08 Sun Nov 21 23:08:28 2021 Lines Count: 13 Words Count: 22 Simbols Count: 212 | 1.65625 KB


*untilted - Laboratory work 5
File
Edit
Format
View
Help
Background color selection
Selecting the color of the current line
On/Off line numbering display
On/Off toolbar display
On/Off status bar display
On/off syntaxic highlight
Switcher Highlight
Change syntax


*untilted - Laboratory work 5
File
Edit
Format
View
Help
About the program
#include "mainwindow.
#include "QLabel"
#include <QApplication>
int main(int argc, char *argv[])
{
QApplication a(argc, argv);
MainWindow w;
w.resize(800,600);
w.show();
return a.exec();
}


*untilted - Laboratory work 5
File Edit Format View Help
#include "mainwindow.h"
#include "QLabel"
#include <QApplication>
int main(int argc, char *argv[])
{
QApplication a(argc, argv);
MainWindow w;
w.resize(800,600);
w.show();
return a.exec();
}
Info
Author:
Ushakov Aleksandr Nikolaevich SCS201
Build date:
Nov 21 2021
15:30:05
The version of Qt that I was going to use:
6.1.3
The version of Qt that is running:
6.1.3
OK


*untilted - Laboratory work 5
File Edit Format View Help
New
Open
Save
Save as
Exit
MainWindow w;
w.resize(800,600);
w.show();
return a.exec();
}

Приложение А

Листинг-2 CodeEditor.h

```
#ifndef CODEEDITOR_H
#define CODEEDITOR_H
#include <QPlainTextEdit>
#include <QColorDialog>
#include <QFontDialog>
class CodeEditor : public QPlainTextEdit
{
    Q_OBJECT

public:
    CodeEditor(QWidget *parent = 0);



    void lineNumberAreaPaintEvent(QPaintEvent *event);
    int lineNumberAreaWidth();
    void setColorLine();
    void setColorBackground();
    QString name;
    QWidget* getLineNumberArea();
```

```
protected:
    void resizeEvent(QResizeEvent *event);
    bool eventFilter(QObject *object,QEvent *event);



private slots:
    void updateLineNumberAreaWidth(int newBlockCount);
    void highlightCurrentLine();
    void updateLineNumberArea(const QRect &, int);
    void undo1();
    void redo1();

private:
    QWidget *lineNumberArea;
    QColor current_color="red";
    QString s="";
    QColorDialog *DialogLine;
    QColorDialog *DialogALL;
    QTextEdit::ExtraSelection selection;


};
#endif // CODEEDITOR_H
```

Листинг-3 CodeEditor.cpp

```
#include "CodeEditor.h"
#include <QTextBlock>
#include <QPainter>
#include <QColor>
#include <QFile>
#include <QDateTime>
#include "LineNumberArea.h"
```

```
CodeEditor::CodeEditor(QWidget *parent) : QPlainTextEdit(parent)
 {

     lineNumberArea = new LineNumberArea(this);
     installEventFilter(this);

     connect(this, SIGNAL(blockCountChanged(int)), this,
SLOT(updateLineNumberAreaWidth(int)));
     connect(this, SIGNAL(updateRequest(QRect,int)), this,
SLOT(updateLineNumberArea(QRect,int)));
     connect(this, SIGNAL(cursorPositionChanged()), this,
SLOT(highlightCurrentLine()));

     updateLineNumberAreaWidth(0);
     highlightCurrentLine();
 }

void CodeEditor::undo1(){
   this->undo();
   QFile file(name);
   file.open(QIODevice::Append);
   QDateTime cur=QDateTime::currentDateTime();
   QString s="\n Undo "+cur.toString();
   QByteArray byte=s.toUtf8();
   file.write(byte);
   file.close();
}
void CodeEditor::redo1(){
    this->redo();
    QFile file(name);
    file.open(QIODevice::Append);
    QDateTime cur=QDateTime::currentDateTime();
    QString s="\n Redo "+cur.toString();
    QByteArray byte=s.toUtf8();
    file.write(byte);
    file.close();
}
bool CodeEditor::eventFilter(QObject *object,QEvent *e){
    QFile file(name);
    file.open(QIODevice::Append);int event=static_cast<QKeyEvent*>(e)-
>key();
    if (QEvent::KeyPress==e->type()){


        if(event==Qt::Key_Enter||event==Qt::Key_Return){
            QDateTime cur=QDateTime::currentDateTime();
            QString s="pressEnter "+cur.toString()+"\n";
            QByteArray byte=s.toUtf8();
            file.write(byte);

        }
```

```
else if(event==Qt::Key_Backspace||event==Qt::Key_Delete){
    QDateTime cur=QDateTime::currentDateTime();
    QString s="press delete symbol "+cur.toString()+"\n";

    QByteArray byte=s.toUtf8();
    file.write(byte);
}
else {
    QString str="",s="";int k=event;
    while (k>0){
        s="";
        int a=k%100;
                        if (a==32)
                        s=' ';
                        if (a==33)
                        s='!';
                        if (a==34)
                        s='\'';
                        if (a==35)
                        s='#';
                        if (a==36)
                        s='$';
                        if (a==37)
                        s='%';
                        if (a==38)
                        s='&';
                        if (a==39)
                        s='\'';
                        if (a==40)
                        s='(';
                        if (a==41)
                        s=')';
                        if (a==42)
                        s='*';
                        if (a==43)
                        s='+';
                        if (a==44)
                        s=',';
                        if (a==45)
                        s='-';
                        if (a==46)
                        s='.';
                        if (a==47)
                        s='/';
                        if (a==48)
                        s='\0';
                        if (a==49)
                        s='1';
                        if (a==50)
                        s='2';
                        if (a==51)
```

```
s='3';
if (a==52)
s='4';
if (a==53)
s='5';
if (a==54)
s='6';
if (a==55)
s='7';
if (a==56)
s='8';
if (a==57)
s='9';
if (a==58)
s=':';
if (a==59)
s=';';
if (a==60)
s='<';
if (a==61)
s='=';
if (a==62)
s='>';
if (a==63)
s='\?';
if (a==64)
s='@';
if (a==65)
s='a';
if (a==66)
s='b';
if (a==67)
s='c';
if (a==68)
s='d';
if (a==69)
s='e';
if (a==70)
s='f';
if (a==71)
s='g';
if (a==72)
s='h';
if (a==73)
s='i';
if (a==74)
s='j';
if (a==75)
s='k';
if (a==76)
s='l';
```

```
                        if (a==77)
                        s='m';
                        if (a==78)
                        s='n';
                        if (a==79)
                        s='o';
                        if (a==80)
                        s='p';
                        if (a==81)
                        s='q';
                        if (a==82)
                        s='r';
                        if (a==83)
                        s='s';
                        if (a==84)
                        s='t';
                        if (a==85)
                        s='u';
                        if (a==86)
                        s='v';
                        if (a==87)
                        s='w';
                        if (a==88)
                        s='x';
                        if (a==89)
                        s='y';
                        if (a==90)
                        s='z';
                        if (a==91)
                        s='[';
                        if (a==92)
                        s='\\';
                        if (a==93)
                        s=']';
                        if (a==94)
                        s='^';
                        if (a==95)
                        s='_';
                        this->s=s;

            str=s+str;
            k/=100;
        }

        QDateTime cur=QDateTime::currentDateTime();
        //QString s1=event;
        str+="  "+cur.toString()+"\n";
        QByteArray byte=str.toUtf8();
        file.write(byte);

}
```

```
    }
        file.flush();
        file.close();


    return QPlainTextEdit::eventFilter(object,e);
}
void CodeEditor::setColorLine(){
    DialogLine = new QColorDialog (this);
    DialogLine->exec();
    current_color = DialogLine->selectedColor();

selection.format.setBackground(QColor(QColor(current_color).lighter(16
0)));
    highlightCurrentLine();
}
void CodeEditor::setColorBackground(){
    DialogALL= new QColorDialog (this);
    DialogALL -> exec();
    QPalette pal = this->palette();
    pal.setColor(QPalette::Active, QPalette::Base, DialogALL-
>selectedColor());
    pal.setColor(QPalette::Inactive, QPalette::Base, Qt::white);
    this->setPalette(pal);
}

int CodeEditor::lineNumberAreaWidth()
 {
     int digits = 1;
     int max = qMax(1, blockCount());
     while (max >= 10) {
         max /= 10;
         ++digits;
     }

     int space = 3 + fontMetrics().maxWidth();

     return space;
 }
void CodeEditor::updateLineNumberAreaWidth(int /* newBlockCount */)
 {
     setViewportMargins(lineNumberAreaWidth(), 0, 0, 0);
 }
void CodeEditor::updateLineNumberArea(const QRect &rect, int dy)
 {
     if (dy)
         lineNumberArea->scroll(0, dy);
     else
         lineNumberArea->update(0, rect.y(), lineNumberArea->width(),
rect.height());
```

```
    if (rect.contains(viewport()->rect()))
        updateLineNumberAreaWidth(0);
 }
void CodeEditor::resizeEvent(QResizeEvent *e)
 {
    QPlainTextEdit::resizeEvent(e);

    QRect cr = contentsRect();
    lineNumberArea->setGeometry(QRect(cr.left(), cr.top(),
lineNumberAreaWidth(), cr.height()));
 }
void CodeEditor::highlightCurrentLine()
 {
    QList<QTextEdit::ExtraSelection> extraSelections;

    if (!isReadOnly()) {
        QTextEdit::ExtraSelection selection;

        QColor lineColor = QColor(current_color).lighter(170);

        selection.format.setBackground(lineColor);
        selection.format.setProperty(QTextFormat::FullWidthSelection,
true);
        selection.cursor = textCursor();
        selection.cursor.clearSelection();
        extraSelections.append(selection);
    }

    setExtraSelections(extraSelections);
 }
void CodeEditor::lineNumberAreaPaintEvent(QPaintEvent *event)
 {
    QPainter painter(lineNumberArea);
    painter.fillRect(event->rect(), Qt::lightGray);
    QTextBlock block = firstVisibleBlock();
        int blockNumber = block.blockNumber();
        int top = (int)
blockBoundingGeometry(block).translated(contentOffset()).top();
        int bottom = top + (int) blockBoundingRect(block).height();
        while (block.isValid() && top <= event->rect().bottom()) {
                if (block.isVisible() && bottom >= event-
>rect().top()) {
                    QString number = QString::number(blockNumber +
1);
                    painter.setPen(Qt::black);
                    painter.drawText(0, top, lineNumberArea-
>width(), fontMetrics().height(),
                                     Qt::AlignRight, number);
                }
                block = block.next();
                top = bottom;
```

```
                    bottom = top + (int)
blockBoundingRect(block).height();
                    ++blockNumber;
                }
            }
QWidget* CodeEditor::getLineNumberArea(){
    return lineNumberArea;
}
```

Приложение Б

Листинг-4 LineNumberArea.h

```
#ifndef LINENUMBERAREA_H
#define LINENUMBERAREA_H
#include <QWidget>
#include <CodeEditor.h>
class LineNumberArea : public QWidget
 {
 public:
     LineNumberArea(CodeEditor *editor);

     QSize sizeHint() const;

 protected:

     void paintEvent(QPaintEvent *event);

 private:
     CodeEditor *codeEditor;
 };
#endif // LINENUMBERAREA_H
```

Листинг-5 LineNumberArea.cpp

```
#include "LineNumberArea.h"
LineNumberArea::LineNumberArea(CodeEditor *editor) : QWidget(editor) {
    codeEditor = editor;
}
QSize LineNumberArea::sizeHint() const {
    return QSize(codeEditor->lineNumberAreaWidth(), 0);
}
void LineNumberArea::paintEvent(QPaintEvent *event) {
    codeEditor->lineNumberAreaPaintEvent(event);
}
```

Приложение В

Листинг-6 DialogAboutMe.h

```
#ifndef DIALOGABOUTME_H
#define DIALOGABOUTME_H
#include "mainwindow.h"
#include "QMenuBar"
#include "QMenu"
#include "QDialog"
#include "QPushButton"
#include "QLineEdit"
#include "QBoxLayout"
#include "QLabel"
#include "QListWidget"
class DialogAboutMe : public QDialog {

    Q_OBJECT

public:
    DialogAboutMe( QWidget* parent = 0 );
    ~DialogAboutMe();

private:
    QBoxLayout* layout;
    QLabel *imageLabel;
    QListWidget *lw;
    QPushButton *accept;
};


#endif // DIALOGABOUTME_H
```

Листинг-7 DialogAboutMe.cpp

```
#include "DialogAboutMe.h"
#include "QMenuBar"
#include "QMenu"
#include "QDialog"
#include "QPushButton"
#include "QLineEdit"
#include "QBoxLayout"
#include "QLabel"
#include "QFrame"
#include "QListWidget"
#include "QString"
#include <QGridLayout>
DialogAboutMe::DialogAboutMe( QWidget* parent ) : QDialog(parent) {
    setWindowTitle("Info");
    setWindowIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\help.jpg"));
    layout = new QHBoxLayout;
    QGridLayout *l=new QGridLayout;
```

```
    imageLabel=new QLabel;
    QPixmap myPixmap=QPixmap("C:\\Users\\The Witcher
Hunt\\Documents\\build-labaWORD-Desktop_x86_windows_msvc2019_pe_64bit-
Debug\\image.png");
    imageLabel->setPixmap( myPixmap );
    layout->addWidget(imageLabel);

    lw = new QListWidget(this);

        lw->addItem("Author:");
        lw->addItem("Ushakov Aleksandr Nikolaevich SCS201");
        lw->addItem("Build date:");
        lw->addItem(__DATE__);
        lw->addItem(__TIME__);
        lw->addItem("The version of Qt that I was going to use:");
        lw->addItem(QT_VERSION_STR);
        lw->addItem("The version of Qt that is running:");
        lw->addItem(qVersion());

    accept=new QPushButton("OK", this);
    connect(accept, SIGNAL(clicked()), this, SLOT(accept()));
    layout->addWidget(lw);
    l->addLayout(layout,0,0);
    l->addWidget(accept,1, 0);
    setLayout(l);

}
DialogAboutMe::~DialogAboutMe() {
}
```

Приложение Г

Листинг-8 ReplaceSearch.h

```
#ifndef REPLACESEARCH_H
#define REPLACESEARCH_H
#include <QDialog>
#include <QLineEdit>
class ReplaceSearch: public QDialog{

    Q_OBJECT

public:

    ReplaceSearch();
    QString line();
    QString replace_area();

private:
```

```
    QLineEdit* Word;
    QLineEdit* replaceWord;
    QPushButton *ok;

private slots:

   void Replace();

signals:

    void SignalofReplace();
};

#endif // REPLACESEARCH_H
```

Листинг-9 ReplaceSearch.cpp

```
#include "ReplaceSearch.h"
#include "QPushButton"
#include "QGridLayout"
#include "QLabel"
ReplaceSearch::ReplaceSearch(){
    setWindowTitle("Search word and replace");
    setWindowIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\find3.jpg"));
    Word = new QLineEdit;
    replaceWord = new QLineEdit;
    ok = new QPushButton("Replace");

    QLabel *a=new QLabel("Find:");
    QLabel *b=new QLabel("Replace on:");

    QGridLayout* layout = new QGridLayout;
    layout->addWidget(Word,1,0);
    layout->addWidget(a,0,0);
    layout->addWidget(replaceWord,3,0);
    layout->addWidget(b,2,0);
    layout->addWidget(ok,4,0);
    setLayout(layout);

    connect(ok, SIGNAL(clicked()),SLOT(Replace()));
}
QString ReplaceSearch::line(){
    return Word->text();
}

QString ReplaceSearch::replace_area(){
    return replaceWord->text();
}

void ReplaceSearch::Replace(){
```

```
    emit SignalofReplace();
}
```

Приложение Д

Листинг-10 Search.h

```
#ifndef SEARCH_H
#define SEARCH_H
#include <QDialog>
#include <QLineEdit>
class Search : public QDialog{

    Q_OBJECT

public:
    Search();
    QString Line();


private:
    QLineEdit *word;
    QPushButton *ok;

};
#endif // SEARCH_H
```

Листинг-11 Search.cpp

```
#include "Search.h"
#include <QPushButton>
#include <QGridLayout>
#include <QLabel>
Search::Search(): QDialog(0)
{
    setWindowTitle("Search word");
    setWindowIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\find3.jpg"));
    QLabel *l=new QLabel("Find:");
    word = new QLineEdit;
    ok = new QPushButton("OK");
    QGridLayout* layout = new QGridLayout;
    layout->addWidget(l);
    layout->addWidget(word,1,0);
    layout->addWidget(ok,2,0);
    setLayout(layout);
    connect(ok, SIGNAL(clicked()),SLOT(accept()));

}
QString Search::Line(){
```

```
    return word->text();
}
```

Приложение Е

Листинг-12 Syntax.h

```
#ifndef SEARCH_H
#define SEARCH_H
#include <QDialog>
#include <QLineEdit>
class Search : public QDialog{

    Q_OBJECT

public:
    Search();
    QString Line();


private:
    QLineEdit *word;
    QPushButton *ok;

};
#endif // SEARCH_H
```

Листинг-13 Syntax.cpp

```
#include "Syntax.h"
#include <QFile>
Syntax::Syntax(): QDialog(0){
    setWindowIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\changesw.png"));
    setWindowTitle("Change highlight syntax");
    change = new QPushButton("Change");
    download = new QPushButton("Download");
    default_ = new QPushButton("Default");
    connect(change,SIGNAL(clicked()),this,SLOT(accept()));
    connect(default_,SIGNAL(clicked()),this,SLOT(Default()));
    connect(download,SIGNAL(clicked()),this,SLOT(Download()));
    keywords = new QLabel("Keywords");
    functions = new QLabel("Functions");
    classFormat = new QLabel("Classes");
    key = new QLineEdit;
    func = new QLineEdit;
    class_ = new QLineEdit;
    QGridLayout* l = new QGridLayout;
    l -> addWidget(key,0,1);
```

```
    l -> addWidget(func,1,1);
    l -> addWidget(class_,2,1);
    l -> addWidget(keywords,0,0);
    l -> addWidget(functions,1,0);
    l -> addWidget(classFormat,2,0);
    l ->addWidget(default_,3,0);
    l ->addWidget(change,3,1);
    l ->addWidget(download,3,2);
    bool a=1;
    for (int i=1;a;i++){
         QString b="Style"+QString::number(i)+".txt";
         QFile f(b);
        if (f.open(QIODevice::ReadOnly))
        {
          QPushButton *t=new QPushButton("Style"+QString::number(i));
          t->setObjectName("Style"+QString::number(i));
          connect(t,SIGNAL(clicked()),this,SLOT(Download1()));
          l->addWidget(t);
        }
        else a=0;
    }

    setLayout(l);
}

void Syntax::Default(){
    key->setText("#ff00ff");
    func->setText("#0000ff");
    class_->setText("#800080");
    accept();
}
void Syntax::Download1(){

    QString a=QString(QObject::sender()->objectName());
    a+=".txt";
    name=a;

    QFile f(a);
    f.open(QIODevice::ReadOnly);

    QByteArray data = f.readAll();
    QString s = QString::fromUtf8(data);
    QString key_="",func_="",classs="";
    for (int i=0;i<s.size()/3;i++){
        key_.append(s[i]);
    }
   for (int i=7;i<14;i++){
        func_.append(s[i]);
    }
    for (int i=14;i<21;i++){
       classs.append(s[i]);
```

```
    }
    key->setText(key_);
    func->setText(func_);
    class_->setText(classs);

    accept();
}
void Syntax::Download(){
    QString a=QFileDialog::getOpenFileName();
    name=a;

    QFile f(a);
    f.open(QIODevice::ReadOnly);

    QByteArray data = f.readAll();
    QString s = QString::fromUtf8(data);
    QString key_="",func_="",classs="";
    for (int i=0;i<s.size()/3;i++){
        key_.append(s[i]);
    }
   for (int i=7;i<14;i++){
        func_.append(s[i]);
    }
    for (int i=14;i<21;i++){
       classs.append(s[i]);
    }
    key->setText(key_);
    func->setText(func_);
    class_->setText(classs);

    accept();
}
```

Приложение Ё

Листинг-14 highlighter.h

```
#ifndef HIGHLIGHTER_H
#define HIGHLIGHTER_H
#include <QMainWindow>
#include <QTextEdit>
#include <QToolBar>
#include <QMenu>
#include <QMenuBar>
#include <QFontDialog>
#include <QBoxLayout>
#include <QHBoxLayout>
#include <QPushButton>
#include <QColorDialog>
#include <QFileDialog>
```

```
#include <QTextCodec>
#include <QMessageBox>
#include <QTextDocumentWriter>
#include <QTextStream>
#include <QStatusBar>
#include <QLabel>
#include <QRegExp>
#include <QTextDocument>
#include <QString>
#include <cmath>
#include <QDateTime>
#include<QDebug>
#include <string>
#include <QTextCursor>
#include <QToolButton>
#include <QSyntaxHighlighter>

class Highlighter : public QSyntaxHighlighter
{
    Q_OBJECT

public:

    Highlighter(QTextDocument *parent = 0, int n = 0);
    Highlighter(QTextDocument *parent, int n, QColor a,QColor b,QColor
c);
    int number ;

    QBrush getKeywordColor();
    QBrush getClassColor();
    QBrush getQuotationColor();
    QBrush getFunctionColor();
    QBrush getMultiLineColor();


    void setKeywordColor( QBrush);
    void setClassColor( QBrush);
    void setQuotationColor( QBrush);
    void setFunctionColor( QBrush);
    void setMultiLineColor( QBrush);
    ~Highlighter();

protected:
    void highlightBlock(const QString &text);

private:
    struct HighlightingRule
    {
        QRegExp pattern;
        QTextCharFormat format;
    };
```

```
    QVector<HighlightingRule> highlightingRules;
    QRegExp commentStartExpression;
    QRegExp commentEndExpression;
    QTextCharFormat keywordFormat;
    QTextCharFormat classFormat;
    QTextCharFormat singleLineCommentFormat;
    QTextCharFormat multiLineCommentFormat;
    QTextCharFormat quotationFormat;
    QTextCharFormat functionFormat;

    QBrush colorKeyword = Qt::magenta;
    QBrush colorClassFormat = Qt::darkMagenta;
    QBrush colorQuotationFormat = Qt::darkGreen;
    QBrush colorFunctionFormat = Qt::blue;
    QBrush colorMultiLineFormat = Qt::red;

};
#endif // HIGHLIGHTER_H
```

Листинг-15 highlighter.cpp

```
#include "highlighter.h"
Highlighter::~Highlighter(){}
Highlighter::Highlighter(QTextDocument *parent, int n):
QSyntaxHighlighter(parent)
{
    HighlightingRule rule;

    keywordFormat.setForeground(colorKeyword);
    keywordFormat.setFontWeight(QFont::Bold);
    QStringList keywordPatterns11;
    QStringList keywordPatterns18;
    QStringList keywordPatterns14;
    QStringList keywordPatterns17;
    QStringList keywordPatterns20;
    number = n;
    keywordPatterns11 <<"\\auto\\b" << "\\bbreak\\b" << "\\bcase\\b"
            << "\\bchar\\b" << "\\bconst\\b" << "\\bcontinue\\b"
            << "\\bdefault\\b" << "\\bdo\\b" << "\\bdouble\\b"
            << "\\belse\\b" << "\\benum\\b" << "\\bextern\\b"
            << "\\bfloat\\b" << "\\bfor\\b" << "\\bgoto\\b"
            << "\\bif\\b" << "\\binline\\b" << "\\bint\\b"
            << "\\blong\\b" << "\\bregister\\b" << "\\brestrict\\b"
            << "\\breturn\\b" << "\\bshort\\b" << "\\bsigned\\b"
            << "\\bsizeof\\b" << "\\bstatic\\b" << "\\bstruct\\b"
            << "\\bswitch\\b" << "\\btypedef\\b" << "\\bunion\\b"
            << "\\bunsigned\\b" << "\\bvoid\\b" << "\\bvolatile\\b"
            << "\\bwhile\\b" << "\\b_Alignas\\b" << "\\b_Alignof\\b"
            << "\\b_Atomic\\b" << "\\b_Bool\\b" << "\\b_Complex\\b"
            << "\\b_Generic\\b" << "\\b_Imaginary\\b" <<
"\\b_Noreturn\\b"
```

```
            << "\\b_Static_assert\\b" << "\\_Thread_local\\b";
    keywordPatterns18
            <<"\\auto\\b" << "\\bbreak\\b" << "\\bcase\\b"
            << "\\bchar\\b" << "\\bconst\\b" << "\\bcontinue\\b"
            <<"\\bdefault\\b" << "\\bdo\\b" << "\\bdouble\\b"
            << "\\belse\\b" << "\\benum\\b" << "\\bextern\\b"
            << "\\bfloat\\b" << "\\bfor\\b" << "\\bgoto\\b"
            << "\\bif\\b" << "\\binline\\b" << "\\bint\\b"
            << "\\blong\\b" << "\\bregister\\b" << "\\brestrict\\b"
            << "\\breturn\\b" << "\\bshort\\b" << "\\bsigned\\b"
            << "\\bsizeof\\b" << "\\bstatic\\b" << "\\bstruct\\b"
            << "\\bswitch\\b" << "\\btypedef\\b" << "\\bunion\\b"
            << "\\bunsigned\\b" << "\\bvoid\\b" << "\\bvolatile\\b"
            << "\\bwhile\\b" << "\\b_Alignas\\b" << "\\b_Alignof\\b"
            <<"\\b_Atomic\\b" << "\\b_Bool\\b" << "\\b_Complex\\b"
            << "\\b_Generic\\b" << "\\b_Imaginary\\b" <<
"\\b_Noreturn\\b"
            << "\\b_Static_assert\\b" << "\\_Thread_local\\b";
    keywordPatterns14
            << "\\balignas\\b" << "\\balignof\\b"
            << "\\bchar16_t\\b" << "\\bchar32_t\\b" <<
"\\bconstexpr\\b"
            << "\\bdecltype\\b" << "\\bnoexcept\\b" << "\\bnullptr\\b"
            << "\\bstatic_assert\\b" << "\\bthread_local\\b"
            << "\\band\\b" << "\\band_eq\\b" << "\\basm\\b"
            << "\\bauto\\b" << "\\bbitand\\b" << "\\bbitor\\b"
            << "\\bbool\\b" << "\\bbreak\\b" << "\\bcase\\b"
            << "\\bcatch\\b" << "\\bchar\\b" << "\\bclass\\b"
            << "\\bcompl\\b" << "\\bconst\\b" << "\\bconst_cast\\b"
            << "\\bcontinue\\b" << "\\bdefault\\b" << "\\bdelete\\b"
            << "\\bdo\\b" << "\\bdouble\\b" << "\\bdynamic_cast\\b"
            << "\\belse\\b" << "\\benum\\b" << "\\bexplicit\\b"
            << "\\bexport\\b" << "\\bextern\\b" << "\\bfalse\\b"
            << "\\bfloat\\b" << "\\bfor\\b"  << "\\bfriend\\b"
            << "\\goto\\b" << "\\bif\\b" << "\\binline\\b"
            << "\\bint\\b" << "\\blong\\b" << "\\bmutable\\b"
            << "\\bnamespace\\b" << "\\bnew\\b" << "\\bnot\\b"
            << "\\bnot_eq\\b" << "\\boperator\\b" << "\\bor\\b"
            << "\\bor_eq\\b" << "\\bprivate\\b" << "\\bprotected\\b"
            << "\\bpublic\\b" << "\\bregister\\b" <<
"\\breinterpret_cast\\b"
            << "\\breturn\\b" << "\\bshort\\b" << "\\bsigned\\b"
            << "\\bsizeof\\b" << "\\bstatic\\b" << "\\bstatic_cast\\b"
            << "\\struct\\b" << "\\bswitch\\b" << "\\btemplate\\b"
            << "\\this\\b" << "\\bthrow\\b" << "\\btrue\\b"
            << "\\btry\\b" << "\\btypedef\\b" << "\\btypeid\\b"
            << "\\bunion\\b" << "\\unsigned\\b" << "\\busing\\b"
            << "\\bvirtual\\b" << "\\bvoid\\b" << "\\bvolatile\\b"
            << "\\bwchar_t\\b" << "\\bwhile\\b" << "\\bxor\\b"
            << "\\bxor_eq\\b" ;
    keywordPatterns17
```

```
            << "\\balignas\\b" << "\\balignof\\b"
            << "\\bchar16_t\\b" << "\\bchar32_t\\b" <<
"\\bconstexpr\\b"
            << "\\bdecltype\\b" << "\\bnoexcept\\b" << "\\bnullptr\\b"
            << "\\bstatic_assert\\b" << "\\bthread_local\\b"
            << "\\band\\b" << "\\band_eq\\b" << "\\basm\\b"
            << "\\bauto\\b" << "\\bbitand\\b" << "\\bbitor\\b"
            << "\\bbool\\b" << "\\bbreak\\b" << "\\bcase\\b"
            << "\\bcatch\\b" << "\\bchar\\b" << "\\bclass\\b"
            << "\\bcompl\\b" << "\\bconst\\b" << "\\bconst_cast\\b"
            << "\\bcontinue\\b" << "\\bdefault\\b" << "\\bdelete\\b"
            << "\\bdo\\b" << "\\bdouble\\b" << "\\bdynamic_cast\\b"
            << "\\belse\\b" << "\\benum\\b" << "\\bexplicit\\b"
            << "\\bexport\\b" << "\\bextern\\b" << "\\bfalse\\b"
            << "\\bfloat\\b" << "\\bfor\\b"  << "\\bfriend\\b"
            << "\\goto\\b" << "\\bif\\b" << "\\binline\\b"
            << "\\bint\\b" << "\\blong\\b" << "\\bmutable\\b"
            << "\\bnamespace\\b" << "\\bnew\\b" << "\\bnot\\b"
            << "\\bnot_eq\\b" << "\\boperator\\b" << "\\bor\\b"
            << "\\bor_eq\\b" << "\\bprivate\\b" << "\\bprotected\\b"
            << "\\bpublic\\b" << "\\bregister\\b" <<
"\\breinterpret_cast\\b"
            << "\\breturn\\b" << "\\bshort\\b" << "\\bsigned\\b"
            << "\\bsizeof\\b" << "\\bstatic\\b" << "\\bstatic_cast\\b"
            << "\\struct\\b" << "\\bswitch\\b" << "\\btemplate\\b"
            << "\\this\\b" << "\\bthrow\\b" << "\\btrue\\b"
            << "\\btry\\b" << "\\btypedef\\b" << "\\btypeid\\b"
            << "\\bunion\\b" << "\\unsigned\\b" << "\\busing\\b"
            << "\\bvirtual\\b" << "\\bvoid\\b" << "\\bvolatile\\b"
            << "\\bwchar_t\\b" << "\\bwhile\\b" << "\\bxor\\b"
            << "\\bxor_eq\\b" ;
    keywordPatterns20
            << "\\bchar8_t\\b" << "\\bconcept\\b"
            << "\\bconsteval\\b" << "\\bconstinit\\b" <<
"\\bco_await\\b"
            << "\\bco_return\\b" << "\\bco_yield\\b" <<
"\\brequires\\b"
            << "\\balignas\\b" << "\\balignof\\b"
            << "\\bchar16_t\\b" << "\\bchar32_t\\b" <<
"\\bconstexpr\\b"
            << "\\bdecltype\\b" << "\\bnoexcept\\b" << "\\bnullptr\\b"
            << "\\bstatic_assert\\b" << "\\bthread_local\\b"
            << "\\band\\b" << "\\band_eq\\b" << "\\basm\\b"
            << "\\bauto\\b" << "\\bbitand\\b" << "\\bbitor\\b"
            << "\\bbool\\b" << "\\bbreak\\b" << "\\bcase\\b"
            << "\\bcatch\\b" << "\\bchar\\b" << "\\bclass\\b"
            << "\\bcompl\\b" << "\\bconst\\b" << "\\bconst_cast\\b"
            << "\\bcontinue\\b" << "\\bdefault\\b" << "\\bdelete\\b"
            << "\\bdo\\b" << "\\bdouble\\b" << "\\bdynamic_cast\\b"
            << "\\belse\\b" << "\\benum\\b" << "\\bexplicit\\b"
            << "\\bexport\\b" << "\\bextern\\b" << "\\bfalse\\b"
```

```
            << "\\bfloat\\b" << "\\bfor\\b"  << "\\bfriend\\b"
            << "\\goto\\b" << "\\bif\\b" << "\\binline\\b"
            << "\\bint\\b" << "\\blong\\b" << "\\bmutable\\b"
            << "\\bnamespace\\b" << "\\bnew\\b" << "\\bnot\\b"
            << "\\bnot_eq\\b" << "\\boperator\\b" << "\\bor\\b"
            << "\\bor_eq\\b" << "\\bprivate\\b" << "\\bprotected\\b"
            << "\\bpublic\\b" << "\\bregister\\b" <<
"\\breinterpret_cast\\b"
            << "\\breturn\\b" << "\\bshort\\b" << "\\bsigned\\b"
            << "\\bsizeof\\b" << "\\bstatic\\b" << "\\bstatic_cast\\b"
            << "\\struct\\b" << "\\bswitch\\b" << "\\btemplate\\b"
            << "\\this\\b" << "\\bthrow\\b" << "\\btrue\\b"
            << "\\btry\\b" << "\\btypedef\\b" << "\\btypeid\\b"
            << "\\bunion\\b" << "\\unsigned\\b" << "\\busing\\b"
            << "\\bvirtual\\b" << "\\bvoid\\b" << "\\bvolatile\\b"
            << "\\bwchar_t\\b" << "\\bwhile\\b" << "\\bxor\\b"
            << "\\bxor_eq\\b" ;

    if (number == 1){
        foreach (const QString &pattern, keywordPatterns11) {
            rule.pattern = QRegExp(pattern);
            rule.format = keywordFormat;
            highlightingRules.append(rule);
        }

    }
    else if (number == 2){
        foreach (const QString &pattern, keywordPatterns18) {
            rule.pattern = QRegExp(pattern);
            rule.format = keywordFormat;
            highlightingRules.append(rule);
        }

    }
    else if(number == 3){
        foreach (const QString &pattern, keywordPatterns14) {
            rule.pattern = QRegExp(pattern);
            rule.format = keywordFormat;
            highlightingRules.append(rule);
        }

    }
    else if(number == 4){
        foreach (const QString &pattern, keywordPatterns17) {
            rule.pattern = QRegExp(pattern);
            rule.format = keywordFormat;
            highlightingRules.append(rule);
        }

    }
    else if(number == 5){
```

```
        foreach (const QString &pattern, keywordPatterns20) {
            rule.pattern = QRegExp(pattern);
            rule.format = keywordFormat;
            highlightingRules.append(rule);
        }

    }
    classFormat.setFontWeight(QFont::Bold);
    classFormat.setForeground(colorClassFormat);
    rule.pattern = QRegExp("\\bQ[A-Za-z]+\\b");
    rule.format = classFormat;
    highlightingRules.append(rule);

    quotationFormat.setForeground(colorQuotationFormat);
    rule.pattern = QRegExp("\".*\"");
    rule.format = quotationFormat;
    highlightingRules.append(rule);

    functionFormat.setFontItalic(true);
    functionFormat.setForeground(colorFunctionFormat);
    rule.pattern = QRegExp("\\b[A-Za-z0-9_]+(?=\\()");
    rule.format = functionFormat;
    highlightingRules.append(rule);
    singleLineCommentFormat.setForeground(Qt::red);
    rule.pattern = QRegExp("//[^\n]*");
    rule.format = singleLineCommentFormat;
    highlightingRules.append(rule);

    multiLineCommentFormat.setForeground(colorMultiLineFormat);

    commentStartExpression = QRegExp("/\\*");
    commentEndExpression = QRegExp("\\*/");
}
Highlighter::Highlighter(QTextDocument *parent, int n, QColor a,QColor
b,QColor c): QSyntaxHighlighter(parent)
{
    colorKeyword=a;colorClassFormat=b;colorFunctionFormat=c;
    HighlightingRule rule;
    keywordFormat.setForeground(a);
    keywordFormat.setFontWeight(QFont::Bold);
    QStringList keywordPatterns11;
    QStringList keywordPatterns18;
    QStringList keywordPatterns14;
    QStringList keywordPatterns17;
    QStringList keywordPatterns20;
    number = n;
    keywordPatterns11 <<"\\auto\\b" << "\\bbreak\\b" << "\\bcase\\b"
            << "\\bchar\\b" << "\\bconst\\b" << "\\bcontinue\\b"
            << "\\bdefault\\b" << "\\bdo\\b" << "\\bdouble\\b"
            << "\\belse\\b" << "\\benum\\b" << "\\bextern\\b"
            << "\\bfloat\\b" << "\\bfor\\b" << "\\bgoto\\b"
```

```
            << "\\bif\\b" << "\\binline\\b" << "\\bint\\b"
            << "\\blong\\b" << "\\bregister\\b" << "\\brestrict\\b"
            << "\\breturn\\b" << "\\bshort\\b" << "\\bsigned\\b"
            << "\\bsizeof\\b" << "\\bstatic\\b" << "\\bstruct\\b"
            << "\\bswitch\\b" << "\\btypedef\\b" << "\\bunion\\b"
            << "\\bunsigned\\b" << "\\bvoid\\b" << "\\bvolatile\\b"
            << "\\bwhile\\b" << "\\b_Alignas\\b" << "\\b_Alignof\\b"
            << "\\b_Atomic\\b" << "\\b_Bool\\b" << "\\b_Complex\\b"
            << "\\b_Generic\\b" << "\\b_Imaginary\\b" <<
"\\b_Noreturn\\b"
            << "\\b_Static_assert\\b" << "\\_Thread_local\\b";
    keywordPatterns18
            <<"\\auto\\b" << "\\bbreak\\b" << "\\bcase\\b"
            << "\\bchar\\b" << "\\bconst\\b" << "\\bcontinue\\b"
            <<"\\bdefault\\b" << "\\bdo\\b" << "\\bdouble\\b"
            << "\\belse\\b" << "\\benum\\b" << "\\bextern\\b"
            << "\\bfloat\\b" << "\\bfor\\b" << "\\bgoto\\b"
            << "\\bif\\b" << "\\binline\\b" << "\\bint\\b"
            << "\\blong\\b" << "\\bregister\\b" << "\\brestrict\\b"
            << "\\breturn\\b" << "\\bshort\\b" << "\\bsigned\\b"
            << "\\bsizeof\\b" << "\\bstatic\\b" << "\\bstruct\\b"
            << "\\bswitch\\b" << "\\btypedef\\b" << "\\bunion\\b"
            << "\\bunsigned\\b" << "\\bvoid\\b" << "\\bvolatile\\b"
            << "\\bwhile\\b" << "\\b_Alignas\\b" << "\\b_Alignof\\b"
            <<"\\b_Atomic\\b" << "\\b_Bool\\b" << "\\b_Complex\\b"
            << "\\b_Generic\\b" << "\\b_Imaginary\\b" <<
"\\b_Noreturn\\b"
            << "\\b_Static_assert\\b" << "\\_Thread_local\\b";
    keywordPatterns14
            << "\\balignas\\b" << "\\balignof\\b"
            << "\\bchar16_t\\b" << "\\bchar32_t\\b" <<
"\\bconstexpr\\b"
            << "\\bdecltype\\b" << "\\bnoexcept\\b" << "\\bnullptr\\b"
            << "\\bstatic_assert\\b" << "\\bthread_local\\b"
            << "\\band\\b" << "\\band_eq\\b" << "\\basm\\b"
            << "\\bauto\\b" << "\\bbitand\\b" << "\\bbitor\\b"
            << "\\bbool\\b" << "\\bbreak\\b" << "\\bcase\\b"
            << "\\bcatch\\b" << "\\bchar\\b" << "\\bclass\\b"
            << "\\bcompl\\b" << "\\bconst\\b" << "\\bconst_cast\\b"
            << "\\bcontinue\\b" << "\\bdefault\\b" << "\\bdelete\\b"
            << "\\bdo\\b" << "\\bdouble\\b" << "\\bdynamic_cast\\b"
            << "\\belse\\b" << "\\benum\\b" << "\\bexplicit\\b"
            << "\\bexport\\b" << "\\bextern\\b" << "\\bfalse\\b"
            << "\\bfloat\\b" << "\\bfor\\b"  << "\\bfriend\\b"
            << "\\goto\\b" << "\\bif\\b" << "\\binline\\b"
            << "\\bint\\b" << "\\blong\\b" << "\\bmutable\\b"
            << "\\bnamespace\\b" << "\\bnew\\b" << "\\bnot\\b"
            << "\\bnot_eq\\b" << "\\boperator\\b" << "\\bor\\b"
            << "\\bor_eq\\b" << "\\bprivate\\b" << "\\bprotected\\b"
            << "\\bpublic\\b" << "\\bregister\\b" <<
"\\breinterpret_cast\\b"
```

```
            << "\\breturn\\b" << "\\bshort\\b" << "\\bsigned\\b"
            << "\\bsizeof\\b" << "\\bstatic\\b" << "\\bstatic_cast\\b"
            << "\\struct\\b" << "\\bswitch\\b" << "\\btemplate\\b"
            << "\\this\\b" << "\\bthrow\\b" << "\\btrue\\b"
            << "\\btry\\b" << "\\btypedef\\b" << "\\btypeid\\b"
            << "\\bunion\\b" << "\\unsigned\\b" << "\\busing\\b"
            << "\\bvirtual\\b" << "\\bvoid\\b" << "\\bvolatile\\b"
            << "\\bwchar_t\\b" << "\\bwhile\\b" << "\\bxor\\b"
            << "\\bxor_eq\\b" ;
    keywordPatterns17
            << "\\balignas\\b" << "\\balignof\\b"
            << "\\bchar16_t\\b" << "\\bchar32_t\\b" <<
"\\bconstexpr\\b"
            << "\\bdecltype\\b" << "\\bnoexcept\\b" << "\\bnullptr\\b"
            << "\\bstatic_assert\\b" << "\\bthread_local\\b"
            << "\\band\\b" << "\\band_eq\\b" << "\\basm\\b"
            << "\\bauto\\b" << "\\bbitand\\b" << "\\bbitor\\b"
            << "\\bbool\\b" << "\\bbreak\\b" << "\\bcase\\b"
            << "\\bcatch\\b" << "\\bchar\\b" << "\\bclass\\b"
            << "\\bcompl\\b" << "\\bconst\\b" << "\\bconst_cast\\b"
            << "\\bcontinue\\b" << "\\bdefault\\b" << "\\bdelete\\b"
            << "\\bdo\\b" << "\\bdouble\\b" << "\\bdynamic_cast\\b"
            << "\\belse\\b" << "\\benum\\b" << "\\bexplicit\\b"
            << "\\bexport\\b" << "\\bextern\\b" << "\\bfalse\\b"
            << "\\bfloat\\b" << "\\bfor\\b"  << "\\bfriend\\b"
            << "\\goto\\b" << "\\bif\\b" << "\\binline\\b"
            << "\\bint\\b" << "\\blong\\b" << "\\bmutable\\b"
            << "\\bnamespace\\b" << "\\bnew\\b" << "\\bnot\\b"
            << "\\bnot_eq\\b" << "\\boperator\\b" << "\\bor\\b"
            << "\\bor_eq\\b" << "\\bprivate\\b" << "\\bprotected\\b"
            << "\\bpublic\\b" << "\\bregister\\b" <<
"\\breinterpret_cast\\b"
            << "\\breturn\\b" << "\\bshort\\b" << "\\bsigned\\b"
            << "\\bsizeof\\b" << "\\bstatic\\b" << "\\bstatic_cast\\b"
            << "\\struct\\b" << "\\bswitch\\b" << "\\btemplate\\b"
            << "\\this\\b" << "\\bthrow\\b" << "\\btrue\\b"
            << "\\btry\\b" << "\\btypedef\\b" << "\\btypeid\\b"
            << "\\bunion\\b" << "\\unsigned\\b" << "\\busing\\b"
            << "\\bvirtual\\b" << "\\bvoid\\b" << "\\bvolatile\\b"
            << "\\bwchar_t\\b" << "\\bwhile\\b" << "\\bxor\\b"
            << "\\bxor_eq\\b" ;
    keywordPatterns20
            << "\\bchar8_t\\b" << "\\bconcept\\b"
            << "\\bconsteval\\b" << "\\bconstinit\\b" <<
"\\bco_await\\b"
            << "\\bco_return\\b" << "\\bco_yield\\b" <<
"\\brequires\\b"
            << "\\balignas\\b" << "\\balignof\\b"
            << "\\bchar16_t\\b" << "\\bchar32_t\\b" <<
"\\bconstexpr\\b"
            << "\\bdecltype\\b" << "\\bnoexcept\\b" << "\\bnullptr\\b"
```

```
            << "\\bstatic_assert\\b" << "\\bthread_local\\b"
            << "\\band\\b" << "\\band_eq\\b" << "\\basm\\b"
            << "\\bauto\\b" << "\\bbitand\\b" << "\\bbitor\\b"
            << "\\bbool\\b" << "\\bbreak\\b" << "\\bcase\\b"
            << "\\bcatch\\b" << "\\bchar\\b" << "\\bclass\\b"
            << "\\bcompl\\b" << "\\bconst\\b" << "\\bconst_cast\\b"
            << "\\bcontinue\\b" << "\\bdefault\\b" << "\\bdelete\\b"
            << "\\bdo\\b" << "\\bdouble\\b" << "\\bdynamic_cast\\b"
            << "\\belse\\b" << "\\benum\\b" << "\\bexplicit\\b"
            << "\\bexport\\b" << "\\bextern\\b" << "\\bfalse\\b"
            << "\\bfloat\\b" << "\\bfor\\b"  << "\\bfriend\\b"
            << "\\goto\\b" << "\\bif\\b" << "\\binline\\b"
            << "\\bint\\b" << "\\blong\\b" << "\\bmutable\\b"
            << "\\bnamespace\\b" << "\\bnew\\b" << "\\bnot\\b"
            << "\\bnot_eq\\b" << "\\boperator\\b" << "\\bor\\b"
            << "\\bor_eq\\b" << "\\bprivate\\b" << "\\bprotected\\b"
            << "\\bpublic\\b" << "\\bregister\\b" <<
"\\breinterpret_cast\\b"
            << "\\breturn\\b" << "\\bshort\\b" << "\\bsigned\\b"
            << "\\bsizeof\\b" << "\\bstatic\\b" << "\\bstatic_cast\\b"
            << "\\struct\\b" << "\\bswitch\\b" << "\\btemplate\\b"
            << "\\this\\b" << "\\bthrow\\b" << "\\btrue\\b"
            << "\\btry\\b" << "\\btypedef\\b" << "\\btypeid\\b"
            << "\\bunion\\b" << "\\unsigned\\b" << "\\busing\\b"
            << "\\bvirtual\\b" << "\\bvoid\\b" << "\\bvolatile\\b"
            << "\\bwchar_t\\b" << "\\bwhile\\b" << "\\bxor\\b"
            << "\\bxor_eq\\b" ;

    if (number == 1){
        foreach (const QString &pattern, keywordPatterns11) {
            rule.pattern = QRegExp(pattern);
            rule.format = keywordFormat;
            highlightingRules.append(rule);
        }

    }
    else if (number == 2){
        foreach (const QString &pattern, keywordPatterns18) {
            rule.pattern = QRegExp(pattern);
            rule.format = keywordFormat;
            highlightingRules.append(rule);
        }

    }
    else if(number == 3){
        foreach (const QString &pattern, keywordPatterns14) {
            rule.pattern = QRegExp(pattern);
            rule.format = keywordFormat;
            highlightingRules.append(rule);
        }
```

```
    }
    else if(number == 4){
        foreach (const QString &pattern, keywordPatterns17) {
            rule.pattern = QRegExp(pattern);
            rule.format = keywordFormat;
            highlightingRules.append(rule);
        }

    }
    else if(number == 5){
        foreach (const QString &pattern, keywordPatterns20) {
            rule.pattern = QRegExp(pattern);
            rule.format = keywordFormat;
            highlightingRules.append(rule);
        }

    }
    classFormat.setFontWeight(QFont::Bold);
    classFormat.setForeground(colorClassFormat);
    rule.pattern = QRegExp("\\bQ[A-Za-z]+\\b");
    rule.format = classFormat;
    highlightingRules.append(rule);

    quotationFormat.setForeground(colorQuotationFormat);
    rule.pattern = QRegExp("\".*\"");
    rule.format = quotationFormat;
    highlightingRules.append(rule);

    functionFormat.setFontItalic(true);
    functionFormat.setForeground(colorFunctionFormat);
    rule.pattern = QRegExp("\\b[A-Za-z0-9_]+(?=\\()");
    rule.format = functionFormat;
    highlightingRules.append(rule);
    singleLineCommentFormat.setForeground(Qt::red);
    rule.pattern = QRegExp("//[^\n]*");
    rule.format = singleLineCommentFormat;
    highlightingRules.append(rule);

    multiLineCommentFormat.setForeground(colorMultiLineFormat);

    commentStartExpression = QRegExp("/\\*");
    commentEndExpression = QRegExp("\\*/");
}
void Highlighter::highlightBlock(const QString &text)
{
    foreach (const HighlightingRule &rule, highlightingRules) {
        QRegExp expression(rule.pattern);
        int index = expression.indexIn(text);
        while (index >= 0) {
            int length = expression.matchedLength();
            setFormat(index, length, rule.format);
```

```
            index = expression.indexIn(text, index + length);
        }
    }
    setCurrentBlockState(0);
    int startIndex = 0;
        if (previousBlockState() != 1)
            startIndex = commentStartExpression.indexIn(text);
    while (startIndex >= 0) {
            int endIndex = commentEndExpression.indexIn(text,
startIndex);
            int commentLength;
            if (endIndex == -1) {
                setCurrentBlockState(1);
                commentLength = text.length() - startIndex;
            } else {
                commentLength = endIndex - startIndex
                                + commentEndExpression.matchedLength();
            }
            setFormat(startIndex, commentLength,
multiLineCommentFormat);
            startIndex = commentStartExpression.indexIn(text,
startIndex + commentLength);
        }
}
QBrush Highlighter::getKeywordColor(){
    return colorKeyword;
}
QBrush Highlighter::getClassColor(){
    return colorClassFormat;
}
QBrush Highlighter::getQuotationColor(){
    return colorQuotationFormat;
}
QBrush Highlighter::getFunctionColor(){
    return colorFunctionFormat;
}
QBrush Highlighter::getMultiLineColor(){
    return colorMultiLineFormat;
}

void  Highlighter::setKeywordColor( QBrush a){
    colorKeyword = a;

}
void  Highlighter::setClassColor( QBrush a){
    colorClassFormat = a;
}
void  Highlighter::setQuotationColor( QBrush a){
    colorQuotationFormat = a;
}
void  Highlighter::setFunctionColor( QBrush a){
```

```
    colorFunctionFormat = a;
}
void  Highlighter::setMultiLineColor( QBrush a){
    colorMultiLineFormat = a;
}
```

Приложение Ж

Листинг-16 MainWindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <QMainWindow>
#include <QMenu>
#include <QLabel>
#include "CodeEditor.h"
#include "Search.h"
#include "highlighter.h"
#include "ReplaceSearch.h"
#include "Syntax.h"
#include <QToolBar>
#include <QSettings>

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    QString name="untitled";
    bool f;
    ~MainWindow();
protected:
    void mousePressEvent(QMouseEvent *event);
private slots:
    void exit();
    void showAboutProgram();
    void New_();
    void Open_();
    bool Save_();
    bool Save_As_();
    void setCurName(const QString &fileName);
    void RefreshStatusBar();
    void SetFontSelection();
    void HideStatusBar();
    void WrapWord();
    void HideNumberArea();
    void ChangeColorLine();
    void SetColorBackground();
    void ActiveSearch();
    void FindWord();
```

```
    void SearchandReplace();
    void replaceEND();
    void SelectWord();
    void SelectLine();
    void Delete();
    void HideToolBar();
    void RefreshZv();
    void HideHighlighter();
    void v11();
    void v18();
    void v14();
    void v17();
    void v20();
    void cSyntax();
private:
    void context();
    void tool();
    bool openDoc(const QString &a);
    QMenuBar *MainMenuBar;
    QMenu *MenuFile,*MenuEdit,*MenuFormat,*MenuView,*MenuHelp;
    QAction *New,*Open,*Save,*Save_as,*Exit;
    QAction
*Cancel,*Repeat,*Copy,*Cut,*Paste,*Find,*Find_and_replace,*Highlight;
    QAction *Perenos,*Font;
    QAction
*ColorBackground,*ColorCurString,*SwitcherNumeration,*SwitcherPanel,*S
witcherStatus,*SwitcherHightlighter,*Hightlighter,*cSyn;
    QAction *AboutProgram;
    QAction* C11;
    QAction* C18;
    QAction* Cpp14;
    QAction* C17;
    QAction* C20;
    QActionGroup* group;
    QLabel *len,* curPos,* last_change;
    CodeEditor *editor;
    Search *search;
    ReplaceSearch *rsearch;
    QString NameFile;
    QDateTime *dateTime;
    QDateTime *sdateTime;
    QToolBar *toolbar;
    Highlighter *highlighter;
    Syntax* syntax;
    QSettings* setting;


};
#endif // MAINWINDOW_H
```

Листинг-17 MainWindow.cpp

```
#include "mainwindow.h"
#include "Search.h"
#include "DialogAboutMe.h"
#include "QMenuBar"
#include "QMenu"
#include "QDialog"
#include "QPushButton"
#include "QLabel"
#include "CodeEditor.h"
#include <QStatusBar>
#include <QApplication>
#include <QFontDialog>
#include <QMessageBox>
#include <QClipboard>
#include "ReplaceSearch.h"
#include <QFileInfo>
#include <QFileDialog>
#include <QToolBar>
#include <QToolButton>
#include <highlighter.h>
#include <QActionGroup>
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
{
    setWindowIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\word.png"));
    setWindowTitle(tr("[*]%1 - %2").arg("untilted").arg("Laboratory
work 5"));
    editor=new CodeEditor(this);
    editor->name=this->name;
    search=new Search;
    rsearch=new ReplaceSearch;
    syntax=new Syntax;
    toolbar = addToolBar("main toolbar");
    setCentralWidget(editor);
    editor->setFocus();
    MenuFile = menuBar()->addMenu("&File");
    MenuEdit = menuBar()->addMenu("&Edit");
    MenuFormat=menuBar()->addMenu("&Format");
    MenuView=menuBar()->addMenu("&View");
    MenuHelp=menuBar()->addMenu("&Help");

    New=MenuFile->addAction("New");
    connect(New,SIGNAL(triggered()),this,SLOT(New_()));
    Open=MenuFile->addAction("Open");
    connect(Open,SIGNAL(triggered()),this,SLOT(Open_()));
    Save=MenuFile->addAction("Save");
    connect(Save,SIGNAL(triggered()),this,SLOT(Save_()));
    Save_as=MenuFile->addAction("Save as");
    connect(Save_as,SIGNAL(triggered()),this,SLOT(Save_As_()));
    Exit=MenuFile->addAction("&Exit");
```

```
    connect(Exit, SIGNAL(triggered()), this, SLOT(exit()));

    Cancel=MenuEdit->addAction("Undo");
    connect(Cancel, SIGNAL(triggered()), editor, SLOT(undo1()));
    Repeat=MenuEdit->addAction("Redo");
    connect(Repeat, SIGNAL(triggered()), editor, SLOT(redo1()));
    Copy=MenuEdit->addAction("Copy");
    connect(Copy, SIGNAL(triggered()), editor, SLOT(copy()));
    Cut=MenuEdit->addAction("Cut");
    connect(Cut, SIGNAL(triggered()), editor, SLOT(cut()));
    Paste=MenuEdit->addAction("Paste");
    connect(Paste, SIGNAL(triggered()), editor, SLOT(paste()));
    Find=MenuEdit->addAction("Find");
    connect(Find,SIGNAL(triggered()), this, SLOT(ActiveSearch()));
    Find_and_replace=MenuEdit->addAction("Find and Replace");

connect(Find_and_replace,SIGNAL(triggered()),this,SLOT(SearchandReplac
e()));
    Highlight=MenuEdit->addAction("Select all");
    connect(Highlight, SIGNAL(triggered()), editor,
SLOT(selectAll()));


    Perenos=MenuFormat->addAction("Word wrap");
    MenuFormat->actions().back()->setCheckable(true);
    Perenos->setChecked(true);
    connect(Perenos,SIGNAL(triggered()),this,SLOT(WrapWord()));
    Font=MenuFormat->addAction("Font Selection");
    connect(Font,SIGNAL(triggered()),this,SLOT(SetFontSelection()));


    ColorBackground=MenuView->addAction("Background color selection");

connect(ColorBackground,SIGNAL(triggered()),this,SLOT(SetColorBackgrou
nd()));
    ColorCurString=MenuView->addAction("Selecting the color of the
current line");

connect(ColorCurString,SIGNAL(triggered()),this,SLOT(ChangeColorLine()
));
    SwitcherNumeration=MenuView->addAction("On/Off line numbering
display");
    MenuView->actions().back()->setCheckable(true);
    SwitcherNumeration->setChecked(true);

connect(SwitcherNumeration,SIGNAL(triggered()),this,SLOT(HideNumberAre
a()));
    SwitcherPanel=MenuView->addAction("On/Off toolbar display");
    MenuView->actions().back()->setCheckable(true);
    SwitcherPanel->setChecked(true);
```

```
connect(SwitcherPanel,SIGNAL(triggered()),this,SLOT(HideToolBar()));
    SwitcherStatus=MenuView->addAction("On/Off status bar display");
    MenuView->actions().back()->setCheckable(true);

connect(SwitcherStatus,SIGNAL(triggered()),this,SLOT(HideStatusBar()))
;
    SwitcherStatus->setChecked(true);
    SwitcherHightlighter=MenuView->addAction("On/off syntaxic
highlight");
    MenuView->actions().back()->setCheckable(true);
    SwitcherHightlighter->setChecked(true);
   // SwitcherHightlighter->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\swhl.png"));

connect(SwitcherHightlighter,SIGNAL(triggered()),this,SLOT(HideHighlig
hter()));


    QMenu* m = new QMenu( "Switcher Highlight", this);
    m->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\swhl.png"));
    group = new QActionGroup(this);
    C11 = new QAction( "C 11", this);
    C11 ->setCheckable(true);
    connect(C11,SIGNAL(triggered()),this,SLOT(v11()));
    C18 = new QAction( "C 18", this);
    C18 ->setCheckable(true);
    connect(C18,SIGNAL(triggered()),this,SLOT(v18()));
    Cpp14 = new QAction( "C++ 14", this);
    Cpp14 ->setCheckable(true);
    connect(Cpp14,SIGNAL(triggered()),this,SLOT(v14()));
    C17 = new QAction( "C++ 17", this);
    C17 ->setCheckable(true);
    connect(C17,SIGNAL(triggered()),this,SLOT(v17()));
    C20 = new QAction( "C++ 20", this);
    C20->setCheckable(true);
    C20->setChecked(true);
    connect(C20,SIGNAL(triggered()),this,SLOT(v20()));

    group->addAction(C11);
    group->addAction(C18 );
    group->addAction(Cpp14);
    group->addAction(C17);
    group->addAction(C20);
    m->addActions(group->actions());
    Hightlighter=MenuView->addMenu(m);
    MenuView->actions().back()->setCheckable(true);
    cSyn=MenuView->addAction("Change syntax");
    cSyn->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\changesw.png"));
```

```
    connect(cSyn,SIGNAL(triggered()),this,SLOT(cSyntax()));


    AboutProgram=MenuHelp->addAction("About the program");
    connect(AboutProgram, SIGNAL(triggered()), this,
SLOT(showAboutProgram()));

    len = new QLabel("Lines Count: " + QString::number(editor-
>document()->blockCount())+" "

                           +  " Words Count: "+ QString::number(editor-
>toPlainText().split(QRegularExpression("(\\s|\\n|\\r)+"),
Qt::SkipEmptyParts).count())
                           + " " + "Simbols Count: " +
QString::number(editor->toPlainText().size())
                           +" | "+
QString::number(floor(double(1024*(editor-
>toPlainText().size())*8)/1024 + 0.5)/1024)+" KB" );

   curPos = new QLabel("Line: "+QString::number((editor-
>textCursor().blockNumber()))+" "
                               + "Column: " +QString::number(editor-
>textCursor().positionInBlock()));
   last_change = new QLabel("Default");
   tool();

    statusBar()->addWidget(curPos);
    statusBar()->addWidget(last_change);
    statusBar()->addWidget(len);
    connect(editor,SIGNAL(cursorPositionChanged()),this
,SLOT(RefreshStatusBar()));
    connect(editor,SIGNAL(cursorPositionChanged()),this
,SLOT(RefreshZv()));
     highlighter = new Highlighter(editor->document(),5);
}
void MainWindow::cSyntax(){
syntax->key->setText((highlighter-
>getKeywordColor()).color()).name());
syntax->func->setText((highlighter-
>getFunctionColor()).color()).name());
syntax->class_->setText((highlighter-
>getClassColor()).color()).name());
syntax->exec();
QColor a;
a.setNamedColor(syntax->key->text());
QColor b;
b.setNamedColor(syntax->func->text());
QColor c;
c.setNamedColor(syntax->class_->text());
int k=highlighter->number;
delete highlighter;
```

```
highlighter =new Highlighter(editor->document(),k,a,b,c);
if (!syntax->name.isEmpty())
setWindowTitle(syntax->name);
QFile file("config.txt");

if (file.open(QIODevice::WriteOnly)){
        file.resize(0);
        file.write((syntax->key->text()+syntax->func->text()+syntax-
>class_->text()).toUtf8());
        file.flush();
        file.close();
}
update();
}
void MainWindow::v11(){
    highlighter = new Highlighter(editor->document(),1);
}

void MainWindow::v18(){
    highlighter = new Highlighter(editor->document(),2);
}

void MainWindow::v14(){
    highlighter = new Highlighter(editor->document(),3);
}

void MainWindow::v17(){
    highlighter = new Highlighter(editor->document(),4);
}

void MainWindow::v20(){
    highlighter = new Highlighter(editor->document(),5);
}
void MainWindow::RefreshZv(){
    setWindowModified(true);
}
void MainWindow::HideToolBar(){
if (SwitcherPanel->isChecked()==false)
    toolbar->hide();
else
    toolbar->show();
}
void MainWindow::HideHighlighter(){
        if (SwitcherHightlighter->isChecked()==false){
            delete highlighter;
        }

        else if (C11->isChecked()==true && SwitcherHightlighter-
>isChecked()== true){
            highlighter = new Highlighter(editor->document(),1);
```

```
        }
        else if(C18->isChecked()==true && SwitcherHightlighter-
>isChecked()== true){
            highlighter = new Highlighter(editor->document(),2);

        }
        else if(Cpp14->isChecked()==true && SwitcherHightlighter-
>isChecked()== true){
            highlighter = new Highlighter(editor->document(),3);

        }
        else if(C17->isChecked()==true && SwitcherHightlighter-
>isChecked()== true){
            highlighter = new Highlighter(editor->document(),4);

        }
        else if(C20->isChecked()==true && SwitcherHightlighter-
>isChecked()== true){
            highlighter = new Highlighter(editor->document(),5);

        }
}
void MainWindow::tool(){
   New->setIcon(style()->standardIcon(QStyle::SP_FileIcon));
   toolbar->addAction(New);
   Open->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\open.png"));
   toolbar->addAction(Open);
   Save->setIcon(style()->standardIcon(QStyle::SP_DialogSaveButton));
   toolbar->addAction(Save);
   Cancel->setIcon(style()->standardIcon(QStyle::SP_ArrowLeft));
   toolbar->addAction(Cancel);
   Repeat->setIcon(style()->standardIcon(QStyle::SP_ArrowRight));
   toolbar->addAction(Repeat);
   Copy->setIcon(style()->standardIcon(QStyle::SP_DialogResetButton));
   toolbar->addAction(Copy);
   Cut->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\cut.png"));
   toolbar->addAction(Cut);
   Paste->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\paste.jpg"));
   toolbar->addAction(Paste);
   Save_as->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\save_as.png"));
   Exit->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\exit.jpg"));
   Font->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\font.png"));
   ColorBackground->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\color1.png"));
```

```
   ColorCurString->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\color2.png"));
   Find->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\find.png"));
   Find_and_replace->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\find1.jpg"));
   AboutProgram->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\help.jpg"));
   Highlight->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\selectall.png"));
   QMenu *toolButton_menu = new QMenu("Find/Find and Replace");
   QToolButton* toolButton = new QToolButton();
   toolButton_menu->addAction(Find);
   toolButton_menu->addAction(Find_and_replace);
   toolButton->setMenu(toolButton_menu);
   toolButton->setIcon(QIcon("C:\\Users\\The Witcher
Hunt\\Downloads\\find3.jpg"));
   toolButton->setPopupMode(QToolButton::InstantPopup);
   toolbar->addWidget(toolButton);

}
void MainWindow::mousePressEvent(QMouseEvent *event){
    if (event->button() == Qt::RightButton){
       context();
    }
}
void MainWindow::setCurName(const QString &a)
 {
     NameFile = a;
     QString b;
     editor->document()->setModified(false);
     if (a.isEmpty())
         b = "untitled.txt";
     else
         b = QFileInfo(a).fileName();
     name=b;editor->name=b;
     setWindowTitle(tr("[*]%1 - %2").arg(b).arg("Laboratory work 5"));
     setWindowModified(false);


}
void MainWindow::New_(){
    editor->clear();
    setCurName(QString());
}

void MainWindow::Open_(){
        QString  a = QFileDialog::getOpenFileName(this, "","");
        if (!a.isEmpty())
             openDoc(a);
}
```

```
void MainWindow::context(){
    QMenu* m = new QMenu(this);

    QAction* Undo = new QAction("Undo", this);
    connect(Undo,SIGNAL(triggered()), editor, SLOT(undo1()));

    QAction* Redo = new QAction("Redo", this);
    connect(Redo,SIGNAL(triggered()),editor, SLOT(redo1()));

    QAction* select_word = new QAction("Select word", this);
    connect(select_word,SIGNAL(triggered()),this, SLOT(SelectWord()));

    QAction* select_line = new QAction("Select line", this);
    connect(select_line,SIGNAL(triggered()),this, SLOT(SelectLine()));

    QAction* copy = new QAction("Copy", this);
    copy->setEnabled(false);
    connect(copy,SIGNAL(triggered()), editor, SLOT(copy()));
    connect(editor, SIGNAL(copyAvailable(bool)), copy,
SLOT(setEnabled(bool)));

    QAction* cut = new QAction("Cut", this);
    cut->setEnabled(false);
    connect(cut,SIGNAL(triggered()),editor, SLOT(cut()));
    connect(editor, SIGNAL(copyAvailable(bool)), cut,
SLOT(setEnabled(bool)));

    QAction* paste = new QAction("Paste", this);
    connect(paste,SIGNAL(triggered()),editor, SLOT(paste()));
    paste ->setEnabled(false);

    QAction* Delete = new QAction("Delete", this);
    connect(Delete ,SIGNAL(triggered()),this, SLOT(Delete()));

    QAction* selectAll = new QAction("Select all", this);
    connect(selectAll,SIGNAL(triggered()),editor, SLOT(selectAll()));

    if (editor->textCursor().hasSelection()){
        copy -> setEnabled(true);
        cut -> setEnabled(true);
        paste -> setEnabled(true);
    }

    if (editor->document()->isEmpty()){
        select_word->setEnabled(false);
        select_line->setEnabled(false);
    }
    else if (!editor->document()->isEmpty()){
        select_word->setEnabled(true);
        select_line->setEnabled(true);
```

```
    }
    QClipboard *p=QApplication::clipboard();
   if(p->ownsClipboard())
       paste->setEnabled(true);
    m->addAction(Undo);
    m->addAction(Redo);
    m->addAction(select_word);
    m->addAction(select_line);
    m->addAction(copy);
    m->addAction(cut);
    m->addAction(paste);
    m->addAction(selectAll);
    m->addAction(Delete);
    m->exec(QCursor::pos());
}
void MainWindow::SearchandReplace(){
    if (!rsearch)
        rsearch = new ReplaceSearch;
    rsearch -> show();
    connect(rsearch, SIGNAL(SignalofReplace()),this
,SLOT(replaceEND()));
}
void MainWindow::ActiveSearch(){
    if (!search)
        search = new Search;
    search->show();
    connect(search, SIGNAL(accepted()),this,SLOT(FindWord()));
}
void MainWindow::SetFontSelection(){
    bool flag;
    QFont font = QFontDialog::getFont(&flag, QFont("Arial", 12),
this);
    if (flag) {
        editor->setFont(font);
    }
}
bool MainWindow::openDoc(const QString &a){
     if (!QFile::exists(a))
         return 0;
     QFile f(a);
     if (!f.open(QFile::ReadOnly))
         return 0;
     QByteArray data = f.readAll();
     QString s = QString::fromLocal8Bit(data);
     editor->setPlainText(s);
     setCurName(a);
     return 1;
 }
void MainWindow::FindWord(){
    bool flag = 0;
    QString str = search->Line();
```

```
    QTextDocument *doc = editor->document();

    QTextCursor highlightCursor(doc);
    QTextCursor cursor(doc);

    cursor.beginEditBlock();

    QTextCharFormat plainFormat(highlightCursor.charFormat());
    QTextCharFormat colorFormat = plainFormat;
    colorFormat.setForeground(Qt::blue);

    while (!highlightCursor.isNull() && !highlightCursor.atEnd()) {

        highlightCursor = doc->find(str, highlightCursor,
QTextDocument::FindWholeWords);

        if (!highlightCursor.isNull()) {
            flag = true;
            highlightCursor.movePosition(QTextCursor::WordRight,
QTextCursor::KeepAnchor);
            highlightCursor.mergeCharFormat(colorFormat);
        }
    }
    cursor.endEditBlock();
    f = false;
    if (flag == false)
        QMessageBox::information(this,"Text not found","Please, repeat
your action again");
}
void MainWindow::HideNumberArea(){
    if (SwitcherNumeration->isChecked()==false){

         editor->getLineNumberArea()->hide();
    }
    else{
       editor->getLineNumberArea()->show();
    }
}
void MainWindow::WrapWord(){
    if (Perenos->isChecked()==false){
        editor->setWordWrapMode(QTextOption::NoWrap);
    }
    else{
        editor->setWordWrapMode(QTextOption::WordWrap);
    }
}
void MainWindow::exit()
    {
    setting = new QSettings("editor.ini", QSettings::IniFormat, this);

    setting->beginGroup("Current_settings");
```

```
    if (SwitcherPanel->isChecked()){
        setting->setValue("ToolBar", "On");
    }
    else{
        setting->setValue("ToolBar", "Off");
    }

    if (C11->isChecked()){
        setting->setValue("C", "11");
    }
    if (C18->isChecked()){
       setting->setValue("C", "18");
    }
    if (Cpp14->isChecked()){
       setting->setValue("C++", "14");
    }
    if (C17->isChecked()){
       setting->setValue("C++", "17");
    }
    if (C20->isChecked()){
        setting->setValue("C++", "20");
    }

    if (SwitcherNumeration->isChecked()){
        setting->setValue("NumerationLine", "On");
    }
    else{
        setting->setValue("NumerationLine", "Off");
    }

    if (cSyn->isChecked()){
        setting->setValue("SyntaxHighlight", "On");
    }
    else{
        setting->setValue("SyntaxHighlight", "Off");
    }
    if (SwitcherStatus->isChecked()){
        setting->setValue("StatusBar", "On");
    }
    else{
        setting->setValue("StatusBar", "Off");
    }
    setting->endGroup();
    this->close();
    }
void MainWindow::ChangeColorLine(){
    editor->setColorLine();

}
MainWindow::~MainWindow()
{
```

```
}
void MainWindow::SelectWord(){
   QTextCursor a = editor->textCursor();
   a.select(QTextCursor::WordUnderCursor);
   editor->setTextCursor(a);
}
 void MainWindow::showAboutProgram()
{

     DialogAboutMe *p=new DialogAboutMe(this);
     p->exec();
}
 void MainWindow::RefreshStatusBar(){
     curPos->setText("Line: "+QString::number((editor-
>textCursor().blockNumber())+1)+" "
                                    + "Column: " +QString::number(editor-
>textCursor().positionInBlock()+1));
     len->setText("Lines Count: " + QString::number(editor-
>document()->blockCount())+" "

                   +  " Words Count: "+ QString::number(editor-
>toPlainText().split(QRegularExpression("(\\s|\\n|\\r)+"),
Qt::SkipEmptyParts).count())
                   + " " + "Simbols Count: " + QString::number(editor-
>toPlainText().size())
                   +" | "+ QString::number(floor(double(1024*(editor-
>toPlainText().size())*8)/1024 + 0.5)/1024)+" KB" );
      last_change->setText((dateTime->currentDateTime()).toString("HH-
mm")+" "+(sdateTime->currentDateTime()).toString());
 }
 void MainWindow::HideStatusBar(){
     if (SwitcherStatus->isChecked()==false){
         statusBar()->hide();
     }
     else {
         statusBar()->show();
     }
 }
 void MainWindow::SetColorBackground(){
    editor->setColorBackground();
 }
 void MainWindow::replaceEND(){
     if(!editor->find(rsearch->line())){
         QMessageBox::information(this, "Text not found" ,"Please,
repeat your action again.");
         rsearch->hide();
         return;
     }
     editor->textCursor().clearSelection();
     editor->textCursor().movePosition(QTextCursor::NextWord,
QTextCursor::KeepAnchor);
```

```
    editor->textCursor().insertText(rsearch->replace_area());
}
void MainWindow::SelectLine(){
   QTextCursor a = editor->textCursor();
   a.select(QTextCursor::LineUnderCursor);
   editor->setTextCursor(a);
}
void MainWindow::Delete(){
    if (editor->textCursor().hasSelection())
        editor->textCursor().removeSelectedText();
}
bool MainWindow::Save_(){
    setWindowModified(false);
     if (NameFile.isEmpty())
        return Save_As_();
     QFile file(NameFile);
     if (file.open(QFile::WriteOnly))
         file.write(editor->toPlainText().toUtf8());
     curPos->setText((dateTime->currentDateTime()).toString("yyyy-MM-
dd")+" " + (dateTime->currentDateTime()).toString("HH-mm"));
     return 0;
}
bool MainWindow::Save_As_(){
    QString a = QFileDialog::getSaveFileName(this, "","");
    if (a.isEmpty())
       return false;
    setCurName(a);
    return Save_();
}
```

Приложение 3

Листинг-18 main.cpp

```
#include "mainwindow.h"
#include "QLabel"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.resize(800,600);
    w.show();
    return a.exec();
}
```