Лабораторная работа №6

Условие:

1. Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Поменять местами i- и j-ю буквы. Для ввода i и j на форме добавить свои поля вода;
2. Дана строка. Подсчитать, сколько различных символов встречается в ней. Вывести их на экран.
3. В записке слова зашифрованы – каждое из них записано наоборот. Расшифровать сообщение.
4. Составить программу, которая будет вводить строку в переменную string. Найти слово, встречающееся в каждом предложении, или сообщить, что такого слова нет.
5. Выберите 10 произвольных букв русского алфавита. Введите произвольное слово. С помощью ключа длиной от 3 до 8 символов произведите шифровку слова в числовую комбинацию.

В Е Ж М Н О П Р С Т КЛЮЧ - 1234

0 1 2 3 4 5 6 7 8 9

МНОЖЕСТВО

3 4 5 2 1 8 9 0 5

1 2 3 4 1 2 3 4 1

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

4 6 8 6 3 1 2 4 6 - РЕЗУЛЬТАТ

Решение:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

/\*

Задание 1.

Дана строка символов,

состоящая из произвольного текста на английском языке,

слова разделены пробелами.

Поменять местами i- и j-ю буквы.

Для ввода i и j на форме добавить свои поля ввода.

\*/

Console.WriteLine("Задание 1.");

string str1new = "", str1 = "";

back1begin: Console.WriteLine("Введите текст: ");

str1 = Console.ReadLine();

if (str1.Length < 2)

goto back1begin;

back1begintext: Console.Write("Введите i-позицию: ");

int i1 = Int32.Parse(Console.ReadLine());

Console.Write("Введите j-позицию, большую чем i: ");

int j1 = Int32.Parse(Console.ReadLine());

if (i1 > j1 || i1 > str1.Length || j1 > str1.Length)

{

goto back1begintext;

}

for (int i = 0; i < i1; i++)

{

str1new += str1.Substring(i, 1);

}

str1new += str1.Substring(j1, 1);

for (int i = i1 + 1; i < j1; i++)

{

str1new += str1.Substring(i, 1);

}

str1new += str1.Substring(i1, 1);

for (int i = j1 + 1; i < str1.Length; i++)

{

str1new += str1.Substring(i, 1);

}

Console.WriteLine(str1new);

Console.WriteLine();

/\*

Задание 2-3.

1. Дана строка. Подсчитать, сколько различных символов встречается в ней. Вывести их на экран.

2. В записке слова зашифрованы – каждое из них записано наоборот. Расшифровать сообщение.

\*/

Console.WriteLine("Задание 2-3.");

Console.Write("Введите слово: ");

string text2 = Console.ReadLine();

Dictionary<char, int> dictionarys = text2.GroupBy(x => x)

.ToDictionary(x => x.Key, x => x.Count());

foreach (KeyValuePair<char, int> keyValuePair in dictionarys)

{

Console.WriteLine("{0} : {1}", keyValuePair.Key, keyValuePair.Value);

}

char[] arr2 = text2.ToCharArray();

Array.Reverse(arr2);

Console.WriteLine(arr2);

Console.WriteLine();

/\*

Задание 4.

Составить программу, которая будет вводить строку в переменную string.

Найти слово, встречающееся в каждом предложении, или сообщить, что такого слова нет.

\*/

Console.WriteLine("Задание 4.");

Console.Write("Введите текст: ");

string[] slova = Console.ReadLine().Split(' ');

List<string> list = new List<string>();

Console.WriteLine("Повторяющиеся слова:");

for (int i = 0; i < slova.Length; i++)

{

for (int j = 0; j < slova.Length; j++)

{

if (i != j)

{

if (slova[i] == slova[j] && !list.Contains(slova[i]))

list.Add(slova[i]);

}

}

}

foreach (string slovo in list.ToArray())

Console.WriteLine(slovo);

Console.WriteLine();

/\*

Задание 5.

Выберите 10 произвольных букв русского алфавита.Введите произвольное слово.

С помощью ключа длиной от 3 до 8 символов произведите шифровку слова в числовую комбинацию.

В Е Ж М Н О П Р С Т КЛЮЧ - 1234

0 1 2 3 4 5 6 7 8 9

МНОЖЕСТВО

3 4 5 2 1 8 9 0 5

1 2 3 4 1 2 3 4 1

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

4 6 8 6 3 1 2 4 6 – РЕЗУЛЬТАТ

\*/

Console.WriteLine("Задание 5.");

string a5 = "АКОЛИНСБУЗ", key5 = "", result5str = "";

int ii5 = 0;

Console.Write("Введите слово, содержащее буквы А К О Л И Н С Б У З: ");

string b5 = Console.ReadLine();

Console.Write("Введите длину ключа от 3 до 8: ");

back5begin: int key5in = Int32.Parse(Console.ReadLine());

if (key5in < 3 || key5in > 8)

goto back5begin;

for (int i = 0; i < b5.Length; i++) //Состовляем ключ в строку по длине введенного слова

{

ii5++;

if (ii5 <= key5in)

{

key5 += ii5;

}

else

{

ii5 = 1;

key5 += ii5;

}

}

Console.WriteLine("Полученный ключ: {0}.", key5);

int n5 = b5.Length + 1;

int [] mas5 = new int [n5];

int[] mas5new = new int[n5];

Console.Write("Полученные коды букв: ");

for (int i = 0; i < b5.Length; i++)

{

mas5[i] = (a5.IndexOf(b5[i])); //Получаем индекс буквы

mas5new[i] = mas5[i] + Int32.Parse(key5.Substring(i, 1)); //Получаем шифрованную букву

Console.Write("{0} ", mas5new[i]);

}

Console.WriteLine();

//Вписываем шифр в строку обратной последовательностью

for (int i = b5.Length - 1; i > -1; i--) //Посимвольно складываем шифрованные буквы в строку задом наперед

{

if (mas5new[i] > 9 && i != 0) //Если число двузначное и не является первым результатной строки,

//записываем из числа второй символ; следующая шифрованная цифра увеличивается на 1

{

result5str += Convert.ToString(mas5new[i]).Substring(1, 1);

mas5new[i - 1] += 1;

}

else if (mas5new[i] < 10) //Добавляем в строку шифрованную цифру

{

result5str += Convert.ToString(mas5new[i]);

}

if (i == 0 && mas5new[i] > 9) //Если число двузначное и первое результатной строки, записываем в обратном виде (было 10, стало 01)

{

result5str += Convert.ToString(mas5new[i]).Substring(1, 1);

result5str += Convert.ToString(mas5new[i]).Substring(0, 1);

}

}

//Переворачиваем строку в результатную

char[] result5strnew = result5str.ToCharArray();

Array.Reverse(result5strnew);

Console.WriteLine(result5strnew);

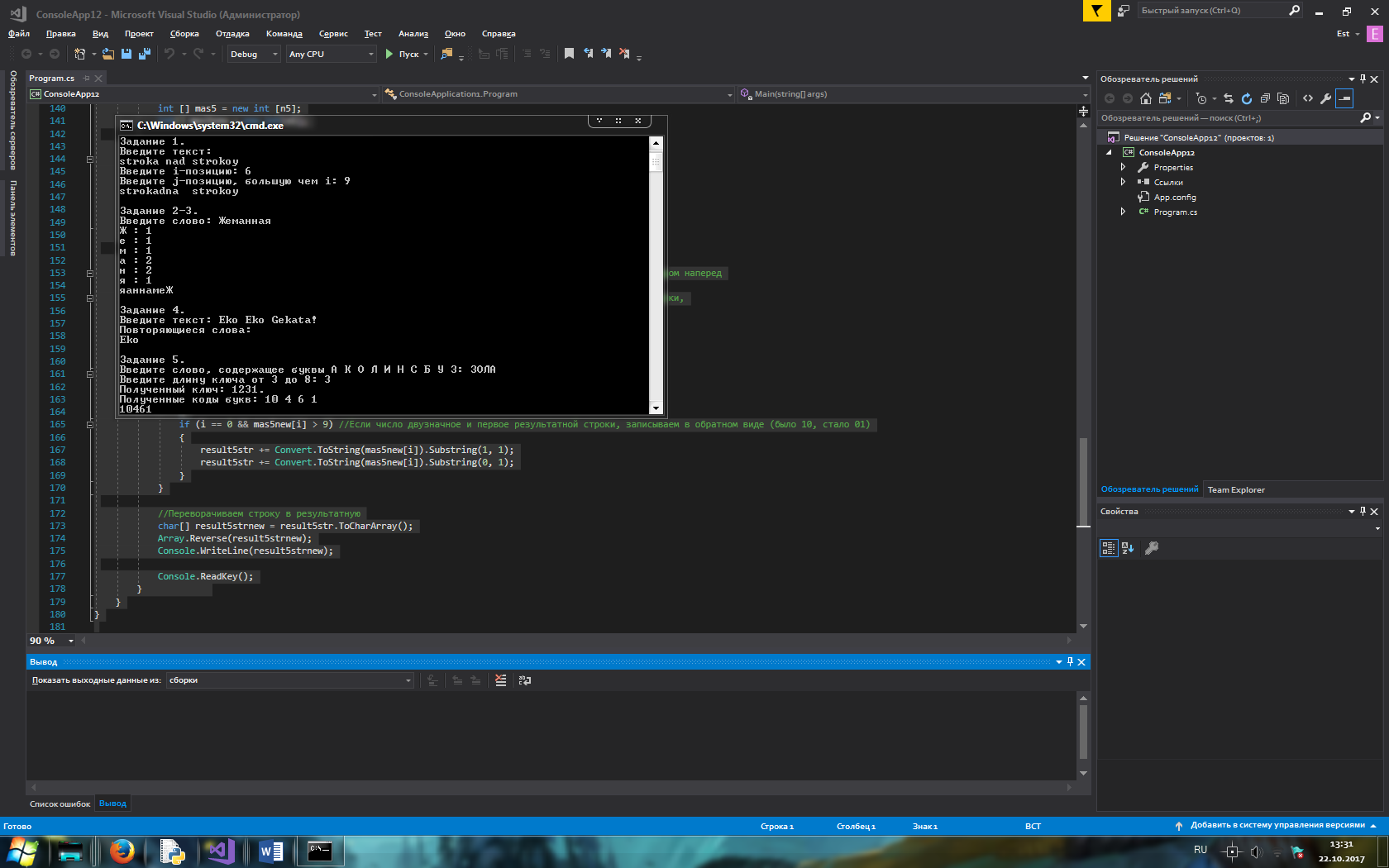
Console.ReadKey();

}

}

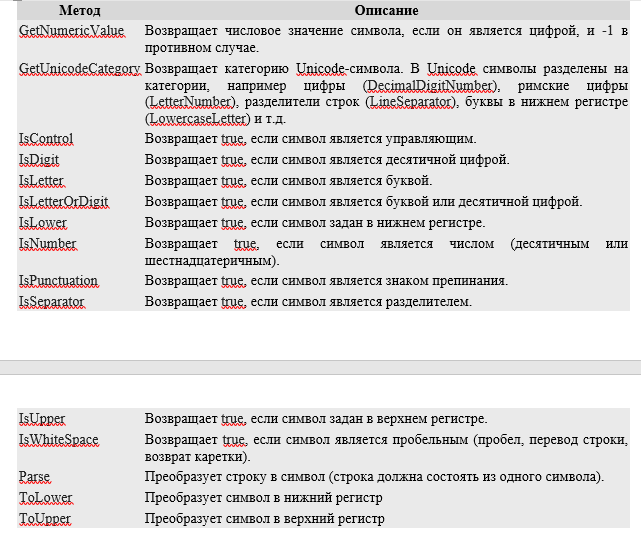
}

Тестирование:



Контрольные вопросы:

1. Обработка текстовой информации является одной из самых распространенных задач современного программировании. С# предоставляет для ее решения широкий набор средств: символы char, неизменяемые строки string, изменяемые строки StringBuider и регулярные выражения Regex. В данном разделе мы рассмотрим работу с символами, неизменяемыми и изменяемыми строками.
2. Символьный тип char предназначен для хранения символа в кодировке Unicode. Символьный тип относится к встроенным типам данных С# и соответствует стандартному классу Сhar библиотеки .Net из пространства имен System. В этом классе определены статические методы, позволяющие задавать вид и категорию символа, а также преобразовывать символ в верхний или нижний регистр, в число. Рассмотрим основные методы:



|  |  |  |
| --- | --- | --- |
| **Название** | **Вид** | **Описание** |
| Compare | Статический метод | Сравнение двух строк в лексикографическом (алфавитном) порядке. Разные реализации метода позволяют сравнивать строки с учетом или без учета регистра. |
| CompareTo | Метод | Сравнение текущего экземпляра строки с другой строкой. |
| Concat | Статический метод | Слияние произвольного числа строк. |
| Copy | Статический метод | Создание копии строки |
| Empty | Статическое поле | Открытое статическое поле, представляющее пустую строку |
| Format | Статический метод | Форматирование строки в соответствии с заданным форматом |
| IndexOf, IndexOfAny, LastIndexOf, LastIndexOfAny | Экземплярные методы | Определение индексов первого и последнего вхождения заданной подстроки или любого символа из заданного набора в данную строку. |
| Insert | Экземплярный метод | Вставка подстроки в заданную позицию |
| Join | Статический метод | Слияние массива строк в единую строку. Между элементами массива вставляются разделители. |
| Length | Свойство | Возвращает длину строки |
| PadLeft, PadRigth | Экземплярные методы | Выравнивают строки по левому или правому краю путем вставки нужного числа пробелов в начале или в конце строки. |
| Remove | Экземплярный метод | Удаление подстроки из заданной позиции |
| Replace | Экземплярный метод | Замена всех вхождений заданной подстроки или символа новыми подстрокой или символом. |
| Split | Экземплярный метод | Разделяет строку на элементы, используя разные разделители. Результаты помещаются в массив строк. |
| StartWith, EndWith | Экземплярные методы | Возвращают true или false в зависимости от того, начинается или заканчивается строка заданной подстрокой. |
| Substring | Экземплярный метод | Выделение подстроки, начиная с заданной позиции |
| ToCharArray | Экземплярный метод | Преобразует строку в массив символов |
| ToLower, ToUpper | Экземплярные методы | Преобразование строки к нижнему или верхнему регистру |
| Trim, TrimStart, TrimEnd | Экземплярные методы | Удаление пробелов в начале и конце строки или только с одного ее конца. |