Отчет по практической работе №28

Наименование работы: Компоновка и элементы управления в WPF.

Индивидуальные задания (вариант №9):

1. Разработать приложение для осуществления над матрицами следующих операций: сложение, вычитание, умножение матриц, умножение матрицы на число, сортировка (нечётные варианты – сортировка элементов строк матрицы, чётные - столбцов), транспонирование, нахождение обратной матрицы, подсчёт определителя матрицы.
2. В Индивидуальное задание №2 используя элементы меню(задание 1) и панель инструментов(задание 2) добавьте возможность сохранить файл, открыть файл и прочитать данные, сохранить файл как… , изминение шрифта и размера текста.

Ход работы:

1. Разработал программу в соответствии с заданием №1. Были реализованы условия, методы, циклы, двумерные массивы.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Lab28.\_1

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

int[,] arr;

int[,] arr2;

int n;

private void bGen\_Click(object sender, RoutedEventArgs e)

{

n = Convert.ToInt32(tn.Text);

arr = new int[n, n];

Random rnd = new Random();

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

{

for (int j = 0; j < n; j++)

{

arr[i, j] = rnd.Next(-100, 100);

}

}

arr2 = new int[n, n];

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

{

for (int j = 0; j < n; j++)

{

arr2[i, j] = rnd.Next(-100, 100);

}

}

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

{

for (int j = 0; j < n; j++)

{

lArrA.Content += arr[i, j] + " ";

lArrB.Content += arr2[i, j] + " ";

}

lArrA.Content += "\n";

lArrB.Content += "\n";

}

}

int[,] arr\_rez;

double opr;

private void bRez\_Click(object sender, RoutedEventArgs e)

{

try

{

if (radioButton1.IsChecked == true)

{

label4.Content = "+";

arr\_rez = new int[n, n];

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

{

for (int j = 0; j < n; j++)

{

arr\_rez[i, j] = arr[i, j] + arr2[i, j];

}

}

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

{

for (int j = 0; j < n; j++)

{

lArrRez.Content += arr\_rez[i, j]+ " ";

}

lArrRez.Content += "\n";

}

}

if (radioButton2.IsChecked == true)

{

label4.Content = "-";

arr\_rez = new int[n, n];

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

{

for (int j = 0; j < n; j++)

{

arr\_rez[i, j] = arr[i, j] - arr2[i, j];

}

}

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

{

for (int j = 0; j < n; j++)

{

lArrRez.Content += arr\_rez[i, j] + " ";

}

lArrRez.Content += "\n";

}

}

if (radioButton3.IsChecked == true)

{

label4.Content = "\*";

arr\_rez = new int[n, n];

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

{

for (int j = 0; j < n; j++)

{

arr\_rez[i, j] = arr[i, j] \* arr2[i, j];

}

}

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

{

for (int j = 0; j < n; j++)

{

lArrRez.Content += arr\_rez[i, j] + " ";

}

lArrRez.Content += "\n";

}

}

if (radioButton4.IsChecked == true)

{

label4.Content = ":";

arr\_rez = new int[n, n];

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

{

for (int j = 0; j < n; j++)

{

arr\_rez[i, j] = arr[i, j] / arr2[i, j];

}

}

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

{

for (int j = 0; j < n; j++)

{

lArrRez.Content += arr\_rez[i, j] + " ";

}

lArrRez.Content += "\n";

}

}

if (radioButton5.IsChecked == true)

{

label4.Content = null;

arr\_rez = new int[n, n];

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

{

for (int j = 0; j < n; j++)

{

arr\_rez[i, j] = arr[i, j] \* Convert.ToInt32(textBox1.Text);

}

}

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

{

for (int j = 0; j < n; j++)

{

lArrRez.Content += arr\_rez[i, j] + " ";

}

lArrRez.Content += "\n";

}

}

if (radioButton6.IsChecked == true)

{

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

{

for (int j = 0; j < n; j++)

{

lArrRezQ.Content += arr\_rez[j, i] + " ";

}

lArrRez.Content += "\n";

}

}

if (radioButton7.IsChecked == true)

{

double sum1 = 1;

double sum2 = 1;

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

{

sum1 \*= arr\_rez[i, i];

}

for (int j = n - 1, i = 0; j >= 0; i++, j--)

{

sum2 \*= arr\_rez[i, j];

}

opr = sum1 - sum2;

MessageBox.Show(Convert.ToString("Определитель = " + opr));

}

if (radioButton8.IsChecked == true && opr != 0)

{

uint round = 0;

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

{

for (int j = 0; j < n; j++)

{

double tmp = arr\_rez[i, j];

arr\_rez[i, j] = Convert.ToInt32(round == 0 ? (1 / opr) \* opr : Math.Round(((1 / opr) \* opr), (int)round, MidpointRounding.ToEven));

}

}

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

{

for (int j = 0; j < n; j++)

{

lArrRezQ.Content += arr\_rez[i, j] + " ";

}

lArrRezQ.Content += "\n";

}

}

else if (opr == 0)

{

MessageBox.Show("Определитель равен нулю, обратной матрицы не существует!");

}

}

catch (Exception)

{

MessageBox.Show("Error!");

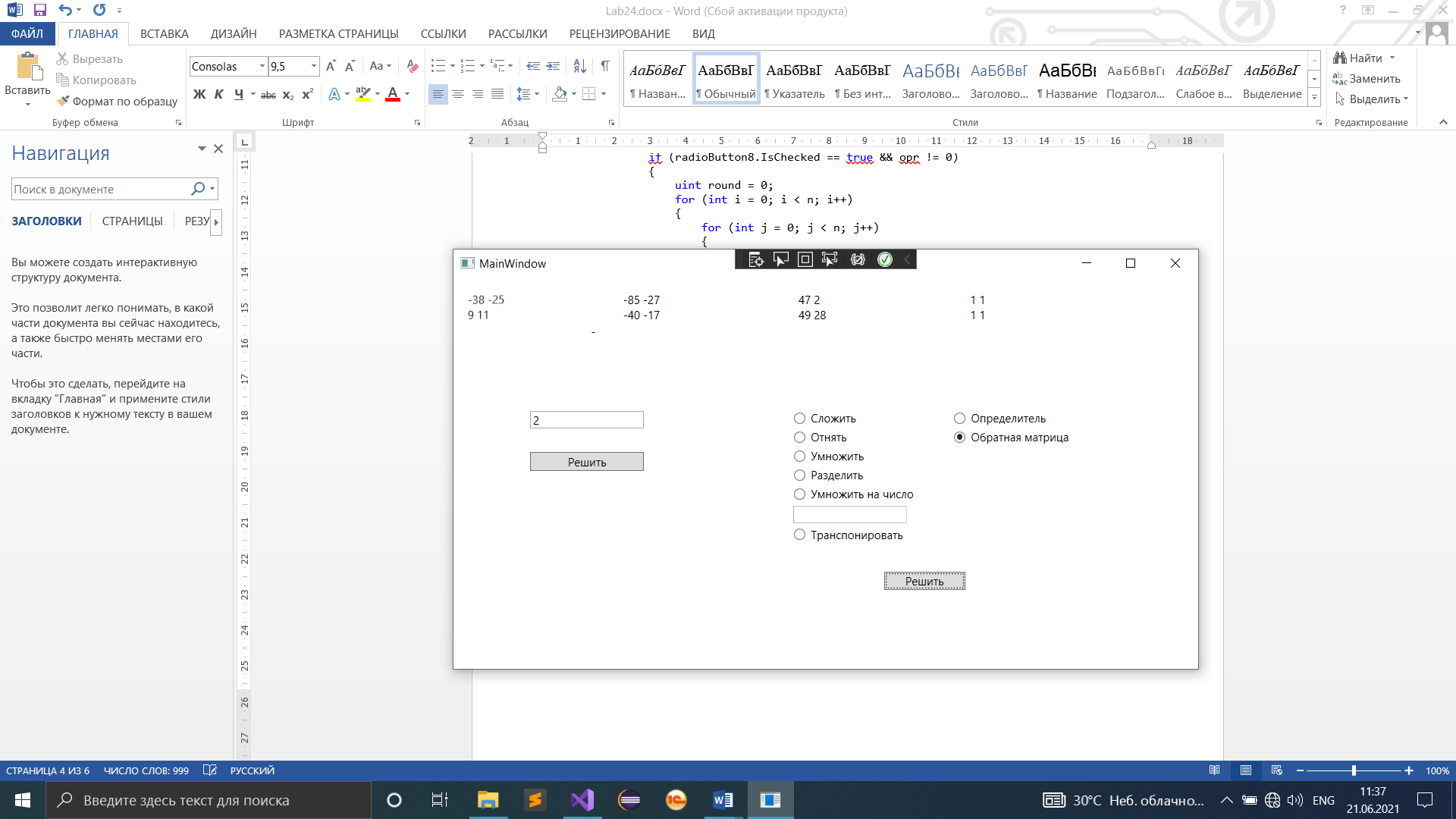
}

}

}

}

Результат выполнения программы Рис 1.1.



**Рисунок 1.1 – Пример работы программы WPF**

1. Разработал программу в соответствии с заданием №1. Были реализованы условия.

Листинг программы:

using System;

using System.Linq;

using System.Collections.Generic;

using System.IO;

using System.Windows;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using Microsoft.Win32;

using System.Windows.Controls;

namespace WpfTutorialSamples.Rich\_text\_controls

{

public partial class RichTextEditorSample : Window

{

public RichTextEditorSample()

{

InitializeComponent();

cmbFontFamily.ItemsSource = Fonts.SystemFontFamilies.OrderBy(f => f.Source);

cmbFontSize.ItemsSource = new List<double>() { 8, 9, 10, 11, 12, 14, 16, 18, 20, 22, 24, 26, 28, 36, 48, 72 };

}

private void rtbEditor\_SelectionChanged(object sender, RoutedEventArgs e)

{

object temp = rtbEditor.Selection.GetPropertyValue(Inline.FontWeightProperty);

temp = rtbEditor.Selection.GetPropertyValue(Inline.FontFamilyProperty);

cmbFontFamily.SelectedItem = temp;

temp = rtbEditor.Selection.GetPropertyValue(Inline.FontSizeProperty);

cmbFontSize.Text = temp.ToString();

}

private void cmbFontFamily\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (cmbFontFamily.SelectedItem != null)

rtbEditor.Selection.ApplyPropertyValue(Inline.FontFamilyProperty, cmbFontFamily.SelectedItem);

}

private void cmbFontSize\_TextChanged(object sender, TextChangedEventArgs e)

{

rtbEditor.Selection.ApplyPropertyValue(Inline.FontSizeProperty, cmbFontSize.Text);

}

private void btnOpen\_Click(object sender, RoutedEventArgs e)

{

OpenFileDialog dlg = new OpenFileDialog();

dlg.Filter = "Rich Text Format (\*.rtf)|\*.rtf|All files (\*.\*)|\*.\*";

if (dlg.ShowDialog() == true)

{

FileStream fileStream = new FileStream(dlg.FileName, FileMode.Open);

TextRange range = new TextRange(rtbEditor.Document.ContentStart, rtbEditor.Document.ContentEnd);

range.Load(fileStream, DataFormats.Rtf);

}

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

SaveFileDialog dlg = new SaveFileDialog();

dlg.Filter = "Rich Text Format (\*.rtf)|\*.rtf|All files (\*.\*)|\*.\*";

if (dlg.ShowDialog() == true)

{

FileStream fileStream = new FileStream(dlg.FileName, FileMode.Create);

TextRange range = new TextRange(rtbEditor.Document.ContentStart, rtbEditor.Document.ContentEnd);

range.Save(fileStream, DataFormats.Rtf);

}

}

private void btnSave\_Click\_1(object sender, RoutedEventArgs e)

{

SaveFileDialog dlg = new SaveFileDialog();

dlg.Filter = "Rich Text Format (\*.rtf)|\*.rtf|All files (\*.\*)|\*.\*";

if (dlg.ShowDialog() == true)

{

FileStream fileStream = new FileStream(dlg.FileName, FileMode.Create);

TextRange range = new TextRange(rtbEditor.Document.ContentStart, rtbEditor.Document.ContentEnd);

range.Save(fileStream, DataFormats.Rtf);

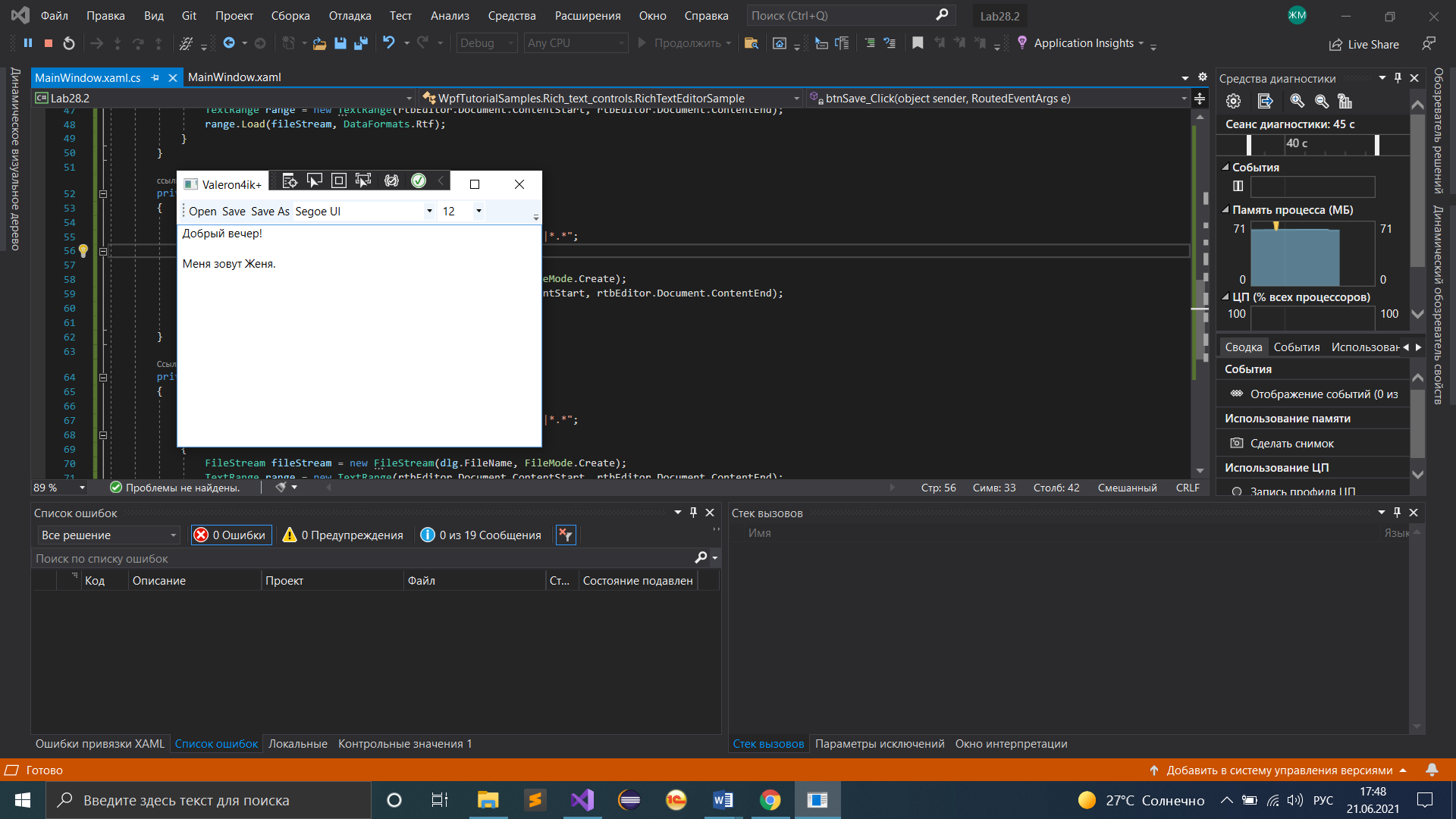
}

}

}

}

Результат выполнения программы Рис 2.1.



**Рисунок 2.1 – Пример работы программы WPF**

Отметка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись преподавателя\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_