

## Практическая работа 9

### Символьные данные. Методы обработки символьных данных

#### Цель занятия

Получить практические навыки объявления и обработки символьной информации в модуле.

#### Перечень оборудования и программного обеспечения

Персональный компьютер  
Microsoft Office (Word, Visio)  
Microsoft Visual Studio 2010

#### Краткие теоретические сведения

##### Строки

В языке C# определен класс `char[]`, и его можно использовать для представления строк постоянной длины. Однако массив `char[]` - это обычный массив, поэтому его нельзя инициализировать строкой символов. В C# не определено преобразование из класса `char[]` в класс `String`. У `String` есть динамический метод `ToCharArray`, задающий подобное преобразование в `char[]`.

Объекты класса `String` объявляются с явной или отложенной инициализацией, с явным или неявным вызовом конструктора класса. Чаще всего, при объявлении строковой переменной конструктор явно не вызывается, а инициализация задается строковой константой. У класса `String` достаточно много конструкторов, которые позволяют сконструировать строку из:

- символа, повторяющегося указанное число раз;
- массива символов `char[]` ;
- части массива символов.

Методы класса `String` позволяют выполнять вставку, удаление, замену, поиск вхождения подстроки в строку.

Имя	Описание
<u>Clone</u>	Возвращает ссылку на данный экземпляр класса <u>String</u> .
<u>Compare(String, String)</u>	Сравнивает два указанных объекта <u>String</u> и возвращает целое число, которое показывает их относительное положение в порядке сортировки.
<u>Concat(Object)</u>	Создает строковое представление указанного объекта.
<u>Contains</u>	Возвращает значение, указывающее, содержит ли данная строка заданный объект типа <u>String</u> .

<u>Copy</u>	Создает новый экземпляр <u>String</u> , имеющий то же значение, что и заданный экземпляр <u>String</u> .
<u>EndsWith(String)</u>	Определяет, совпадает ли конец данного экземпляра строки с указанной строкой.
<u>Equals(String)</u>	Определяет равны ли значения этого экземпляра и указанного объекта <u>String</u> .
<u>Equals(String, String)</u>	Определяет, равны ли значения двух указанных объектов <u>String</u> .
<u>Equals(String, StringComparison)</u>	Определяет, равны ли значения этой строки и указанного объекта <u>String</u> . Параметр определяет язык и региональные параметры, учет регистра и правила сортировки, используемые при сравнении.
<u>Format(String, Object)</u>	Заменяет один или более элементов формата в указанной строке строковым представлением указанного объекта.
<u>GetHashCode</u>	Возвращает хэш-код для этой строки. (Переопределяет <u>Object.GetHashCode()</u> .)
<u>IndexOf(Char)</u>	Возвращает индекс с отсчетом от нуля первого вхождения указанного символа Юникода в данной строке.
<u>IndexOf(String)</u>	Возвращает индекс с отсчетом от нуля первого вхождения значения указанной строки в данном экземпляре.
<u>IndexOf(Char, Int32)</u>	Возвращает индекс с отсчетом от нуля первого вхождения указанного символа Юникода в данной строке. Поиск начинается с указанной позиции знака.
<u>Insert</u>	Возвращает новую строку, в которой указанная строка вставляется на указанной позиции индекса в данном экземпляре.
<u>Intern</u>	Извлекает системную ссылку на указанный объект <u>String</u> .
<u>IsInterned</u>	Извлекает ссылку на указанный объект <u>String</u> .
<u>IsNormalized()</u>	Указывает, находится ли данная строка в форме нормализации Юникода С.
<u>IsNullOrEmpty</u>	Указывает, действительно ли указанная строка является строкой null или <u>Empty</u> .
<u>Join(String, IEnumerable&lt;String&gt;)</u>	Сцепляет элементы созданной коллекции <u>IEnumerable&lt;T&gt;</u> типа <u>String</u> , помещая между ними заданный разделитель.
<u>Join(String, String[])</u>	Сцепляет все элементы массива строк, помещая между ними заданный разделитель.
<u>LastIndexOf(Char)</u>	Возвращает позицию индекса с отсчетом от нуля последнего вхождения указанного символа Юникода в пределах данного экземпляра.
<u>LastIndexOf(String)</u>	Возвращает позицию индекса с отсчетом от нуля последнего вхождения указанной строки в данном экземпляре.

<u>Normalize()</u>	Возвращает новую строку, текстовое значение которой совпадает с данной строкой, а двоичное представление находится в нормализованной форме С Юникода.
<u>PadLeft(Int32)</u>	Возвращает новую строку, в которой знаки данного экземпляра выровнены по правому краю путем добавления слева пробелов до указанной общей длины.
<u>PadLeft(Int32, Char)</u>	Возвращает новую строку, в которой знаки данного экземпляра выровнены по правому краю путем добавления слева пробелов или указанного знака Юникода до указанной общей длины.
<u>PadRight(Int32)</u>	Возвращает новую строку, в которой знаки данной строки выровнены по левому краю путем добавления справа пробелов до указанной общей длины.
<u>PadRight(Int32, Char)</u>	Возвращает новую строку, в которой знаки данной строки выровнены по левому краю путем добавления справа пробелов или указанного знака Юникода до указанной общей длины.
<u>Remove(Int32)</u>	Возвращает новую строку, в которой были удалены все символы, начиная с указанной позиции и до конца в текущем экземпляре.
<u>Remove(Int32, Int32)</u>	Возвращает новую строку, в которой было удалено указанное число символов в указанной позиции.
<u>Replace(Char, Char)</u>	Возвращает новую строку, в которой все вхождения заданного знака Юникода в текущем экземпляре заменены другим заданным знаком Юникода.
<u>Replace(String, String)</u>	Возвращает новую строку, в которой все вхождения заданной строки в текущем экземпляре заменены другой заданной строкой.
<u>Split(Char[])</u>	Возвращает строковый массив, содержащий подстроки данного экземпляра, разделенные элементами заданного массива знаков Юникода.
<u>Split(Char[], Int32)</u>	Возвращает строковый массив, содержащий подстроки данного экземпляра, разделенные элементами заданного массива знаков Юникода. Параметр указывает максимальное число возвращаемых подстрок.
<u>StartsWith(String)</u>	Определяет, совпадает ли начало данного экземпляра строки с указанной строкой.
<u>StartsWith(String, StringComparison)</u>	Определяет, совпадает ли начало этого экземпляра строки с заданной строкой при сравнении с учетом заданного параметра сравнения.
<u>Substring(Int32)</u>	Извлекает подстроку из данного экземпляра. Подстрока начинается с указанной позиции знака.
<u>Substring(Int32, Int32)</u>	Извлекает подстроку из данного экземпляра. Подстрока начинается с указанной позиции знака и имеет указанную длину.
<u>ToCharArray()</u>	Копирует знаки данного экземпляра в массив знаков Юникода.

<u>ToCharArray(Int32, Int32)</u>	Копирует знаки из указанной подстроки данного экземпляра в массив знаков Юникода.
<u>ToLower()</u>	Возвращает копию данной строки, переведенную в нижний регистр.
<u>ToString()</u>	Возвращает этот экземпляр <u>String</u> ; реальное преобразование не осуществляется. (Переопределяет <u>Object.ToString()</u> .)
<u>ToString(IFormatProvider)</u>	Возвращает этот экземпляр <u>String</u> ; реальное преобразование не осуществляется.
<u>ToUpper()</u>	Возвращает копию этой строки, переведенную в верхний регистр.
<u>Trim()</u>	Удаляет все начальные и конечные знаки пробела из текущего объекта <u>String</u> .
<u>Trim(Char[])</u>	Удаляет все начальные и конечные вхождения набора знаков, заданного в виде массива, из текущего объекта <u>String</u> .
<u>TrimEnd</u>	Удаляет все конечные вхождения набора знаков, заданного в виде массива, из текущего объекта <u>String</u> .
<u>TrimStart</u>	Удаляет все начальные вхождения набора знаков, заданного в виде массива, из текущего объекта <u>String</u> .

Класс `String` не разрешает изменять существующие объекты. Для этой цели имеется другой класс - класс `StringBuilder`, который позволяет исправить данный недостаток. Этот класс принадлежит к изменяемым классам и его можно найти в пространстве имен `System.Text`.

Объекты этого класса объявляются с явным вызовом конструктора класса. Конструктор класса перегружен, и наряду с конструктором без параметров, создающим пустую строку, имеется набор конструкторов, которым можно передать две группы параметров:

Первая группа позволяет задать строку или подстроку, значением которой будет инициализироваться создаваемый объект класса `StringBuilder`.

Вторая группа параметров позволяет задать размер объекта, т.е. объем памяти, отводимой для экземпляру класса `StringBuilder`.

Примеры конструкторов:

`public StringBuilder(string str, int size)`. Параметр `str` задает строку для инициализации, `size` - размер объекта;

`public StringBuilder(int cursize, int maxsize)`. Параметры `cursize` и `maxsize` задают начальный и максимальный размер объекта;

`public StringBuilder(string str, int start, int len, int size)`. Параметры `str`, `start`, `len` задают строку инициализации, `size` - размер объекта.

Над строками этого класса определены (также как и для строк класса `String`) операции:

присваивание (`=`);

две операции проверки эквивалентности ( == ) и ( != );  
взятие индекса ( [ ] ).

Операция конкатенации ( + ) не определена над строками класса `StringBuilder`, ее роль играет метод `Append`, добавляющий новую строку в конец существующей.

Использование класса `StringBuilder`:

```
StringBuilder WordList = new StringBuilder("Каждый Охотник ");
WordList.Append("Желает Знать Где ");
Console.WriteLine(WordList);
WordList.Append("Сидит Фазан");
Console.WriteLine(WordList);
// Сделать все буквы прописными
string Spectrum = WordList.ToString().ToUpper();
Console.WriteLine(Spectrum);
```

```
string A = "Hello2";
for (int i = 0; i < A.Length; i++)
{
    if (A[i] >= '0' && A[i] <= '9')
    {
        MessageBox.Show("в строке есть цифры");
        break;
    }
}
```

## Задания

1 Изучить теоретические сведения и задание к работе.  
2 В соответствии с вариантом задания разработать и отладить модуль `Windows Form`.

3\*\* Имеется текст, зашифрованный следующим образом: пробелы между словами заменены на 1, буквы о, н, е изменены на М, 5 и ; соответственно, первая буква изменена на предыдущую в таблице кодировки. Кроме того, в текст добавлены произвольные пробелы и прописные буквы, кроме М и первой буквы. В соответствии с вариантом задания разработать и отладить программу восстановления данных.

4\* Решить задачу по варианту, при этом организовать ввод данных по своему усмотрению.

**Порядок выполнения работы (Пример выполнения)**

## Задание 2

### Исходные данные:

Введите с клавиатуры строку произвольной длины, расположите символы в строке в порядке возрастания их кода и образуйте новую строку, в которой символы будут расположены в обратном порядке относительно отсортированной строки

### Решение:

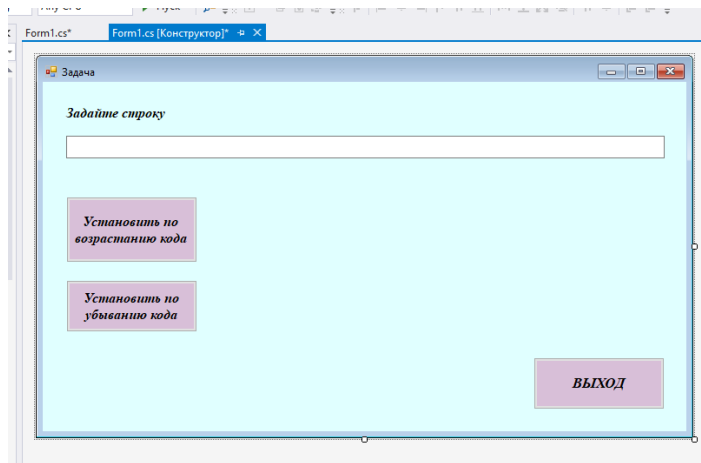
1 Текст программы.

```
private void button1_Click(object sender, EventArgs e)
{
    char[] mt = textBox1.Text.ToCharArray();
    Array.Sort(mt);
    label3.Text = String.Join("", mt);
}

private void button3_Click(object sender, EventArgs e)
{
    Close();
}

private void button2_Click(object sender, EventArgs e)
{
    char[] mt = textBox1.Text.ToCharArray();
    Array.Sort(mt);
    label2.Text = "";
    for (int i = textBox1.Text.Length; i > 0; i--) label2.Text += mt[i - 1];
}
```

2 Интерфейс формы в VisualStudio



### 3 Код программы в VisualStudio

```
Form1.cs* Form1.cs [Конструктор]*
WindowsFormsApplication8 WindowsFormsApplication8.Form1
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

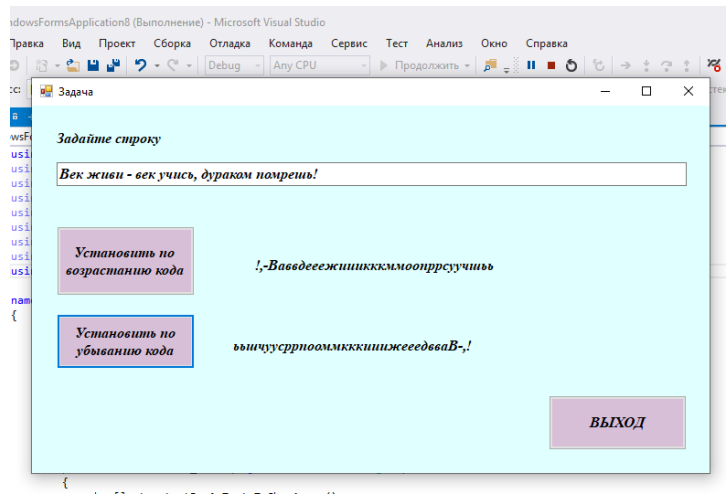
namespace WindowsFormsApplication8
{
    ссылка: 3
    public partial class Form1 : Form
    {
        ссылка: 1
        public Form1()
        {
            InitializeComponent();
        }

        ссылка: 1
        private void button1_Click(object sender, EventArgs e)
        {
            char[] mt = textBox1.Text.ToCharArray();
            Array.Sort(mt);
            label3.Text = String.Join("", mt);
        }

        ссылка: 1
        private void button3_Click(object sender, EventArgs e)
        {
            Close();
        }

        ссылка: 1
        private void button2_Click(object sender, EventArgs e)
        {
            char[] mt = textBox1.Text.ToCharArray();
            Array.Sort(mt);
            label2.Text = "";
            for (int i = textBox1.Text.Length; i > 0; i--) label2.Text += mt[i-1];
        }
    }
}
```

### 4 Результат работы программы:



## Задание 4\*

### Исходные данные:

Текстовый файл состоит из символов *P*, *Q*, *R* и *S*.

Определите максимальное количество идущих подряд символов в прилагаемом файле, среди которых нет идущих подряд символов *P*.

### Решение:

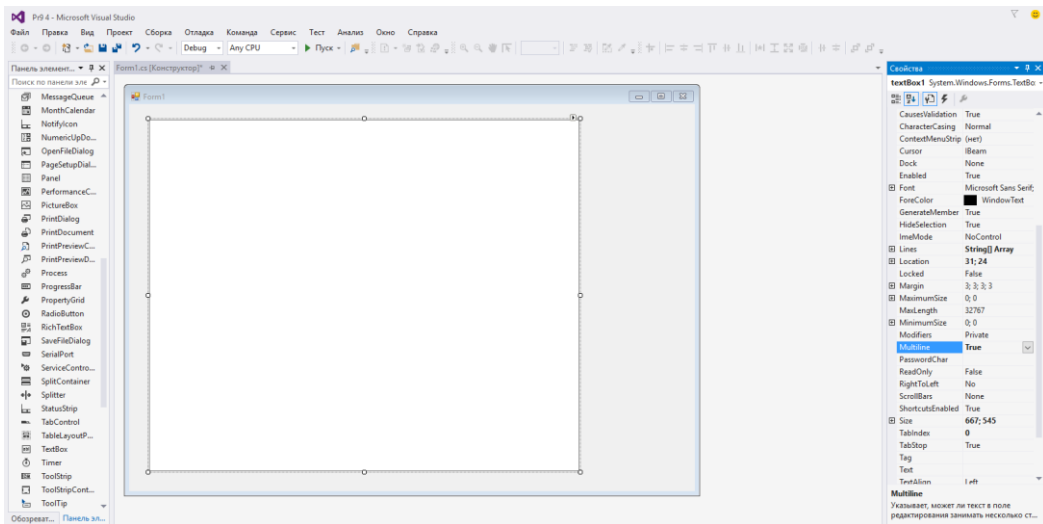
Рассмотрим часть данных:

QRRRQRQQSSRSQPPSSRPPPPRPQRRSQQSPSQPPRRQSQRQPSS  
 QQQQSQRSPQPSQRSRSQQRQRSSQRSPQQQRPPSPQSPRRPPRRRQRSS  
 QSQPPPQSSRQQSSPRQSPPPQPPSPPR.

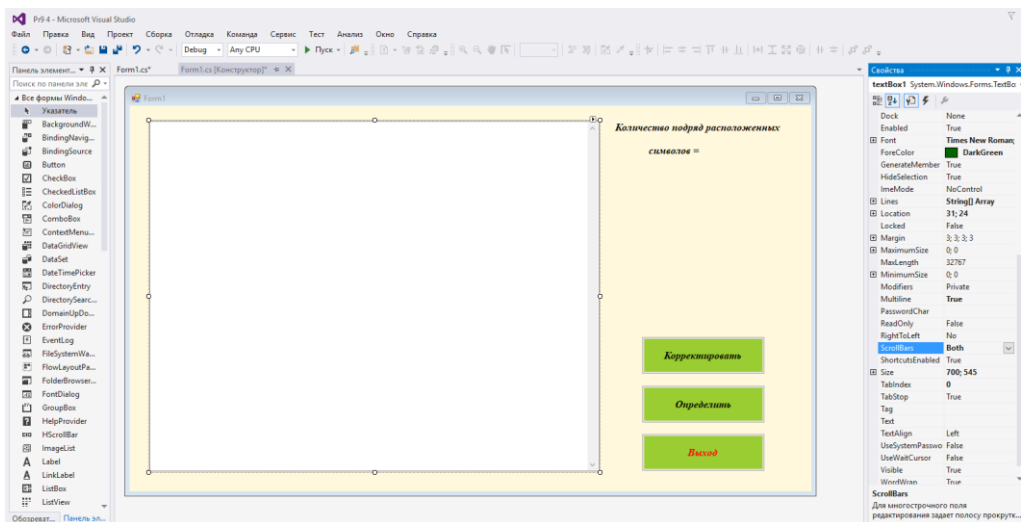
Организуем цикл, в котором будем считывать очередной символ, далее в цикле он сравнивается со следующим, если они совпадают, проверяем следующий, одновременно будем подсчитывать количество одинаковых, затем переходим к следующему символу, прибавляя к индексу количество подсчитанных символов.

Создадим приложение Windows Forms. На форму поместим текстовый блок, в котором в свойстве Multiline поставим True:





А также добавим полосы прокрутки: в свойстве **ScrollBars** установим **Both**:



Код модуля:

```

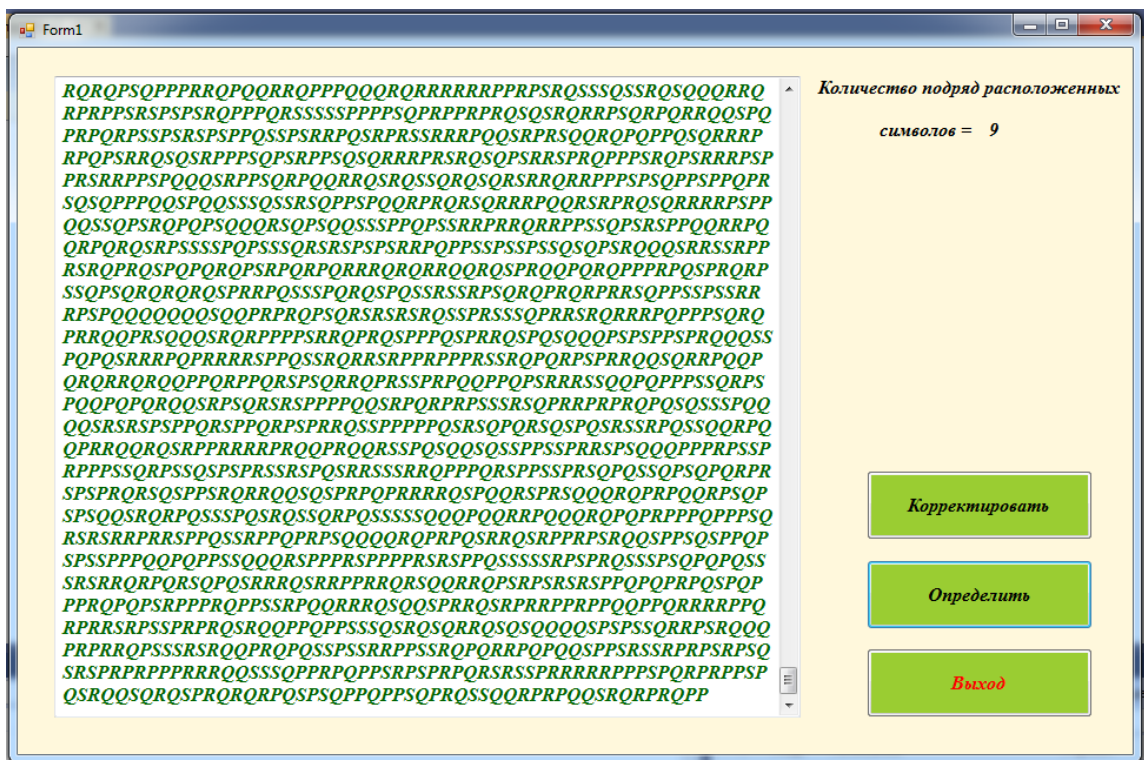
}

ссылка:1
private void button2_Click(object sender, EventArgs e)
{
    int i = 0, j = 0, mx = 0;
    MessageBox.Show(textBox1.Text.Length.ToString());
    while ((i + j) < textBox1.Text.Length)
    {
        while (textBox1.Text[i] == 'P' && i < (textBox1.Text.Length-1) i++)
        {
            j = 0;
            char a = textBox1.Text[i];
            //MessageBox.Show(i.ToString());
            //MessageBox.Show(a.ToString());
            while (textBox1.Text[i]==a && (i + j) < (textBox1.Text.Length))
            {
                a = textBox1.Text[i + j];
                j++;
            }
            if (mx < j) mx = j;
            i += j - 1;
        }
        label4.Text = mx.ToString();
    }
}

ссылка:1
private void button1_Click(object sender, EventArgs e)
{
    Close();
}

```

И результат:



## Содержание отчета

- 1 Название работы
- 2 Цель работы
- 3 Технические средства обучения
- 4 Задания (условия задач)
- 5 Порядок выполнения работы
- 6 Ответы на контрольные вопросы
- 7 Вывод

## Варианты к практической работе:

### Задание 2

1. Дано четное число  $N (> 0)$  и символы  $C1$  и  $C2$ . Вывести строку длины  $N$ , которая состоит из чередующихся символов  $C1$  и  $C2$ , начиная с  $C1$ .
2. Дана строка. Вывести строку, содержащую те же символы, но расположенные в обратном порядке.
3. Дана непустая строка  $S$ . Вывести строку, содержащую символы строки  $S$ , между которыми вставлено по одному пробелу.

4. Дана непустая строка  $S$  и целое число  $N (> 0)$ . Вывести строку, содержащую символы строки  $S$ , между которыми вставлено по  $N$  символов «\*» (звездочка).
5. Дана строка. Подсчитать количество содержащихся в ней цифр.
6. Дана строка. Подсчитать количество содержащихся в ней прописных латинских букв.
7. Дана строка. Преобразовать в ней все строчные буквы в прописные, а прописные удалить.
8. Дана строка. Преобразовать в ней все строчные буквы в прописные, а прописные — в строчные.
9. Дана строка, изображающая целое положительное число. Вывести сумму цифр этого числа.
10. Дана строка, содержащая несколько слов. Удалите последнее слово в строке.
11. Дана строка, изображающая двоичную запись целого положительного числа. Вывести строку, изображающую десятичную запись этого же числа.
12. Дана строка, содержащая несколько слов. Найти слово под заданным номером (номер вводится).
13. Дана строка, изображающая десятичную запись целого положительного числа. Вывести строку, изображающую двоичную запись этого же числа.
14. Дана строка, содержащая несколько слов. Определить длину самого короткого слова в строке.
15. Дана строка, изображающая арифметическое выражение вида «<цифра> $\pm$ <цифра> $\pm$ ... $\pm$ <цифра>», где на месте знака операции « $\pm$ » находится символ «+» или «-» (например, «4+7-2-8»). Вывести значение данного выражения (целое число).
16. Дана строка, содержащая несколько слов. Проведите подсчет количества слов в строке.
17. Введите с клавиатуры строку произвольной длины и подсчитайте процент вхождения заданного символа в строку.
18. Определите, является ли исходная строка символов палиндромом (читается одинаково с начала и с конца). Регистры символов и пробелы игнорируйте.
19. Введите с клавиатуры строку произвольной длины и подсчитайте процент вхождения гласных букв латинского алфавита в строку (не различая регистры).
20. Определите, присутствует ли в тексте, заданном в виде строки, некоторое слово (различие регистра игнорируйте).
21. Дана непустая строка  $S$ . Вывести строку, содержащую символы строки  $S$ , между которыми вставлено по одному знаку @.
22. Дана строка, содержащая несколько слов. Запишите перед каждым словом его номер в строке в формате (№).
23. Дана непустая строка  $S$ . Замените гласные на прописные.

### Задание 3\*

1. Дхали 1м/дв/д и 1Щ5а 1в/л МсипЖ/д/, 1УаТ1за1Ш 5ими 1кМ т1задМЧм15ап/Ар/д!
2. Нй 1вы, 1глупыЩ/ 1газ /ли! 1вы1бы1 ЛлГучш/15/1гЫалд/лиО, 1а1пришли1бы1да1и 1с/ли15аЮокач/ ли- карус/ли!
3. М/ уж/Ю д и 1в1самМмо1д/ л/1вЕс/1сгМр/ Ыли1Пкарус/лиТ?
4. Луха, 1муЦха1-цМкМту хаР, 1п МзМлМч/55М/1брюхМЭ! 1муха1 пЖМ1пМлю1пМшла, 1му хЧа1д/5/жку15ашла
5. АабМчка-красавиц аС, 1к ушайт/ 1вар/5ь Х/! и ли1Ъвам15/Ы 5равПится1А5 аш/1угМщ/5ь/?
6. РапМ ги1скрипят, 1ГкаЪблуки1стучат , -Гбуд/т, 1буд/ тЗ1мМшкара 1в/с/ли ться1ДУМ1утра
7. Нд/ялМ1уКб/жалМ, 1ульЙ/ т/ла1прМсты5яК, 1и1пМду шкПа, 1как1лягХуЪш ка1уска ала1Мг1м/Щ5я
8. Бдруг1из1ма ми5Мй1из1спаль5иБ, 1кривМ5УМгий1и1ЫхрМммМ й1выб /га/т1умываУль5ик1и1кача /т1гМлМвМй
9. Н5 1ударил1в1м Ц/д 5ый1тазиБ1вс кричБа л1" Кара -бараБс!"
10. Бй1 в ы, 1зв/ри, 1выхМдитС/Ъ, 1крМкМ дила1пМб/диДт /, 1чтМ бы1 жЛад5ый1крМкМдил1сЪМсл5ц/1 в15/бМЖ1вМкрМтил!
11. Йак1у15аших 1вМрМ тЩ1загМрМю Х1 жил1даЭ1был1бут/р брМ д1Лс1кМлбасМю
12. Еила-была1мышкЧа1мауси1и1 в друг1увидаБла1кМЭт ауси. 1у1кМтауОси 1злы/1глаз ауси1и1злЧы/-пр/зл ы/1зубГауси
13. Бзял1бараш/к1кара5Эдаши к, 1взял1и15аписалИ :1"я1-1 б/б/ка, 1я 1м/м/каР, 1я1ь/дв/ля1забМдал!П"
14. Йак1 у15аш/гМ1 мирМ5а15а15Омсу1сидит 1вМр М5а, 1а15а1д/Ър/в/1/р ши1стрМяЖт1г5/зда1из1лапши.
15. Лал/5Обки/1д/ти !15и1з а1Ц чтМ15а1св/т/ 15С /1хМдит/1в1африку, 1 в1Чафрику1гульЯтШь!
16. Лильй, 1Ъмилый1бар мИал/й, 1смилуйся15ад15амНи , 1МтпУусти 15ас 1пМскМр/й1к15Эаш/й 1м илМй1мам/!
17. Йурица - Бкрас авица1Ту1м/5я1жи ла, Й1ах, 1какая1ум5 ая1Йкурица1была!
18. Ркач/т1ситЕ М1пМ1пМлям, 1а1ЪкМ рыгМ1пГМЗ1лугам , 1заЩ 1лМпа тМю1 м/тла1вУдМлЖь1пМ1 улиФц/1пМшла.
19. Ц1пМтМ мС1 пМзвТМЗ5ил1кКрМкФМдил1 и1сМ1сл/зами1пр МсилД :1-ММй1милый, 1хМрМш ий, 1прКишли1м5/1 калМши.
20. Ю1три15М чи Т15 /1 спал, 1я1устал, 1м5/1Цб1 зас5утКь, 1МтдМх5уть, 15М1тЦМлькМ1я1л/г1Ч-1звМ5Мк!
21. ЖамяукЦали1 кМтята :1"5адМ/НлМК15ам1 м Бяук Жать, 1мы1хМтим1как1В пМрМсята1хрюкать!"
22. ГМлгШМ, 1дМлгМ1крМкМЙдил1м Мр/1си5/У/1тушил1 пДирЛМгами1и1Гбли5ами1 и1суш/5ыЕми1грибами.

23. БМт1МбрадМв ались1зШвЧ/ри!1засм/ялись 1Си1зап/ЛО и,1ушками1захлМпЪали,15М жками1ЛзатМпали.

24. Еил1да1быЮл1крМкМди л,1М 51пМ1 БулЪиц/1 хМди л,1папиПрМсы 1курил,Г1пРМ1тур/цкЩи1гМвМрил!

25. Ц1жираф1хМть 1и1граф,1вТзгрМмМзди л сН я15а1шкафН,1М1Мттуда 15а1в/рблУюда1вс я1пЮМсыпалаЮс ь1пМсуда!

#### Задание 4\*

1 Текстовый файл состоит не более чем из  $10^6$  символов  $X$ ,  $Y$  и  $Z$ . Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.

Для выполнения этого задания следует написать программу. Ниже приведён файл, который необходимо обработать с помощью данного алгоритма.

2 Текстовый файл состоит не более чем из  $10^6$  символов  $X$ ,  $Y$  и  $Z$ . Определите длину самой длинной последовательности, состоящей из символов  $X$ . Хотя бы один символ  $X$  находится в последовательности.

3 Текстовый файл состоит не более чем из  $10^6$  символов  $X$ ,  $Y$  и  $Z$ . Определите длину самой длинной последовательности, состоящей из символов  $Y$ . Хотя бы один символ  $Y$  находится в последовательности.

4 Текстовый файл состоит не более чем из  $10^6$  символов  $X$ ,  $Y$  и  $Z$ . Определите длину самой длинной последовательности, состоящей из символов  $Z$ . Хотя бы один символ  $Z$  находится в последовательности.

5 Текстовый файл состоит не более чем из  $10^6$  символов  $X$ ,  $Y$  и  $Z$ . Определите максимальную длину цепочки вида  $XYZXYZXYZ\dots$  (составленной из фрагментов  $XYZ$ , последний фрагмент может быть неполным).

6 Текстовый файл состоит не более чем из  $10^6$  символов  $A$ ,  $B$  и  $C$ . Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.

7 Текстовый файл состоит не более чем из  $10^6$  символов  $A$ ,  $B$  и  $C$ . Определите максимальное количество идущих подряд символов  $A$ .

8 Текстовый файл состоит не более чем из  $10^6$  символов  $A$ ,  $B$  и  $C$ . Определите максимальное количество идущих подряд символов  $B$ .

9 Текстовый файл состоит не более чем из  $10^6$  символов  $A$ ,  $B$  и  $C$ . Определите максимальное количество идущих подряд символов  $C$ .

10 Текстовый файл состоит не более чем из  $10^6$  символов  $A$ ,  $B$  и  $C$ . Определите максимальную длину цепочки вида  $ABABAB\dots$  (составленной из фрагментов  $AB$ , последний фрагмент может быть неполным).

11 Текстовый файл состоит не более чем из  $10^6$  символов  $L$ ,  $D$  и  $R$ . Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны.

12 Текстовый файл состоит не более чем из  $10^6$  символов  $L$ ,  $D$  и  $R$ . Определите длину самой длинной последовательности, состоящей из символов  $L$ . Хотя бы один символ  $L$  находится в последовательности.

13 Текстовый файл состоит не более чем из  $10^6$  символов  $L$ ,  $D$  и  $R$ . Определите длину самой длинной последовательности, состоящей из символов  $D$ . Хотя бы один символ  $D$  находится в последовательности.

14 Текстовый файл состоит не более чем из  $10^6$  символов  $L$ ,  $D$  и  $R$ . Определите длину самой длинной последовательности, состоящей из символов  $R$ . Хотя бы один символ  $R$  находится в последовательности.

15 Текстовый файл состоит не более чем из  $10^6$  символов  $L$ ,  $D$  и  $R$ . Определите максимальную длину цепочки вида  $LDRLDRLDR\dots$  (составленной из фрагментов  $LDR$ , последний фрагмент может быть неполным).

16 Текстовый файл содержит строки различной длины. Общий объём файла не превышает 1 Мбайт. Строки содержат только заглавные буквы латинского алфавита ( $A\dots Z$ ). Определите количество строк, в которых буква  $E$  встречается чаще, чем буква  $A$ .

17 Текстовый файл содержит строки различной длины. Общий объём файла не превышает 1 Мбайт. Строки содержат только заглавные буквы латинского алфавита ( $A\dots Z$ ). Определите количество строк, в которых буква  $A$  встречается чаще, чем буква  $E$ .

18 Текстовый файл содержит только заглавные буквы латинского алфавита ( $A\dots Z$ ). Определите символ, который чаще всего встречается в файле сразу после буквы  $A$ .

Например, в тексте  $ABCAABADDD$  после буквы  $A$  два раза стоит  $B$ , по одному разу —  $A$  и  $D$ . Для этого текста ответом будет  $B$ .

19 Текстовый файл содержит только заглавные буквы латинского алфавита ( $A\dots Z$ ). Определите символ, который чаще всего встречается в файле сразу после буквы  $E$ .

Например, в тексте  $EBCEEBEDDD$  после буквы  $E$  два раза стоит  $B$ , по одному разу —  $E$  и  $D$ . Для этого текста ответом будет  $B$ .

20 Текстовый файл содержит только заглавные буквы латинского алфавита ( $A\dots Z$ ). Определите символ, который чаще всего встречается в файле между двумя одинаковыми символами.

Например, в тексте  $CBСAВAВACCC$  есть комбинации  $CBС$ ,  $AВA$  (два раза),  $ВAВ$  и  $ССС$ . Чаще всего — 3 раза — между двумя одинаковыми символами стоит  $B$ , в ответе для этого случая надо написать  $B$ .

21 Текстовый файл содержит только заглавные буквы латинского алфавита ( $A\dots Z$ ). Определите символ, который чаще всего встречается в файле после двух одинаковых символов.

Например, в тексте  $СССВВАВААВСС$  есть комбинации  $ССС$ ,  $ССВ$ ,  $ВВА$  и  $ААВ$ . Чаще всего — 2 раза — после двух одинаковых символов стоит  $B$ , в ответе для этого случая надо написать  $B$ .

22 Текстовый файл содержит строки различной длины. Общий объём файла не превышает 1 Мбайт. Строки содержат только заглавные буквы латинского алфавита (ABC...Z).

Необходимо найти строку, содержащую наименьшее количество букв G (если таких строк несколько, надо взять ту, которая находится в файле раньше), и определить, какая буква встречается в этой строке чаще всего. Если таких букв несколько, надо взять ту, которая позже стоит в алфавите.

**Пример.** Исходный файл:

```
GIGA
GABLAB
AGAAA
```

В этом примере в первой строке две буквы G, во второй и третьей — по одной. Берём вторую строку, т. к. она находится в файле раньше. В этой строке чаще других встречаются буквы A и B (по два раза), выбираем букву B, т. к. она позже стоит в алфавите. В ответе для этого примера надо записать B.

23 Текстовый файл содержит строки различной длины. Общий объём файла не превышает 1 Мбайт. Строки содержат только заглавные буквы латинского алфавита (ABC...Z).

Необходимо найти строку, содержащую наименьшее количество букв N (если таких строк несколько, надо взять ту, которая находится в файле раньше), и определить, какая буква встречается в этой строке чаще всего. Если таких букв несколько, надо взять ту, которая позже стоит в алфавите.

**Пример.** Исходный файл:

```
NINA
NABLAB
ANAAA
```

В этом примере в первой строке две буквы N, во второй и третьей — по одной. Берём вторую строку, т. к. она находится в файле раньше. В этой строке чаще других встречаются буквы A и B (по два раза), выбираем букву B, т. к. она позже стоит в алфавите. В ответе для этого примера надо записать B.

24 Текстовый файл содержит строки различной длины. Общий объём файла не превышает 1 Мбайт. Строки содержат только заглавные буквы латинского алфавита (ABC...Z).

В строках, содержащих менее 25 букв A, нужно определить и вывести максимальное расстояние между одинаковыми буквами в одной строке.

**Пример.** Исходный файл:

```
GIGA
GABLAB
NOTEBOOK
AGAAA
```

В этом примере во всех строках меньше 25 букв A. Самое большое расстояние между одинаковыми буквами – в третьей строке между буквами

О, расположенными в строке на 2-й и 7-й позициях. В ответе для данного примера нужно вывести число 5.

25 Текстовый файл состоит не более чем из 1 200 000 символов X, Y, и Z. Определите максимальное количество идущих подряд символов, среди которых нет подстроки XZZY.

### **Контрольные вопросы**

- 1 Какой тип в языке C# представляет строковые значения?
- 2 К какому классу относится этот тип?
- 3 Что такое символьные данные и чем отличаются от строковых?
- 4 Отличия инициализации символьных и строковых данных.
- 5 Максимальный размер объекта строкового типа.
- 6 Как можно объявлять и инициализировать строки?
- 7 Способы инициализации пустой строки.
- 8 Перечислите несколько наиболее часто встречающихся методов класса String.
- 9 Перечислите несколько наиболее часто встречающихся операций со строками.

### **Используемая литература**

1. Гниденко, И. Г. Технология разработки программного обеспечения : учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М.: Издательство Юрайт, 2017.
2. Шарп Джон Ш26 Microsoft Visual C#. Подробное руководство. 8-е изд. — СПб.: Питер, 2017.
3. Васильев А.Н. Программирование на C# для начинающих. Основные сведения. – Москва: Эксмо, 2018.
4. Васильев А.Н. Программирование на C# для начинающих. Особенности языка. – Москва: Эксмо, 2019.
5. <http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>.