Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

# ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 10

з навчальної дисципліни “Базові методології та технології програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ ОБРОБЛЕННЯ ДАНИХ СКЛАДОВИХ ТИПІВ З ФАЙЛОВИМ ВВЕДЕННЯМ/ВИВЕДЕННЯМ

ВИКОНАВ

студент академічної групи

КІ-21-1 Ожеховський Владислав

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_  П.С. Усік

Кропивницький – 2022

# Лабораторна робота №10

**Тема** Реалізація програмних модулів оброблення даних складових типів з файловим введенням/виведенням.

**Мета роботи:** полягає у набутті ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою програмування С++ програмних модулів створення й оброблення даних типів масив, структура, об’єднання, множина, перелік, перетворення типів даних, використання файлових потоків та функцій стандартних бібліотек для оброблення символьної інформації.

## Варіант 3

**Завдання:**

1. Реалізувати програмні модулі розв’язування задач 10.1–10.3 як складові статичної бібліотеки libModulesПрізвище.а (проект ModulesПрізвище лабораторних робіт №8–9).
2. Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв’язування задач 10.1–10.3.

# ХІД РОБОТИ

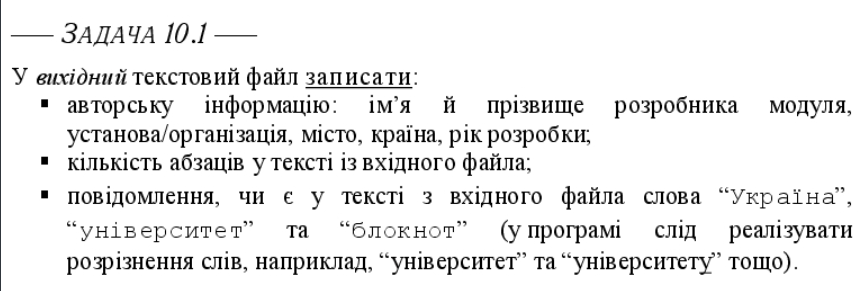


Рисунок 1 - Умова задачі 10.1

### **Строга постановка задачі:**

Вхідні дані: текст в форматі string записане до вхідного файлу.

Вихідні дані: записані до вихідного файлу: авторська інформація, кількість абзаців (ціле, невід`ємне число), результат пошуку слів в тексті.

### **Проектування програмного модуля:**

**authorInfo()** - приймає в якості аргументу назву вихідного файлу. Виконується очищення вмісту файлу та запис авторської інформації.  
**numParagraphsInFile ()** - отримує на вхід назву вихідного файлу та текст, що міститься у вхідному файлі. Цикл ітерації якого дорівнює довжині тексту, реалізовано для перевірки наявності символу переносу рядка. На вихід отримуємо дозапис у вихідний файл кількості абзаців в тексті.

**findWordInText ()** - отримує на вхід назву вихідного файлу та текст, що міститься у вхідному файлі. Текст ініціалізований як рядок, котрий надалі використовується в функції бібліотеки <сstring> для пошуку вхідного слова. Істинність пошуку визначається співпадінням позиції каретки у файлі с позицією результату стандартної функції find(). До вихідного файлу записується результат пошуку.

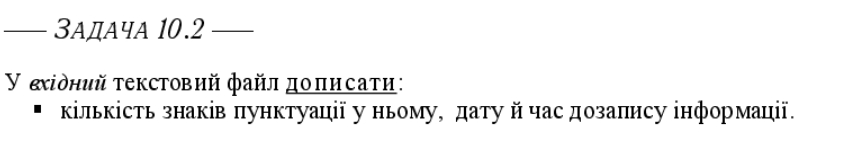


Рисунок 2 - Умова задачі 10.2

### **Строга постановка задачі:**

Вхідні дані: текст в форматі string записане до вхідного файлу .

Вихідні дані: записані до вихідного файлу: кількість знаків пунктуації (ціле, невід` ємне число), дата та час дозапису.

**Проектування програмного модуля:**

**numPunctuationMarksInFile ()** - приймає на вхід назву вхідного файлу та текст, що міститься в ньому. Спираючись на реалізацію функції пошуку символа переносу рядка, повторюємо цикл з **numParagraphsInFile()**, проте на кожній ітерації перевіряємо чи даний символ дорівнює одному з символів в масиві зі знаками пунктуації. У вхідний файл ведеться дозапис кількості знаків пунктуації в ньому.

**timestampInFile()** - приймає аргумент назву вхідного файлу. В даний файл записується дата та час дозапису інформації за допомогою стандартної функції з бібліотеки <ctime>.

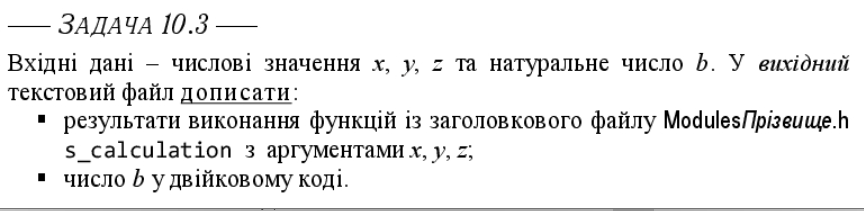


Рисунок 3 - Умова задачі 10.3

### **Строга постановка задачі:**

Вхідні та вихідні дані вичерпно описані в умові завдання.

### **Проектування програмного модуля:**

**sCalculationResInFile()** - аргументи: назва вихідного файлу та три дійсних числа. До вихідного файлу записується результат виконання функції s\_caculation, що приймає в якості аргументів три дійсні значення.

**decimalToBinary()** - вхідні дані: назва вихідного файлу та натуральне число. Оголошується масив розміром в 32 елементи (стільки бітів вміщує змінна типу int) і в циклі, з використанням оператору ‘%’ заповнюємо массив остачами від ділення на 2. В наступному циклі виводимо поелементно, починаючи з останнього, вміст масиву до вихідного файлу.

### **Результат виконання TestDriver.exe:**

Запис тексту до вхідного файлу: true

10.1(1) - Запис авторської інформації до вихідного файлу: true

10.1(2) - Результат підрахунку кількості абзаців та дозапис до вихідного файлу: true

10.1(3) - Результат пошуку слів та дозапис до вихідного файлу: true

10.2(1) - Результат підрахунку кількості пунктуаційних знаків та дозапис до вхідного файлу: true

10.2(2) - Дозапис часу редагування файлу до вхідного файлу: true

10.3(1) - Дозапис результату функції s\_calculation до вихідного файлу: true

10.3(2) - Результат конвертування десяткового числа в двійкове та дозапис до вихідного файлу: true

Запис тексту до вхідного файлу: true

10.1(1) - Запис авторської інформації до вихідного файлу: true

10.1(2) - Результат підрахунку кількості абзаців та дозапис до вихідного файлу: true

10.1(3) - Результат пошуку слів та дозапис до вихідного файлу: true

10.2(1) - Результат підрахунку кількості пунктуаційних знаків та дозапис до вхідного файлу: true

10.2(2) - Дозапис часу редагування файлу до вхідного файлу: true

10.3(1) - Дозапис результату функції s\_calculation до вихідного файлу: true

10.3(2) - Результат конвертування десяткового числа в двійкове та дозапис до вихідного файлу: true

Запис тексту до вхідного файлу: true

10.1(1) - Запис авторської інформації до вихідного файлу: true

10.1(2) - Результат підрахунку кількості абзаців та дозапис до вихідного файлу: true

10.1(3) - Результат пошуку слів та дозапис до вихідного файлу: true

10.2(1) - Результат підрахунку кількості пунктуаційних знаків та дозапис до вхідного файлу: true

10.2(2) - Дозапис часу редагування файлу до вхідного файлу: true

10.3(1) - Дозапис результату функції s\_calculation до вихідного файлу: true

10.3(2) - Результат конвертування десяткового числа в двійкове та дозапис до вихідного файлу: true

Запис тексту до вхідного файлу: true

10.1(1) - Запис авторської інформації до вихідного файлу: true

10.1(2) - Результат підрахунку кількості абзаців та дозапис до вихідного файлу: true

10.1(3) - Результат пошуку слів та дозапис до вихідного файлу: true

10.2(1) - Результат підрахунку кількості пунктуаційних знаків та дозапис до вхідного файлу: true

10.2(2) - Дозапис часу редагування файлу до вхідного файлу: true

10.3(1) - Дозапис результату функції s\_calculation до вихідного файлу: true

10.3(2) - Результат конвертування десяткового числа в двійкове та дозапис до вихідного файлу: true

Запис тексту до вхідного файлу: true

10.1(1) - Запис авторської інформації до вихідного файлу: true

10.1(2) - Результат підрахунку кількості абзаців та дозапис до вихідного файлу: true

10.1(3) - Результат пошуку слів та дозапис до вихідного файлу: true

10.2(1) - Результат підрахунку кількості пунктуаційних знаків та дозапис до вхідного файлу: true

10.2(2) - Дозапис часу редагування файлу до вхідного файлу: true

10.3(1) - Дозапис результату функції s\_calculation до вихідного файлу: true

10.3(2) - Результат конвертування десяткового числа в двійкове та дозапис до вихідного файлу: true

**Висновок:**

Проаналізувавши завдання лабораторної роботи та користуючись поняттям модульності програмного забезпечення було прийнято рішення про градуювання кожного завдання на підфункції, що сконцентровані на вирішенні конкретної, логічної частини завдання. Метою розбиття функцій є досягнення більш детального декларування результатів тестування окремих частин програми.

До вихідного коду статичної бібліотеки, створеної під час виконання лабораторної роботи №8 та №9, було додано реалізації функцій для розв’язування задач 10.1-10.3. Після компіляції проекту бібліотеки в теці \obj оновився файл з розширенням .а, що і є файлом статичної бібліотеки. Наступним кроком було додавання до заголовкового файлу бібліотеки прототипів функцій.

Для виведення результатів виконання функцій зі статичної бібліотеки labModulesOzhekhovskyi.a у консольне вікно застосунку TestDriver.exe було використано маніпулятор потокового виводу boolalpha із стандартної бібліотеки <iomanip>. Таким чином, підтверджується рівність очікуваних даних з отриманими у випадку, якщо функція повертає значення не типу bool. функції, з результатом повернення bool визначають, чи є, насамперед, істинним виконання даної функції, чи ні.

Враховано коректне відкриття та закриття файлових потоків для всіх функцій. Так, на початку функціонування кожної процедури оголошується змінна типу, відповідного до мети взамодії файлових потоків (ofstream - для запису / ifstream - для читання). Додатково, в аргументах даних змінних, окрім вказання назви файлу, вказано тип відкриття файлу (ios::out - для запису / ios::in - для зчитування / ios::app - для дозапису). В режимі ios::out працюють лише ті процедури, що мають виконати перезапис всього файлу (тобто функції, що вперше взаємодіють з файлом).

Аби запобігти повторенню однакової частини коду в кожній функції, на початку функціонування TestDriver.exe викликається abilityToEdit(), що перевіряє обидва файли на можливість до відкриття / редагування.

Також, для автоматизації процесу була опущена умова лабораторної роботи щодо необхідності запису будь-якого українського тексту до вхідного файлу власноруч. Натомість реалізовано додаткову процедуру, що за прийнятим аргументом в якості будь-якого українського тексту виконує його запис до вхідного файлу.

Пошук знаків пунктуації, символу переносу рядка у тексті здійснюється завдяки перевірці кожного елементу рядка чи дорівнює він одному зі знаків пунктуації та символу переносу рядка. Надалі виконується порівняння з константними значеннями результату. Якщо умова виконується - пошук вважається успішним.

Функція визначення двійкового представлення числа основана на стандартному (“ручному”) методі переведення числа з десяткової системи числення до двійкової. Врахувавши, що число типу int вміщує в собі 32 біти, оголошуємо масив з відповідним розміром для його подальшого заповнення двійковими 0 та 1. В першому циклі вхідне число поступово ділиться на 2 з отриманням остачі (використовується оператор %) та ділиться на 2 звичайним методом для подальшої ітерації. Цикл завершується, коли числа стає рівне 0. Так як біти в числі рахуються зправа-наліво, створено наступний цикл, в котрому відбувається поелементний запис до вихідного файлу елементів масиву, починаючи з останнього і закінчуючи початковим.

Результат unit test–у всіх тест кейсів є Passed. Автоматизоване тестування проведено успішно.

**ДОДАТОК А**

(TestSuite до завдання 10.1)

|  |  |  |
| --- | --- | --- |
|  | Назва тестового набору /  **Test Suite Description** | TS\_10\_1 |
|  | Назва проекта/ПЗ /  **Name of project** | TestDriver.ехе |
|  | Рівень тестування /  **Level of testing** | Модульний |
|  | Автор тест-сьюта /  **Test Suite Author** | Ожеховський Владислав |
|  | Виконавець /  **Implementer** | Ожеховський Владислав |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Actions** | **Expected Result** | **Test Result** |
| TS\_1 | 1. Запис  "Ще не вмерла України і слава, і воля.  Ще нам, браття молодії, усміхнеться доля.  Згинуть наші вороженьки, як роса на сонці,  Запануєм і ми, браття, у своїй сторонці." у prjInputFile.txt  2. Запис інформації у prjOutputFile.txt  3. Запис кількості абзаців у prjOutputFile.txt  4. Запис пошуку слів “Україна” “університет” “блокнот” у prjOutputFile.txt | **Вміст prjInputFile.txt:**  Ще не вмерла України і слава, і воля.  Ще нам, браття молодії, усміхнеться доля.  Згинуть наші вороженьки, як роса на сонці,  Запануєм і ми, браття, у своїй сторонці  **Вміст prjOutputFile.txt:**  ===========================================  Виконавець: Ожеховський Владислав  Рік розробки: 2022  Місто/Країна: Кропивницький/Україна  ВНЗ: Центральний Національний Технічний Університет  ===========================================  Кількість абзаців з вхідного файлу: 4  Знайдено слово 'Українa'  Cлово 'університет' - не знайдено  Cлово 'блокнот' - не знайдено | **Passed** |
| TS\_2 | 1. Запис  "Наш клас писав про навчальні заклади, я написав про університет." у prjInputFile.txt  2. Запис інформації у prjOutputFile.txt  3. Запис кількості абзаців у prjOutputFile.txt  4. Запис пошуку слів “Україна” “університет” “блокнот” у prjOutputFile.txt | **Вміст prjInputFile.txt:**  Наш клас писав про навчальні заклади, я написав про університет  **Вміст prjOutputFile.txt:**  ===========================================  Виконавець: Ожеховський Владислав  Рік розробки: 2022  Місто/Країна: Кропивницький/Україна  ВНЗ: Центральний Національний Технічний Університет  ===========================================  Кількість абзаців з вхідного файлу: 1  Cлово 'Україна' - не знайдено  Знайдено слово 'університет'  Cлово 'блокнот' - не знайдено | **Passed** |
| TS\_3 | 1. Запис  "В блокнот люди записують якусь інформацію." у prjInputFile.txt  2. Запис інформації у prjOutputFile.txt  3. Запис кількості абзаців у prjOutputFile.txt  4. Запис пошуку слів “Україна” “університет” “блокнот” у prjOutputFile.txt | **Вміст prjInputFile.txt:**  В блокнот люди записують якусь інформацію  **Вміст prjOutputFile.txt:**  ===========================================  Виконавець: Ожеховський Владислав  Рік розробки: 2022  Місто/Країна: Кропивницький/Україна  ВНЗ: Центральний Національний Технічний Університет  ===========================================  Кількість абзаців з вхідного файлу: 1  Cлово 'Україна' - не знайдено  Cлово 'університет' - не знайдено  Знайдено слово 'блокнот' | **Passed** |
| TS\_4 | 1. Запис  "Душу й тіло ми положим за нашу свободу,  І покажем, що ми, браття, козацького роду." у prjInputFile.txt  2. Запис інформації у prjOutputFile.txt  3. Запис кількості абзаців у prjOutputFile.txt  4. Запис пошуку слів “Україна” “університет” “блокнот” у prjOutputFile.txt | **Вміст prjInputFile.txt:**  Душу й тіло ми положим за нашу свободу,  І покажем, що ми, браття, козацького роду  **Вміст prjOutputFile.txt:**  ===========================================  Виконавець: Ожеховський Владислав  Рік розробки: 2022  Місто/Країна: Кропивницький/Україна  ВНЗ: Центральний Національний Технічний Університет  ===========================================  Кількість абзаців з вхідного файлу: 2  Cлово 'Україна' - не знайдено  Cлово 'університет' - не знайдено  Cлово 'блокнот' - не знайдено | **Passed** |
| TS\_5 | 1. Запис  "А завзяття, праця щира свого ще докаже,  Ще ся волі в Україні піснь гучна розляже,  За Карпати відоб'ється, згомонить степами,  України слава стане поміж ворогами." у prjInputFile.txt  2. Запис інформації у prjOutputFile.txt  3. Запис кількості абзаців у prjOutputFile.txt  4. Запис пошуку слів “Україна” “університет” “блокнот” у prjOutputFile.txt | **Вміст prjInputFile.txt:**  А завзяття, праця щира свого ще докаже,  Ще ся волі в Україні піснь гучна розляже,  За Карпати відоб'ється, згомонить степами,  України слава стане поміж ворогами  **Вміст prjOutputFile.txt:**  ==========================================  Виконавець: Ожеховський Владислав  Рік розробки: 2022  Місто/Країна: Кропивницький/Україна  ВНЗ: Центральний Національний Технічний Університет  ===========================================  Кількість абзаців з вхідного файлу: 4  Знайдено слово 'Українa'  Cлово 'університет' - не знайдено  Cлово 'блокнот' - не знайдено | **Passed** |

**ДОДАТОК Б**

(TestSuite до завдання 10.2)

|  |  |  |
| --- | --- | --- |
|  | Назва тестового набору /  **Test Suite Description** | TS\_10\_2 |
|  | Назва проекта/ПЗ /  **Name of project** | TestDriver.ехе |
|  | Рівень тестування /  **Level of testing** | Модульний |
|  | Автор тест-сьюта /  **Test Suite Author** | Ожеховський Владислав |
|  | Виконавець /  **Implementer** | Ожеховський Владислав |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Action** | **Expected Result** | **Test Result** |
| **TS\_01** | 1. Запис кількості знаків пунктуації у InputFile.txt  2. Запис часу дозапису у InputFile.txt | **Вміст prjInputFile.txt:**  Ще не вмерла України і слава, і воля.  Ще нам, браття молодії, усміхнеться доля.  Згинуть наші вороженьки, як роса на сонці,  Запануєм і ми, браття, у своїй сторонці.  Кількість знаків пунктуації вхідного файлу: 10  Дата та час дозапису інформаціїї: [date] | **Passed** |
| **TS\_02** | 1. Запис кількості знаків пунктуації у InputFile.txt  2. Запис часу дозапису у InputFile.txt | **Вміст prjInputFile.txt:**  Наш клас писав про навчальні заклади, я написав про університет.  Кількість знаків пунктуації вхідного файлу: 2  Дата та час дозапису інформаціїї: [date] | **Passed** |
| **TS\_03** | 1. Запис кількості знаків пунктуації у InputFile.txt  2. Запис часу дозапису у InputFile.txt | **Вміст prjInputFile.txt:**  В блокнот люди записують якусь інформацію.  Кількість знаків пунктуації вхідного файлу: 1  Дата та час дозапису інформаціїї: [date] | **Passed** |
| **TS\_04** | 1. Запис кількості знаків пунктуації у InputFile.txt  2. Запис часу дозапису у InputFile.txt | **Вміст prjInputFile.txt:**  Душу й тіло ми положим за нашу свободу,  І покажем, що ми, браття, козацького роду.  Кількість знаків пунктуації вхідного файлу: 5  Дата та час дозапису інформаціїї: [date] | **Passed** |
| **TS\_05** | 1. Запис кількості знаків пунктуації у InputFile.txt  2. Запис часу дозапису у InputFile.txt | **Вміст prjInputFile.txt:**  А завзяття, праця щира свого ще докаже,  Ще ся волі в Україні піснь гучна розляже,  За Карпати відоб'ється, згомонить степами,  України слава стане поміж ворогами.  Кількість знаків пунктуації вхідного файлу: 6  Дата та час дозапису інформаціїї: [date] | **Passed** |

**ДОДАТОК В**

(TestSuite до завдання 10.3)

|  |  |  |
| --- | --- | --- |
|  | Назва тестового набору /  **Test Suite Description** | TS\_10\_3 |
|  | Назва проекта/ПЗ /  **Name of project** | TestDriver.ехе |
|  | Рівень тестування /  **Level of testing** | Модульний |
|  | Автор тест-сьюта /  **Test Suite Author** | Ожеховський Владислав |
|  | Виконавець /  **Implementer** | Ожеховський Владислав |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Action** | **Expected Result** | **Test Result** |
| **TS\_01** | 1. Запис результату s\_calculation (6, 3, -1) у outputFile.txt  2. Запис двійкового числа 5 у outputFile.txt | **Вміст outputFile.txt:**  **[…]**  Результат виконання s\_calculation: 728.516  Двійкове представлення числа 5 : 101 | **Passed** |
| **TS\_02** | 1. Запис результату s\_calculation (3.3, -4, 0.5) у outputFile.txt  2. Запис двійкового числа 27 у outputFile.txt | **Вміст outputFile.txt:**  **[…]**  Результат виконання s\_calculation: -3072.09  Двійкове представлення числа 27 : 11011 | **Passed** |
| **TS\_03** | 1. Запис результату s\_calculation (7, 0.8, 2) у outputFile.txt  2. Запис двійкового числа 422 у outputFile.txt | **Вміст outputFile.txt:**  **[…]**  Результат виконання s\_calculation: 2.43363  Двійкове представлення числа 422 : 110100110 | **Passed** |
| **TS\_04** | 1. Запис результату s\_calculation (0.5, -1, 7) у outputFile.txt  2. Запис двійкового числа 156 у outputFile.txt | **Вміст outputFile.txt:**  **[…]**  Результат виконання s\_calculation: -2.87187  Двійкове представлення числа 156 : 10011100 | **Passed** |
| **TS\_05** | 1. Запис результату s\_calculation (14, 2, 3) у outputFile.txt  2. Запис двійкового числа 5238 у outputFile.txt | **В Вміст outputFile.txt:**  **[…]**  Результат виконання s\_calculation: 97.6177  Двійкове представлення числа 5238 : 1010001110110 | **Passed** |

**ДОДАТОК Г**

(Вихідний код завдання 10.1-2-3)

**… / ModulesOzhekhovskyi / main.cpp**

#include "ModulesOzhekhovskyi.h"

#include <bitset>

#include <iostream>

#include <sstream>

using namespace std;

double s\_calculation(double x, double y, double z){

return ( sin(x) / sqrt( fabs( (y \* z) / (x + y) ) ) ) + (3 \* pow(y, 5));

}

**. . .**

//Lab\_10

void filesToEdit(string outputFileName, string inputFileName){

ofstream inputFile(inputFileName);

ofstream outputFile(outputFileName);

if (!inputFile || !outputFile) {

cout << "Неможливо відкрити файл для редагування\a" << endl;

inputFile.close();

outputFile.close();

exit(1);

}

inputFile.close();

outputFile.close();

}

bool inputTxtFile(string inputFileName, string inputText){

ofstream inputFile(inputFileName, ios::out);

inputFile << inputText << endl << endl;

inputFile.close();

return true;

}

//Lab\_10.1

bool authorInfo(string outputFileName){

ofstream outputFile(outputFileName, ios::out);

outputFile << "====================================================================" << endl

<< " Виконавець: Ожеховський Владислав" << endl

<< " Рік розробки: 2022" << endl

<< " Місто/Країна: Кропивницький/Україна" << endl

<< " ВНЗ: Центральний Національний Технічний Університет" << endl

<< "====================================================================" << endl << endl;

outputFile.close();

return true;

}

int numParagraphsInFile(string outputFileName, string textInInputFile){

ofstream outputFile(outputFileName, ios::app);

int numParagraphs = 1;

for (int i = 0; i < textInInputFile.length(); i++) {

if (textInInputFile[i] == '\n')

numParagraphs++;

}

outputFile << "Кількість абзаців з вхідного файлу: " << numParagraphs << endl << endl;

outputFile.close();

return numParagraphs;

}

bool findWordInText(string outputFileName, string textInInputFile){

ofstream outputFile(outputFileName, ios::app);

size\_t foundWord1 = textInInputFile.find("Україн");

size\_t foundWord2 = textInInputFile.find("університет");

size\_t foundWord3 = textInInputFile.find("блокнот");

if (foundWord1 != string::npos)

outputFile << "Знайдено слово 'Українa'" << endl << endl;

else

outputFile << "Cлово 'Україна' - не знайдено" << endl << endl;

if (foundWord2 != string::npos)

outputFile << "Знайдено слово 'університет'" << endl << endl;

else

outputFile << "Cлово 'університет' - не знайдено" << endl << endl;

if (foundWord3 != string::npos)

outputFile << "Знайдено слово 'блокнот'" << endl << endl;

else

outputFile << "Cлово 'блокнот' - не знайдено" << endl << endl;

outputFile.close();

return true;

}

//Lab\_10.2

int numPunctuationMarksInFile(string inputFileName, string textInInputFile){

ofstream inputTxtFile(inputFileName, ios::app);

char punctuationMarks[10] = { ',', '.', ':', ';', '!', '?', '-', '(', ')' };

int numPunctuationMarks = 0;

for (int i = 0; i < textInInputFile.length(); i++) {

for(int j = 0; j < 10; j++)

if (punctuationMarks[j] == textInInputFile[i]) {

numPunctuationMarks++;

}

}

inputTxtFile << "Кількість знаків пунктуації вхідного файлу: "

<< numPunctuationMarks << endl << endl;

inputTxtFile.close();

return numPunctuationMarks;

}

bool timestampInFile(string inputFileName){

ofstream inputFile(inputFileName, ios::app);

time\_t rawtime;

time(&rawtime);

inputFile << "Дата та час дозапису інформаціїї: " << ctime(&rawtime) << endl << endl;

inputFile.close();

return true;

}

//Lab\_10.3

double sCalculationResInFile(string outputFileName, double x, double y, double z){

ofstream outputFile("outputFile.txt", ios::app);

outputFile << "Результат виконання s\_calculation: " << s\_calculation(x, y, z) << endl << endl;

outputFile.close();

return s\_calculation(x, y, z);

}

bool decimalToBinary(string outputFileName, unsigned int number){

if (number <= 0) {

return false;

}

ofstream outputFile(outputFileName, ios::app);

int binaryCels[32];

int i;

unsigned int convertedNum = number;

for(i = 0; convertedNum > 0; i++) {

binaryCels[i] = convertedNum % 2;

convertedNum = convertedNum / 2;

}

outputFile << "Двійкове представлення числа " << number << " : ";

for(i = i - 1; i >= 0; i--) {

outputFile << binaryCels[i];

}

outputFile << endl << endl;

outputFile.close();

return true;

}

**… / ModulesOzhekhovskyi / ModulesOzhekhovskyi.h**

#ifndef MODULESOZHEKHOVSKYI\_H\_INCLUDED

#define MODULESOZHEKHOVSKYI\_H\_INCLUDED

#include <iostream>

#include <cmath>

#include <cstring>

using namespace std;

struct tornado {

int category = -1;

string description = "";

};

struct averageTemp{

double averageTempC;

double averageTempF;

string description = "";

};

double s\_calculation(double x, double y, double z);

tornado tornadoCategory(double speedWind);

averageTemp averageDailyTemp(double sumTempC);

int bitsNum(unsigned short int num);

void filesToEdit(string outputFileName, string inputFileName);

bool inputTxtFile(string inputFileName, string inputText);

string inputText();

bool authorInfo(string outputFileName);

int numParagraphsInFile(string outputFileName, string textInInputFile);

bool findWordInText(string outputFileName, string textInInputFile);

int numPunctuationMarksInFile(string inputFileName, string textInInputFile);

bool timestampInFile(string inputFileName);

double sCalculationResInFile(string outputFileName, double x, double y, double z);

bool decimalToBinary(string outputFileName, unsigned int number);

#endif // MODULESOZHEKHOVSKYI\_H\_INCLUDED

**ДОДАТОК Д**

**… / TestDriver / main.cpp**

#include <iostream>

#include "ModulesOzhekhovskyi.h"

#include <conio.h>

#include <cstring>

using namespace std;

int main()

{

system("chcp 65001 & cls");

const string inputText[5] = { "Ще не вмерла України і слава, і воля.\n"

"Ще нам, браття молодії, усміхнеться доля.\n"

"Згинуть наші вороженьки, як роса на сонці,\n"

"Запануєм і ми, браття, у своїй сторонці.",

"Наш клас писав про навчальні заклади, я написав про університет.",

"В блокнот люди записують якусь інформацію.",

"Душу й тіло ми положим за нашу свободу,\n"

"І покажем, що ми, браття, козацького роду.",

"А завзяття, праця щира свого ще докаже,\n"

"Ще ся волі в Україні піснь гучна розляже,\n"

"За Карпати відоб'ється, згомонить степами,\n"

"України слава стане поміж ворогами." };

const int numPunctuationMarks[5] = {10, 2, 1, 5, 6};

const int numParagraphs[5] = {4, 1, 1, 2, 4};

const double sCalcX[5] = { 6, 3.3, 7, 0.5, 14 };

const double sCalcY[5] = { 3, -4, 0.8, -1, 2 };

const double sCalcZ[5] = { -1, 0.5, 2, 7, 3 };

const unsigned int naturalNum[5] = {5, 27, 422, 156, 5238 };

const double expectedResult[5] = { 728.516, -3072.094, 2.433, -2.872, 97.617 };

for(int i = 0; i < 5; i++){

filesToEdit("outputFile.txt", "inputFile.txt");

cout << boolalpha

<< "Запис тексту до вхідного файлу: \t\t\t\t\t\t\t\t"

<< inputTxtFile("inputFile.txt", inputText[i]) << endl;

cout << boolalpha

<< "10.1(1) - Запис авторської інформації до вихідного файлу: \t\t\t\t\t"

<< authorInfo("outputFile.txt") << endl;

cout << boolalpha

<< "10.1(2) - Результат підрахунку кількості абзаців та дозапис до вихідного файлу: \t\t"

<< (numParagraphsInFile("outputFile.txt", inputText[i]) == numParagraphs[i]) << endl;

cout << boolalpha

<< "10.1(3) - Результат пошуку слів та дозапис до вихідного файлу: \t\t\t\t\t"

<< findWordInText("outputFile.txt", inputText[i]) << endl;

cout << boolalpha

<< "10.2(1) - Результат підрахунку кількості пунктуаційних знаків та дозапис до вхідного файлу: \t"

<< (numPunctuationMarksInFile("inputFile.txt", inputText[i]) == numPunctuationMarks[i]) << endl;

cout << boolalpha

<< "10.2(2) - Дозапис часу редагування файлу до вхідного файлу: \t\t\t\t\t"

<< timestampInFile("inputFile.txt") << endl;

cout << boolalpha

<< "10.3(1) - Дозапис результату функції s\_calculation до вихідного файлу: \t\t\t\t"

<< (floor(sCalculationResInFile("inputFile.txt", sCalcX[i], sCalcY[i], sCalcZ[i]) \* 1000) / 1000 == expectedResult[i]) << endl;

cout << boolalpha

<< "10.3(2) - Результат конвертування десяткового числа в двійкове та дозапис до вихідного файлу: \t"

<< decimalToBinary("outputFile.txt", naturalNum[i]) << endl << endl << endl;

\_getch();

}

return 0;

}