Лабораторна робота №3  
Богуславський Альберт кн-922В

Варіант 1

Задача 1:

|  |  |
| --- | --- |
| **№ варіанту** | **Зміст завдання, виконуваного функцією** |
| 1 | Підраховує кількість слів в рядку |

Основні кроки роботи:

Введення тексту: Користувач вводить строку, яку програма обробляє.

Ігнорування пробілів: Програма пропускає всі пробільні символи (пробіли та табуляції) на початку або між словами.

Розпізнавання слів: Програма знаходить слова (послідовності символів, розділені пробілами) та підраховує їх кількість.

Виведення результату:

Виводиться кожне слово з його порядковим номером.

Після завершення виводиться загальна кількість слів.

Код програми:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task\_1

{

internal class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

Console.InputEncoding = System.Text.Encoding.Unicode;

Console.Write("Введіть строку: ");

string input = Console.ReadLine();

// Розділення рядка на слова, пропускаючи пробільні символи

string[] words = input.Split(new char[] { ' ', '\t' }, StringSplitOptions.RemoveEmptyEntries);

int wordCount = words.Length; // Кількість слів у рядку

Console.WriteLine("\nРезультат:");

// Виводимо слово та його номер у тексті

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

{

Console.WriteLine($"{words[i]} - {i + 1}");

}

Console.WriteLine($"\nКількість слів у тексті: {wordCount}");

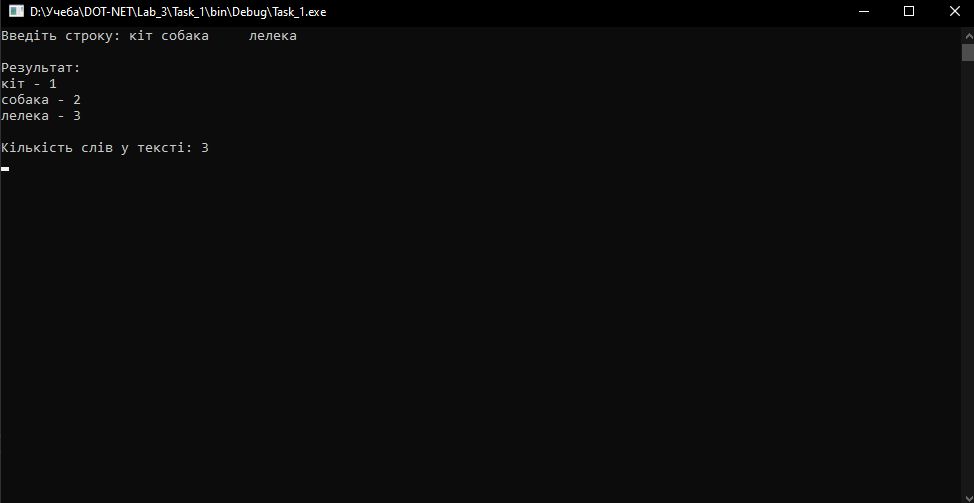
Console.ReadKey();

}

}

}

Виконання:



Задача 2

Виконати минуле завдання за варіантом із відліком з кінця, тобто 30 -1-й, 29-2-ий, 28-3-ій, і т.д.

Основні кроки роботи:

Зчитує текст: Користувач вводить рядок тексту.

Розбиває рядок на слова: Текст розділяється на окремі слова, ігноруючи пробіли та табуляції.

Обчислює кількість слів: Визначає кількість слів у введеному рядку.

Виводить слова у зворотному порядку: Для кожного слова у зворотному порядку виводить слово разом із його порядковим номером (від початку списку).

Виводить загальну кількість слів: Після обробки виводиться кількість слів у рядку.

Код програми:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task\_2

{

internal class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

Console.InputEncoding = System.Text.Encoding.Unicode;

Console.Write("Введіть строку: ");

string input = Console.ReadLine();

// Розділення рядка на слова, пропускаючи пробільні символи

string[] words = input.Split(new char[] { ' ', '\t' }, StringSplitOptions.RemoveEmptyEntries);

int wordCount = words.Length; // Кількість слів у рядку

Console.WriteLine("\nРезультат:");

// Виведення слів у зворотному порядку

for (int i = wordCount - 1; i >= 0; i--)

{

Console.WriteLine($"{words[i]} - {wordCount - i}");

}

Console.WriteLine($"\nКількість слів у тексті: {wordCount}");

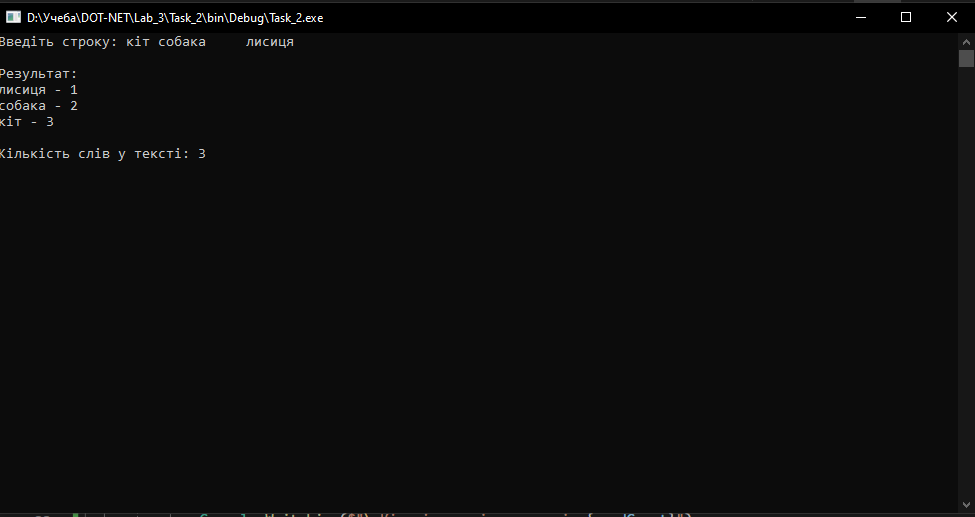
Console.ReadKey();

}

}

}

Виконання:



Задача 3

У текстовому рядку підрахувати кількість букв 'о';

Основні кроки роботи:

Читання рядка: Користувач вводить рядок тексту.

Розбиття на слова: Програма розбиває рядок на окремі слова, ігноруючи пробіли та табуляції.

Обчислення кількості символів 'о': Перевіряється кожен символ рядка на наявність літери 'о', ведеться підрахунок.

Виведення слів: Програма виводить усі слова, що були виділені з вхідного рядка.

Результати: Виводиться кількість символів 'о', знайдених у введеному рядку.

Код програми:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task\_3

{

internal class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

Console.InputEncoding = System.Text.Encoding.Unicode;

Console.Write("Введіть рядок: ");

string input = Console.ReadLine();

// Розділення рядка на слова, пропускаючи пробільні символи

string[] words = input.Split(new char[] { ' ', '\t' }, StringSplitOptions.RemoveEmptyEntries);

int wordCount = words.Length;

int a = 0; // Лічильник символів о

Console.WriteLine();

Console.Write("Введений рядок: ");

// Перевірка кожного символа на наявність о

foreach (char c in input)

{

if (c == 'о')

{

a++;

}

}

// Виведення всіх слів

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

{

Console.Write($"{words[i]} ");

}

Console.WriteLine();

Console.WriteLine($"Кількість літер о: {a}");

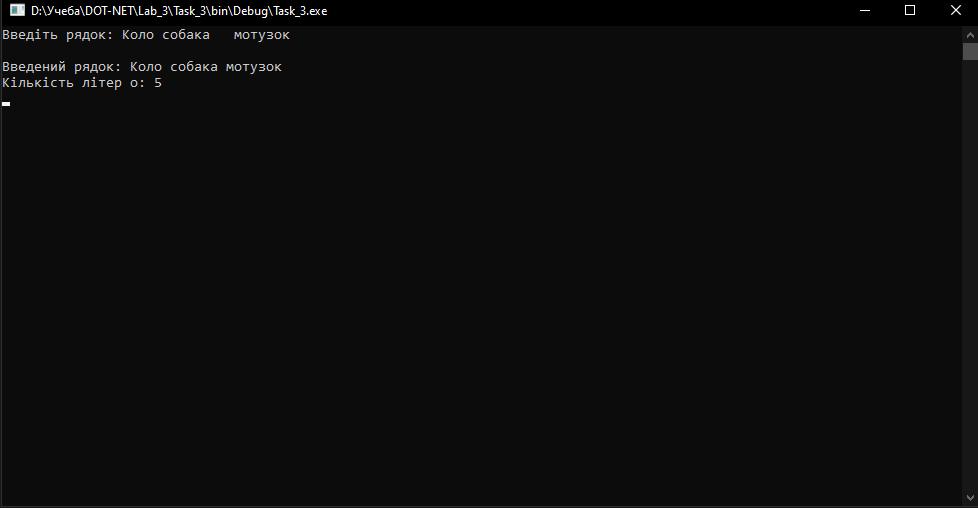
Console.ReadKey();

}

}

}

Виконання:



Завдання 4

Замінити в текстовому рядку кожну крапку трьома крапками

Основні кроки роботи:

Читання рядка від користувача: Програма запитує у користувача введення текстового рядка через консоль.

Заміна символів: Кожна крапка (.) у введеному рядку замінюється на три крапки (...) за допомогою методу string.Replace().

Розділення рядка на слова: Відформатований рядок розбивається на окремі слова. Розділення здійснюється за пробілами або табуляціями, при цьому порожні елементи ігноруються завдяки параметру StringSplitOptions.RemoveEmptyEntries.

Підрахунок кількості слів: Визначається кількість слів у рядку.

Виведення результату: Модифікований рядок (із замінами) виводиться у консоль з додатковим форматуванням — слова виводяться через пробіл.

Очікування завершення: Програма завершується після натискання клавіші.

Код програми:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task\_4

{

internal class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

Console.InputEncoding = System.Text.Encoding.Unicode;

Console.Write("Введіть рядок: ");

string input = Console.ReadLine();

// Заміна кожної крапки трьома крапками

string modifiedInput = input.Replace(".", "...");

// Розділення рядка на слова, пропускаючи пробільні символи

string[] words = modifiedInput.Split(new char[] { ' ', '\t' }, StringSplitOptions.RemoveEmptyEntries);

int wordCount = words.Length;

Console.WriteLine();

Console.Write("Введений рядок із замінами: ");

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

{

Console.Write($"{words[i]} ");

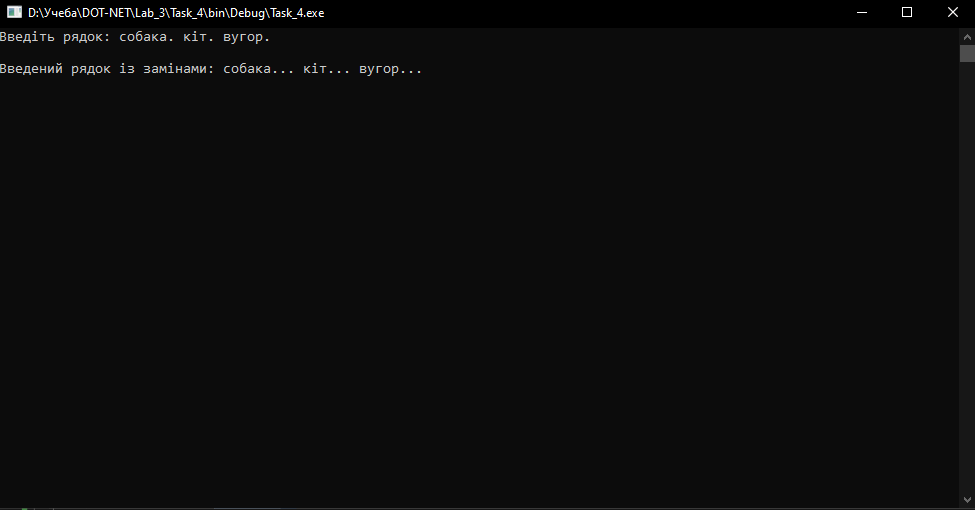
}

Console.ReadKey();

}

}

}



Завдання 5

Перетворити даний рядок переводячи першу букву кожного слова до верхнього регістру

Основні кроки роботи:

Введення рядка користувачем  
Програма запитує у користувача введення рядка тексту через консоль.

Обробка тексту

Розбиває рядок на слова, пропускаючи пробільні символи (пробіли та табуляції).

Кожне слово перетворюється так, щоб перша літера була великою, а решта — маленькими.

Виведення результату  
Виводить результат у вигляді рядка, де всі слова мають першу велику літеру.

Код програми:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task\_5

{

internal class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

Console.InputEncoding = System.Text.Encoding.Unicode;

Console.Write("Введіть рядок: ");

string input = Console.ReadLine();

// Розділення рядка на слова, пропускаючи пробільні символи

string[] words = input.Split(new char[] { ' ', '\t' }, StringSplitOptions.RemoveEmptyEntries);

// Перетворення перших букв кожного слова на великі

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

{

words[i] = char.ToUpper(words[i][0]) + words[i].Substring(1).ToLower();

}

int wordCount = words.Length;

Console.WriteLine();

Console.Write("Рядок з великими літерами: ");

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

{

Console.Write($"{words[i]} ");

}

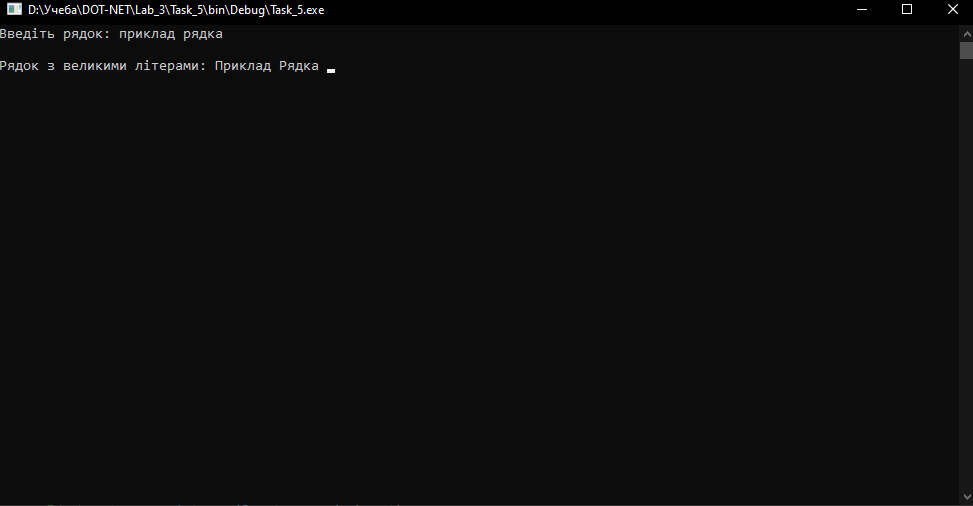
Console.ReadKey();

}

}

}

Виконання:



Завдання 6

Основні кроки роботи:

Видалення тексту в круглих дужках разом із самими дужками:

За допомогою регулярного виразу \\(.\*?\\) видаляється весь текст, розміщений у круглих дужках, включаючи самі дужки.

Розділення рядка на слова:

Рядок, очищений від тексту в дужках, розділяється на окремі слова. Пробільні символи, включаючи табуляцію, пропускаються.

Підрахунок кількості слів з непарною довжиною:

Програма обчислює кількість слів, довжина яких є непарним числом.

Підрахунок частоти входження літер:

Для кожної букви в рядку (незалежно від регістру) підраховується, скільки разів вона зустрічається. Цей підрахунок враховує лише літери, ігноруючи пробіли та інші символи.

Виведення результатів:

Очищений рядок (без тексту в дужках).

Кількість слів з непарною довжиною.

Частота входження кожної літери.

Код прогами:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task\_6

{

internal class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

Console.InputEncoding = System.Text.Encoding.Unicode;

Console.Write("Введіть рядок: ");

string input = Console.ReadLine();

// Видалення тексту в круглих дужках разом із самими дужками

string textWithoutParentheses = System.Text.RegularExpressions.Regex.Replace(input, "**\\(**.\*?**\\)**", "").Trim();

// Розділення рядка на слова, пропускаючи пробільні символи

string[] words = textWithoutParentheses.Split(new char[] { ' ', '\t' }, StringSplitOptions.RemoveEmptyEntries);

// Підрахунок слів з непарною довжиною

int oddLengthWordCount = words.Count(word => word.Length % 2 != 0);

// Підрахунок частоти входження кожної літери

Dictionary<char, int> Frequency = new Dictionary<char, int>();

foreach (char c in textWithoutParentheses.Replace(" ", ""))

{

if (char.IsLetter(c))

{

char lowerChar = char.ToLower(c);

if (Frequency.ContainsKey(lowerChar))

{

Frequency[lowerChar]++;

}

else

{

Frequency[lowerChar] = 1;

}

}

}

Console.WriteLine();

// Виведення всіх слів

Console.WriteLine("Введений рядок без тексту в дужках: " + textWithoutParentheses);

// Виведення кількості слів з непарною довжиною

Console.WriteLine("Кількість слів з непарною довжиною: " + oddLengthWordCount);

// Виведення частоти входження кожної літери

Console.WriteLine("Частота входження кожної літери:");

foreach (var entry in Frequency)

{

Console.WriteLine($"Літера '{entry.Key}': {entry.Value}");

}

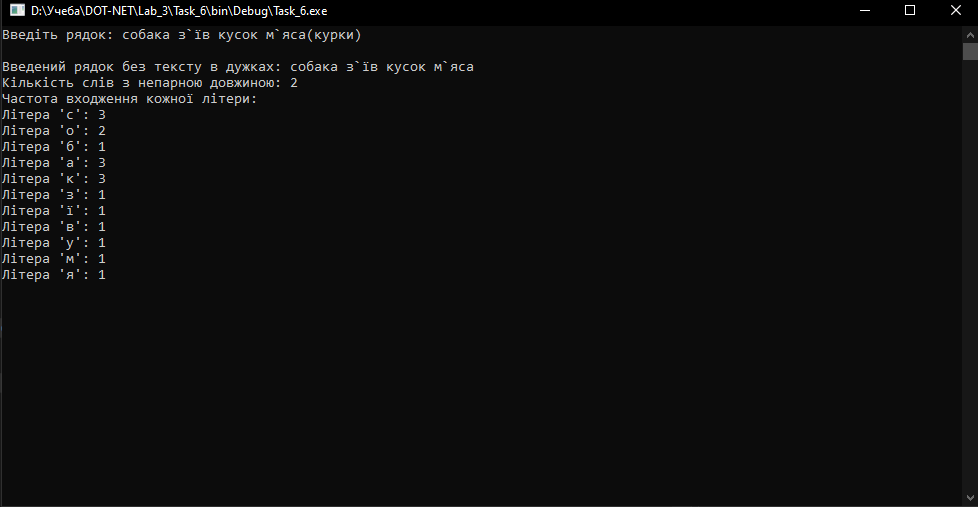
Console.ReadLine();

}

}

}

Виконання:



Завдання 7.1

Модифікувати перше завдання цієї роботи таким чином, щоб введення початкового рядка чи рядків та виведення результату обробки цього рядка чи рядків були реалізовані за допомогою технологій використання текстових файлів у мові C#.

Основні кроки роботи:

Читає перший рядок із вхідного файлу.

Якщо файл порожній або не містить даних, виводить повідомлення та завершує виконання.

Розділяє рядок на окремі слова, ігноруючи пробіли та табуляції.

Рахує кількість слів.

Виводить кожне слово з його порядковим номером у тексті.

Додає загальну кількість слів до результату.

Код програми:

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task\_7.\_1

{

internal class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

Console.InputEncoding = System.Text.Encoding.Unicode;

string inputFilePath = "..\\..\\..\\txt\\text.txt";

string outputFilePath = "..\\..\\..\\txt\\7.1.txt";

try

{

using (StreamReader sr = new StreamReader(inputFilePath))

using (StreamWriter sw = new StreamWriter(outputFilePath))

{

string input = sr.ReadLine();

if (input == null)

{

Console.WriteLine("Файл порожній або не містить даних.");

return;

}

// Розділення рядка на слова, пропускаючи пробільні символи

string[] words = input.Split(new char[] { ' ', '\t' }, StringSplitOptions.RemoveEmptyEntries);

int wordCount = words.Length; // Кількість слів у рядку

sw.WriteLine("\nРезультат:");

// Виводимо слово та його номер у тексті

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

{

sw.WriteLine($"{words[i]} - {i + 1}");

}

sw.WriteLine($"\nКількість слів у тексті: {wordCount}");

}

}

catch (Exception e)

{

Console.WriteLine("Опис: " + e.Message);

}

finally

{

Console.WriteLine("Стоп.");

}

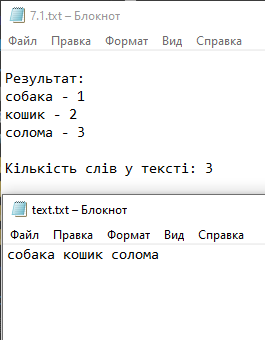
Console.ReadKey();

}

}

}

Виконання:



Завдання 7.2

Модифікувати друге завдання цієї роботи таким чином, щоб введення початкового рядка чи рядків та виведення результату обробки цього рядка чи рядків були реалізовані за допомогою технологій використання текстових файлів у мові C#.

Основні кроки роботи:

Читає перший рядок із вхідного файлу.

Якщо файл порожній або не містить даних, виводить повідомлення та завершує виконання.

Розділяє рядок на окремі слова, ігноруючи пробіли та табуляції.

Рахує кількість слів.

Виводить кожне слово в обратному порядку.

Додає загальну кількість слів до результату.

Код програми:

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task\_7.\_1

{

internal class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

Console.InputEncoding = System.Text.Encoding.Unicode;

string inputFilePath = "..\\..\\..\\txt\\text.txt";

string outputFilePath = "..\\..\\..\\txt\\7.2.txt";

try

{

using (StreamReader sr = new StreamReader(inputFilePath))

using (StreamWriter sw = new StreamWriter(outputFilePath))

{

string input = sr.ReadLine();

if (input == null)

{

Console.WriteLine("Файл порожній або не містить даних.");

return;

}

// Розділення рядка на слова, пропускаючи пробільні символи

string[] words = input.Split(new char[] { ' ', '\t' }, StringSplitOptions.RemoveEmptyEntries);

int wordCount = words.Length; // Кількість слів у рядку

sw.WriteLine("\nРезультат:");

// Виводимо слово та його номер у тексті

for (int i = wordCount - 1; i >= 0; i--)

{

sw.WriteLine($"{words[i]} - {wordCount - i}");

}

sw.WriteLine($"\nКількість слів у тексті: {wordCount}");

}

}

catch (Exception e)

{

Console.WriteLine("Опис: " + e.Message);

}

finally

{

Console.WriteLine("Стоп.");

}

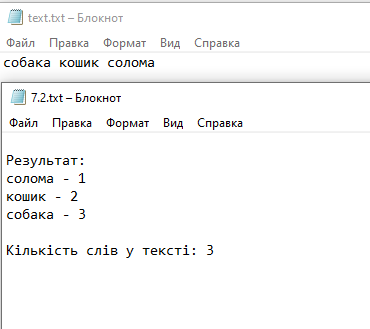
Console.ReadKey();

}

}

}

Виконання:



Завдання 7.3

Модифікувати третє завдання цієї роботи таким чином, щоб введення початкового рядка чи рядків та виведення результату обробки цього рядка чи рядків були реалізовані за допомогою технологій використання текстових файлів у мові C#.

Основні кроки роботи:

Читає перший рядок із вхідного файлу.

Якщо файл порожній або не містить даних, виводить повідомлення та завершує виконання.

Розділяє рядок на окремі слова, ігноруючи пробіли та табуляції.

Рахує кількість літер о.

Виводить кожне слово в рядку.

Додає загальну кількість літер о до результату.

Код програми:

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task\_7.\_3

{

internal class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

Console.InputEncoding = System.Text.Encoding.Unicode;

string inputFilePath = "..\\..\\..\\txt\\text.txt";

string outputFilePath = "..\\..\\..\\txt\\7.3.txt";

try

{

using (StreamReader sr = new StreamReader(inputFilePath))

using (StreamWriter sw = new StreamWriter(outputFilePath))

{

string input = sr.ReadLine();

// Розділення рядка на слова, пропускаючи пробільні символи

string[] words = input.Split(new char[] { ' ', '\t' }, StringSplitOptions.RemoveEmptyEntries);

int wordCount = words.Length;

int a = 0; // Лічильник символів о

sw.WriteLine();

sw.Write("Введений рядок: ");

// Перевірка кожного символа на наявність о

foreach (char c in input)

{

if (c == 'о')

{

a++;

}

}

// Виведення всіх слів

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

{

sw.Write($"{words[i]} ");

}

sw.WriteLine();

sw.WriteLine($"Кількість літер о: {a}");

}

}

catch (Exception e)

{

Console.WriteLine("Опис: " + e.Message);

}

finally

{

Console.WriteLine("Стоп.");

}

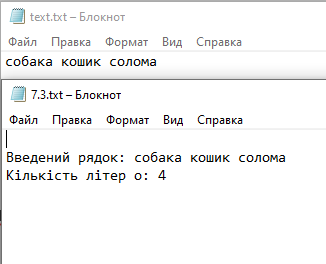
Console.ReadKey();

}

}

}

Виконання:



Завдання 8.1

Модифікувати четверте завдання цієї роботи таким чином, щоб виведення результату обробки рядків були реалізовані за допомогою технологій використання двійкових файлів у мові C# із можливістю перевірки коректності результатів шляхом виведення контенту двійкового файлу на екран консолі.

Основні кроки роботи:  
Користувач вводить рядок тексту в консоль.

Програма заміняє всі крапки (".") на три крапки ("...").

Відредагований рядок розділяється на окремі слова, і всі слова записуються в бінарний файл.

Після запису програма відкриває той самий бінарний файл для читання.

Вміст файлу виводиться на консоль.

Код прогами:

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task\_8.\_1

{

internal class Program

{

static void Main(string[] args)

{

string outputDatPath = "..\\..\\..\\dat\\8.1.dat";

// Записуємо в бінарний файл

using (BinaryWriter writer = new BinaryWriter(File.Open(outputDatPath, FileMode.OpenOrCreate)))

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

Console.InputEncoding = System.Text.Encoding.Unicode;

Console.Write("Введіть рядок: ");

string input = Console.ReadLine();

// Заміна кожної крапки трьома крапками

string modifiedInput = input.Replace(".", "...");

// Розділення рядка на слова

string[] words = modifiedInput.Split(new char[] { ' ', '\t' }, StringSplitOptions.RemoveEmptyEntries);

int wordCount = words.Length;

writer.Write("\n");

writer.Write("Введений рядок із замінами: ");

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

{

writer.Write($"{words[i]} ");

}

}

// читаємо з того ж файлу

using (BinaryReader reader = new BinaryReader(File.Open(outputDatPath, FileMode.Open)))

{

// виводимо вміст

while (reader.BaseStream.Position < reader.BaseStream.Length)

{

string line = reader.ReadString();

Console.Write(line);

}

}

Console.ReadKey();

}

}

}

Виконання:



Завдання 8.2

Модифікувати п`яте завдання цієї роботи таким чином, щоб виведення результату обробки рядків були реалізовані за допомогою технологій використання двійкових файлів у мові C# із можливістю перевірки коректності результатів шляхом виведення контенту двійкового файлу на екран консолі.\

Основні кроки роботи:  
Користувач вводить рядок тексту в консоль.

Програма заміняє всі маленькі перші букви слів в рядку

Відредагований рядок розділяється на окремі слова, і всі слова записуються в бінарний файл.

Після запису програма відкриває той самий бінарний файл для читання.

Вміст файлу виводиться на консоль.

Код програми:

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task\_8.\_2

{

internal class Program

{

static void Main(string[] args)

{

string outputDatPath = "..\\..\\..\\dat\\8.2.dat";

using (BinaryWriter writer = new BinaryWriter(File.Open(outputDatPath, FileMode.OpenOrCreate)))

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

Console.InputEncoding = System.Text.Encoding.Unicode;

Console.Write("Введіть рядок: ");

string input = Console.ReadLine();

// Розділення рядка на слова, пропускаючи пробільні символи

string[] words = input.Split(new char[] { ' ', '\t' }, StringSplitOptions.RemoveEmptyEntries);

// Перетворення перших букв кожного слова на великі

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

{

words[i] = char.ToUpper(words[i][0]) + words[i].Substring(1).ToLower();

}

int wordCount = words.Length;

writer.Write("\n");

writer.Write("Рядок з великими літерами: ");

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

{

writer.Write($"{words[i]} ");

}

}

using (BinaryReader reader = new BinaryReader(File.Open(outputDatPath, FileMode.Open)))

{

// виводимо вміст

while (reader.BaseStream.Position < reader.BaseStream.Length)

{

string line = reader.ReadString();

Console.Write(line);

}

}

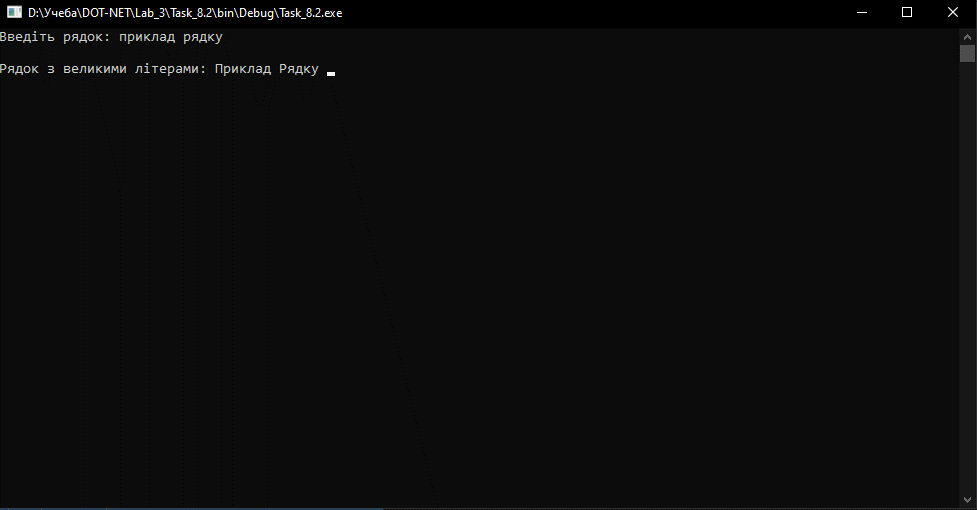
Console.ReadLine();

}

}

}

Виконання:



Висновок: набули навички в написанні програм з використанням рядків та застосування файлових операцій.