Міністерство освіти України

Національний технічний університет "ХПІ"

кафедра “ННІКНІТ"

**Звіт**

**Лабораторна робота 2**

з дисципліни "Програмна технологія .NET"

Виконав: студент групи КН-921Г

Верещак А. Г.

Перевірив:

Івашко А.В.

Харків 2023

**Зміст**

[Завдання 3](#__RefHeading___Toc2388_3286642050)

[Робота з рядками та файлове введення/виведення у мові C# 5](#__RefHeading___Toc2390_3286642050)

[Хід роботи 6](#__RefHeading___Toc2392_3286642050)

[Завдання 1: 6](#__RefHeading___Toc2239_3286642050)

[Завдання 2: 7](#__RefHeading___Toc2241_3286642050)

[Завдання 3: 8](#__RefHeading___Toc2243_3286642050)

[Завдання 4: 9](#__RefHeading___Toc2245_3286642050)

[Завдання 5: 10](#__RefHeading___Toc2247_3286642050)

[Завдання 6: 11](#__RefHeading___Toc2249_3286642050)

[Завдання 7: 12](#__RefHeading___Toc2251_3286642050)

[Завдання 8: 13](#__RefHeading___Toc2253_3286642050)

[Висновок 14](#__RefHeading___Toc2394_3286642050)

Лабораторна робота №3. **Робота із рядками та файлове введення / виведення у мові C# .**

**Мета роботи**: набути навичок в написанні програм з використанням рядків та застосування файлових операцій.

# Завдання

1. Використовуючи завдання власного варіанта до лабораторної роботи 12 з електронних методичних вказівок з основ програмування та алгоритмічних мов створити проект з реалізації цих завдань засобами мови C#.
2. Виконати минуле завдання за варіантом із відліком з кінця, тобто 30 -1-й, 29- 2-ий, 28-3-ій, і т.д.
3. Виконати завдання згідно із номером вашого варіанту: Визначити позиції входження у текстовий рядок букви 'С';
4. Виконати завдання згідно із номером вашого варіанту: В текстовому рядку замінити пробіли двома пробілами;
5. Виконати завдання згідно із номером вашого варіанту: Визначити кількість слів у рядку, які починаються на українські   
   голосні літери;
6. З клавіатури вводиться текстовий рядок. Розробити програму, яка реалізує вказані дії: підраховує кількість слів у тексті;  
   та виводить на екран слово, що містить найбільшу кількість літер
7. Модифікувати перші три 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#.
8. Модифікувати четверте і 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.

# Робота з рядками та файлове введення/виведення у мові C#

**Робота з рядками (Strings):**

* **Рядок (String)** в мові C# є послідовністю символів і є незмінним (immutable). Це означає, що після створення рядка ви не можете змінити його вміст без створення нового рядка.
* **Створення рядків**: Рядки можна створювати за допомогою літералів у подвійних лапках або за допомогою конструктора string.
* **Операції над рядками**: Мова C# має багато методів для роботи з рядками, таких як отримання підстроки, знаходження позиції підстроки, заміна підстроки, конвертація рядків до верхнього або нижнього регістру тощо.

**Файлове введення/виведення:**

* **Файловий ввід/вивід (File I/O)** дозволяє програмі зчитувати дані з файлів і записувати дані до файлів.
* Для читання з файлу в C# використовують класи System.IO.File або System.IO.StreamReader. Клас File дозволяє зчитати весь файл як одну строку або вміст файла у вигляді байтів.
* Для запису в файл в C# використовують класи System.IO.File або System.IO.StreamWriter. Клас File дозволяє записати строку або байти у файл.
* Для безпечного відкриття і закриття файлів можна використовувати ключове слово using, що дозволяє автоматично закривати файл після завершення роботи з ним.
* Обробка помилок при роботі з файлами важлива, оскільки файли можуть бути недоступні, або виникати інші помилки під час введення/виведення.

Загальний підхід до роботи з рядками та файлами в C# полягає в використанні стандартних бібліотек, таких як System.IO, і дотриманні найкращих практик, таких як обробка помилок та закриття файлових ресурсів після завершення роботи з ними

# Хід роботи

# Завдання 1:

Створюємо новий проект Task\_1.

Для вирішення поставленого завдання пишемо наступний код:

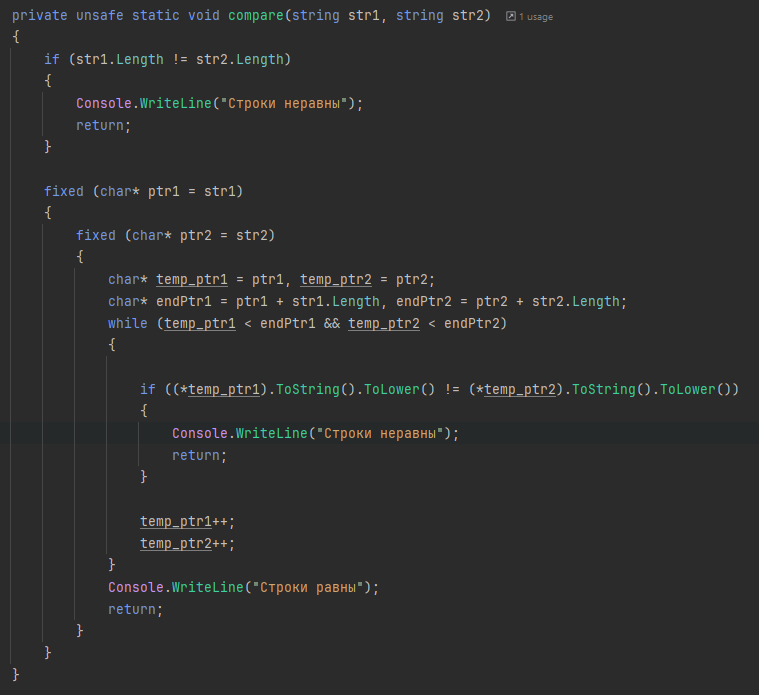


Рисунок 3.1 — Фрагмент коду Task\_1

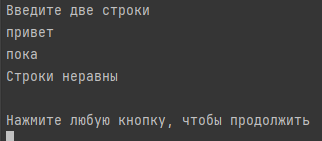


Рисунок 3.2 — Робота програми Task\_1

# Завдання 2:

Створюємо новий проект Task\_2.

Для вирішення поставленого завдання пишемо код:

# 

Рисунок 3.3 — Фрагмент коду Task\_2

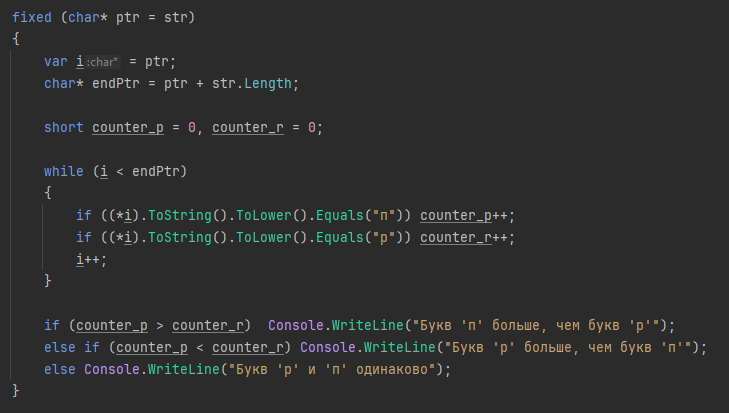
# 

Рисунок 3.4 — Робота коду Task\_2

# Завдання 3:

Створюємо новий проект Task\_3.

Для вирішення поставленого завдання пишемо код:

Рисунок 3.5 — Фрагмент коду Task\_3

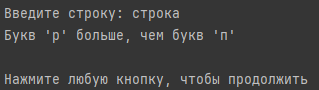


Рисунок 3.6 — Робота коду Task\_3

# Завдання 4:

Створюємо новий проект Task\_4.

Для вирішення поставленого завдання пишемо наступний код:

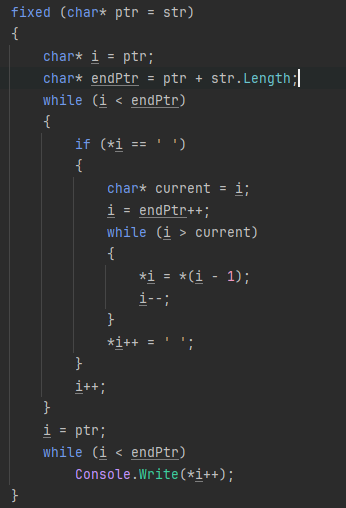


Рисунок 3.7 — Фрагмент коду Task\_4

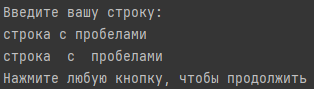


Рисунок 3.8 — Робота коду Task\_4

# Завдання 5:

Створюємо новий проект Task\_5.

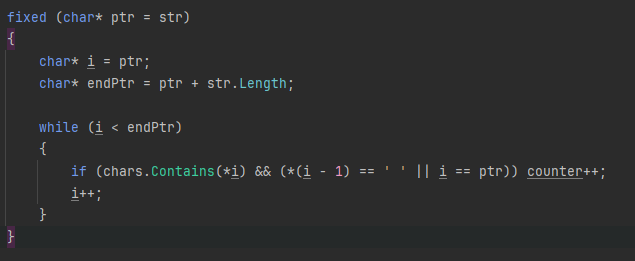
Для вирішення поставленого завдання пишемо наступний код:

Рисунок 3.9 — Код програми Task\_5

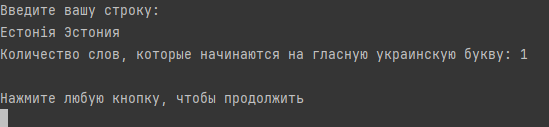


Рисунок 3.10 — Робота коду Task\_5

# Завдання 6:

Створюємо новий проект Task\_6.

Для вирішення поставленого завдання пишемо наступний код:

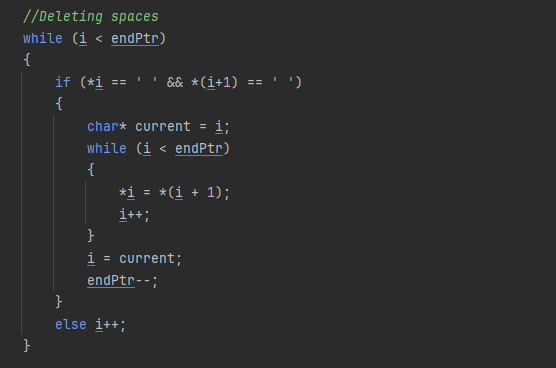


Рисунок 3.11 — Фрагмент коду Task\_6

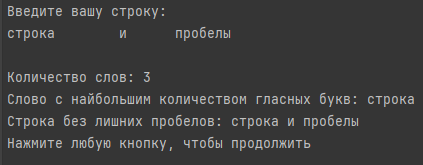


Рисунок 3.12 — Робота коду Task\_6

# Завдання 7:

Створюємо новий проект Task\_7.

Для вирішення поставленого завдання пишемо наступний код:

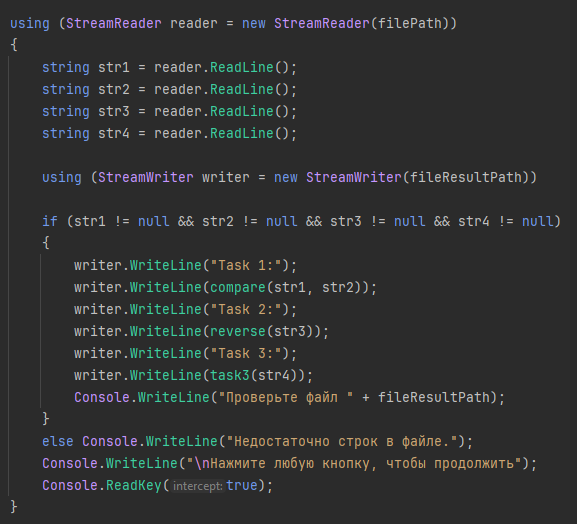
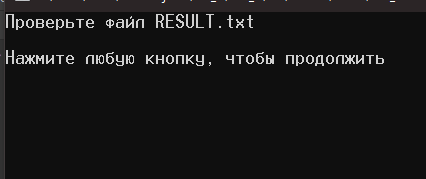


Рисунок 3.13 — Фрагмент коду Task\_7



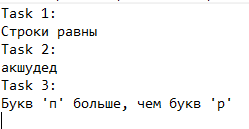


Рисунок 3.14 — Робота коду Task\_7

# Завдання 8:

Створюємо новий проект Task\_8.

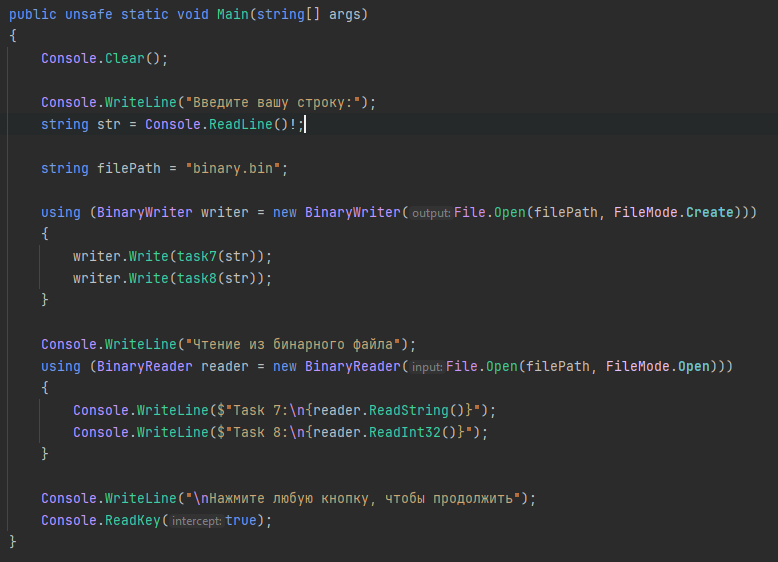
Для вирішення поставленого завдання пишемо наступний код:

Рисунок 3.15 — Фрагмент коду Task\_8

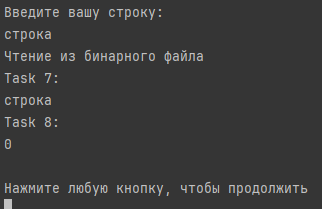


Рисунок 3.16 — Робота коду Task\_8

# Висновок

В ході розгляду різних програмних завдань у мові програмування C#, було продемонстровано роботу із рядками та файловим введенням і виведенням. Важливо відзначити, що робота із рядками та файлами є ключовою складовою для багатьох програм, оскільки дозволяє зчитувати та обробляти інформацію з різних джерел і зберігати результати для подальшого використання. Основні відомості та вміння, отримані під час вивчення цих концепцій, можуть бути корисними у великому спектрі задач і додатків, від опрацювання текстових даних до роботи із структурованими даними у форматі файлів.

У висновку, робота із рядками включає в себе методи обробки та маніпулювання текстовими даними, такі як розділення, пошук та заміна. Файлове введення і виведення надає можливість зчитувати дані з файлів та записувати результати в файли. Розуміння цих концепцій є важливим для розробки програм, які взаємодіють із зовнішніми джерелами даних та забезпечують збереження та обробку інформації для подальшого використання.  
У висновку хотів би рекомендувати кілька корисних статей та літератури, які можуть допомогти глибше розібратися у роботі із рядками та файловим введенням/виведенням в мові C#: