1. **«СТРОКИ. КЛАССЫ STRING И STRINGBUILDER»**

Задание 1. Дан фрагмент текста, запрашиваемый у пользователя. Написать программу, проверяющую, является ли он палиндромом, то есть читается ли одинаково слева направо и справа налево. При проверке регистр букв, знаки препинания и пробелы не учитываются. Например, строка «Аргентина манит негра.» — палиндром.

Листинг программы:

using System.ComponentModel.DataAnnotations;

namespace Zad\_1

{

internal class Program

{

static void Main(string[] args)

{

string text = "Аргентина манит негра";

char[] separators = new char[] { ' ', ';', ',', '\r', '\t', '\n' };

text = text.Replace(separators, "");

text = text.ToLower();

string message;

message = text == new string(text.Reverse().ToArray()) ? "Это палиндром" : "Это не палиндром";

Console.WriteLine(message);

}

}

public static class ExtensionMethods

{

public static string Replace(this string s, char[] separators, string newVal)

{

string[] temp;

temp = s.Split(separators, StringSplitOptions.RemoveEmptyEntries);

return string.Join(newVal, temp);

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Аргентина манит негра | Это палиндром |

Анализ результатов:



Рисунок 1.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Ввести текст, состоящий только из цифр и букв. Выяснить, верно ли, что сумма числовых значений цифр, находящихся в тексте, равна длине текста.

Листинг программы:

using System.Text.RegularExpressions;

namespace task2

{

internal class Program

{

static void Main(string[] args)

{

Console.Write("Введите текст: ");

string text = Console.ReadLine();

Regex regex = new Regex(@"\d");

int summ = 0;

MatchCollection matches = regex.Matches(text);

foreach (Match match in matches)

{

summ += int.Parse(match.Value);

}

string message = summ == text.Length ? "Сумма числовых значений цифр, находящихся в тексте, равна длине текста." : "Сумма числовых значений цифр, находящихся в тексте, не равна длине текста.";

Console.WriteLine(message);

}

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Последний день лета | Сумма числовых значений цифр, находящихся в тексте, не равна длине текста. |

Анализ результатов:



Рисунок 1.2 – Результат работы программы

Источник: собственная разработка

Задание 3. С клавиатуры вводится предложение. Результаты всех действий

вывести на экран.

 Поменять местами первое и последнее слова в предложении.

 Склеить второе и третье слова в предложении.

 Третье слово предложения вывести в обратном порядке

 В первом слове предложения вырезать первые две буквы.

Листинг программы:

using System.Text.RegularExpressions;

namespace task3

{

class Program

{

static void Swap(ref string[] matches)

{

string temp = matches[0];

matches[0] = matches[matches.Length - 1];

matches[matches.Length - 1] = temp;

}

static string CutAndConnect(ref string[] matches)

{

matches[1] = matches[1] + matches[2];

char[] charArray = matches[2].ToCharArray();

Array.Reverse(charArray);

matches[2] = new string(charArray);

matches[0] = matches[0].Substring(2);

return String.Join(" ", matches);

}

static void Main()

{

Console.WriteLine("Type a sentence");

string sentence = Console.ReadLine();

var matches = Regex.Matches(sentence, @"\w+[^\s]\*\w+|\w")

.Cast<Match>()

.Select(m => m.Value)

.ToArray();

Swap(ref matches);

string result = CutAndConnect(ref matches);

foreach (string match in matches)

{

Console.WriteLine(match);

}

Console.WriteLine("Result: " + result);

}

}

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Простой нрав не так уж и хорош | рош нравне ен так уж и Простой |

Анализ результатов:

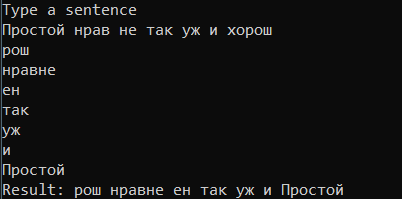


Рисунок 1.3 – Результат работы программы

Источник: собственная разработка

Задание 4. Известны фамилия, имя и отчество пользователя. Найти его код личности. Правило получения кода личности: каждой букве ставится в соответствие число - порядковый номер буквы в алфавите. Эти числа складываются. Если полученная сумма не является однозначным числом, то цифры числа снова складываются и так до тех пор, пока не будет получено однозначное число.

Листинг программы:

namespace task4

{

internal class Program

{

static void Main(string[] args)

{

Console.Write("Введите строку: ");

string input = Console.ReadLine();

string formated = FormatString(input);

int calculatedNumber = CalculateNumber(formated);

Console.WriteLine(calculatedNumber);

}

static string FormatString(string input)

{

input = input.ToLower();

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

{

if (input[i] == ' ')

{

input = input.Remove(i, 1);

}

}

return input;

}

static int CalculateNumber(string input)

{

const string alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя";

int sum = 0;

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

{

sum += alphabet.IndexOf(input[i]) + 1;

}

while (sum > 9)

{

int temp = sum % 10;

sum = sum / 10 + temp;

}

return sum;

}

}

}

Таблица 1.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Крупкевич Михаил Евгеньевич | 6 |

Анализ результатов:



Рисунок 1.4 – Результат работы программы

Источник: собственная разработка