Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

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

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

[https://github.com/odorenskyi/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи КН-24

Безкровна В. В.

ПЕРЕВІРИЛА

викладачка кафедри кібербезпеки   
та програмного забезпечення

Анастасія КОВАЛЕНКО

Кропивницький – 2025

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

**МЕТА: набуття 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 - 11**

**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**

Завантажили власний Git-репозиторій.

У \Lab10 заповнили файл README.md, створили теки prj, TestSuite, Software, Report; отриманий вміст теки \Lab10 завантажили до Git репозиторію.

Аналіз задачі 10.1:

Дано: слово або один знак пунктуації.

Вивести: інформація про автора; кількість літер у слові, якщо введено слово; якщо введено один знак пунктуації – повідомлення, що у файлі знак пунктуації, у вхідному файлі додати рядок «Хай щастить!»; якщо введено щось інакше – ім’я і прізвище першого космонавта України; повідомлення, чи є слово із вхідного файла у частині вірша Анатолія Тарана «Хай щастить у вашій хаті, мамо».

Постановка задачі 10.1:

Вхід: input.txt.

Вихід: output.txt, якщо введено один знак пунктуації – output.txt, input.txt.

Аналіз вимог задачі 10.1:

* Мова повідомлень – українська;
* Вхідний файл \*.txt створюється користувачем;
* Імена вхідного й вихідного файлів є параметрами модуля;
* Перед читанням та записом файлів відбувається перевірка їх відкриття;
* Після завершення роботи, відбувається закриття відкритих файлів.

Вміст текстових файлів задачі 10.1:

Файл input.txt: слово або один знак пунктуації, якщо введено один знак пунктуації – «Хай щастить!».

Файл output.txt: інформація про автора, або «У файлі знак пунктуації», або «Леонід Каденюк» та «Слово міститься у наступній частині вірша Анатолія Тарана «Хай щастить у Вашій хаті, мамо»: …» або «Слово НЕ міститься у наступній частині вірша Анатолія Тарана «Хай щастить у Вашій хаті, мамо»: …».

Проектування архітектури задачі 10.1:

1. Підключення необхідних бібліотек;
2. Ініціалізація функції, оголошення file\_input, file\_output;
3. Відкриття необхідних файлів;
4. Оголошення необхідних змінних, ініціалізація poem;
5. Якщо введено слово, виводиться кількість літер у слові, програма продовжується; якщо введено один знак пунктуації – виводиться повідомлення, що у файлі знак пунктуації, у вхідному файлі записується рядок «Хай щастить!», відкриті файли закриваються, відбувається вихід з модуля; інакше – виводиться ім’я і прізвище першого космонавта України, відкриті файли закриваються, відбувається вихід з модуля;
6. Якщо слово входить у частину вірша, виводиться повідомлення, що слово міститься у частині вірша; Якщо слово не входить у частину вірша, виводиться повідомлення, що слово не міститься у частині вірша;
7. Відкриті файли закриваються, відбувається вихід з модуля.

Детальне проектування програмного модуля задачі 10.1:

1. Підключення бібліотек iostream, fstream, vector і algorithm;
2. Підключення простору імен std;
3. Ініціалізація функції, оголошення file\_input, file\_output;
4. Відкриття файлу output для письма, та файлу input для читання;
5. Виведення ім’я й прізвища розробника модуля, установи, міста, країни, року розробки;
6. Ініціалізація poem, found; оголошення text, words, word;
7. Цикл розбиває вірш на окремі слова;
8. Вхідний текст зчитується з input;
9. Файл input закривається;
10. Якщо вхідний текст містить 1 літерал і є знаком пунктуації, в output виводиться повідомлення «У файлі знак пунктуації»; для письма відкривається файл input, в input виводиться повідомлення «Хай щастить!»; Файли input та output закриваються. Модуль завершує роботу;
11. Якщо вхідний текст є словом, то виводиться повідомлення "Кількість літер у слові: " та значення text.length(). Перевірка виконується за допомогою булевої функції is\_word(string text), яка проходить по вхідному тексту, й перевіряє чи є кожен літерал буквою. Робота програми продовжується;
12. У інших варіантах, в output виводиться повідомлення «Леонід Каденюк». Файл output закривається, модуль завершує роботу;
13. Відбувається переведення вхідного слова, та слів вірша у нижній реєстр для коректного порівняння. Відбувається перевірка на наявність вхідного слова у частині вірша;
14. Якщо слово є у частині вірша, виводиться повідомлення «Слово міститься у наступній частині вірша Анатолія Тарана «Хай щастить у Вашій хаті, мамо»: …» і текст частини вірша, якщо слова немає у частині вірша, виводиться повідомлення «Слово НЕ міститься у наступній частині вірша Анатолія Тарана «Хай щастить у Вашій хаті, мамо»: …» і текст частини вірша;
15. Файл input закривається;
16. Вихід з модуля.

Аналіз задачі 10.2:

Дано: слово або один знак пунктуації.

Вивести: вірш А. Тарана «Хай щастить у вашій хаті, мамо» та дата й час дозапису інформації.

Постановка задачі 10.2:

Вхід: input.txt.

Вихід: input.txt.

Аналіз вимог задачі 10.2:

* Мова повідомлень – українська;
* Вхідний файл \*.txt створюється користувачем;
* Імена вхідного й вихідного файлів є параметрами модуля;
* Перед читанням та записом файлів відбувається перевірка їх відкриття;
* Після завершення роботи, відбувається закриття відкритих файлів.

Вміст текстових файлів задачі 10.2:

Файл input.txt: слово або один знак пунктуації, додані вірш А. Тарана «Хай щастить у вашій хаті, мамо» та дата й час дозапису інформації

Файл output.txt не задіяний.

Проектування архітектури задачі 10.2:

1. Підключення необхідних бібліотек;
2. Ініціалізація функції, оголошення file\_input;
3. Відкриття необхідного файлу у режимі додавання;
4. Виведення тексту вірша;
5. Оголошується змінна now зберігає поточний час, рядок розміром 100 символів для зберігання відформатованої дати та часу. Функція strftime форматує дату і час у рядок date\_time у форматі %Y-%m-%d %H:%M:%S;
6. Рядок date\_time виводиться;
7. Файл закривається. Відбувається вихід з модуля.

Детальне проектування програмного модуля задачі 10.2:

1. Підключення бібліотек iostream, fstream, string і ctime;
2. Підключення простору імен std;
3. Ініціалізація функції, оголошення file\_input;
4. Відкриття необхідного файлу в режимі додавання;
5. Виведення повного тексту вірша;
6. Ініціалізація змінної now типу time\_t, яка приймає значення функції time(NULL);
7. Оголошення рядка date\_time розміром 100 символів для зберігання відформатованої дати та часу;
8. Функція strftime форматує дату і час у рядок, за форматом %Y-%m-%d %H:%M:%S, і записує її у date\_time;
9. Виведення значення date\_time;
10. Закриття файлу input. Закінчення роботи модуля.

Аналіз задачі 10.3:

Дано: слово або один знак пунктуації, значення x, y та b.

Вивести: результати виконання функції s\_calculation(x, y) та число b у двійковому коді.

Постановка задачі 10.3:

Вхід: input.txt.

Вихід: output.txt.

Аналіз вимог задачі 10.3:

* Мова повідомлень – українська;
* Вхідний файл \*.txt створюється користувачем;
* Імена вхідного й вихідного файлів є параметрами модуля;
* Перед читанням та записом файлів відбувається перевірка їх відкриття;
* Після завершення роботи, відбувається закриття відкритих файлів.

Вміст текстових файлів задачі 10.3:

Файл input.txt: слово або один знак пунктуації.

Файл output.txt: результати виконання функції s\_calculation(x, y) та число b у двійковому коді.

Проектування архітектури задачі 10.3:

1. Підключення необхідних бібліотек;
2. Ініціалізація функції, оголошення x, y, b, file\_output;
3. Відкриття необхідного файлу у режимі додавання;
4. Виведення функції s\_calculation(x, y) з заданими x та y;
5. Переведення числа b у бітове представлення bit\_b;
6. Виведення bit\_b;
7. Файл закривається. Відбувається вихід з модуля.

Детальне проектування програмного модуля задачі 10.3:

1. Підключення бібліотек iostream, fstream, bitset, string, cstdlib;
2. Підключення простору імен std;
3. Ініціалізація функції, оголошення x, y, b, file\_output;
4. Відкриття файлу output в режимі додавання;
5. Виведення функції s\_calculation(x, y) з заданими x та y;
6. Переведення числа b у бітове представлення bit\_b;
7. Виведення bit\_b;
8. Закриття файлу output. Закінчення роботи модуля.

Розробили три набори контрольних прикладів до задач 10.1–10.3 для виконання модульного тестування функцій (Додаток А, Додаток Б, Додаток В).

В Code::Blocks IDE відкрили проект статичної бібліотеки ModulesBezkrovna з \Lab8\prj, створений під час виконання лабораторної роботи № 8.

За отриманими під час проектування програмних модулів артефактами виконали конструювання функцій: мовою програмування С++ реалізували функції, які за наданим інтерфейсом реалізовують розв’язування задач 10.1, 10.2 та 10.3 відповідно.

Скомпілювали проект статичної бібліотеки ModulesBezkrovna.

Відкрили проект заголовкового файлу ModulesBezkrovna, створений під час виконання лабораторної роботи №8, та доповнили його прототипами реалізованих функцій 10.1–10.3.

У середовищі Code::Blocks створили в теці \Lab10\prj проект консольного додатка, іменували його TestDriver.

Мовою програмування С++ реалізували консольний застосунок – тестовий драйвер для модульного тестування функцій розв’язування задач 10.1–10.3 за допомогою розроблених тест-сьютів з \Lab10\TestSuite та вхідного і вихідного текстових файлів.

Створений застосунок TestDriver.ехе перемістили у \Lab10\Software.

За допомогою TestDriver.ехе виконали автоматизоване тестування розроблених функцій розв’язування задач 10.1–10.3.

Результати модульного тестування функцій 10.1-10.3:

For task 10\_1

Test case 1 PASSED

Test case 2 PASSED

Test case 3 PASSED

Test case 4 PASSED

For task 10\_2

Test case 1 PASSED

Test case 2 PASSED

Test case 3 PASSED

For task 10\_3

Test case 1 PASSED

Test case 2 PASSED

Test case 3 PASSED

Вихідний код проектів ModulesBezkrovna та TestDriver включили до звіту як додатки (див. Додаток Г, Додаток Д).

**ВИСНОВОК:** в цій лабораторній роботі ми набули 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.

Ми досягли мети роботи:

1. Виконавши самостійну підготовку до виконання лабораторної роботи №10 для успішного виконання завдань.
2. Охайно оформивши роботу.
3. Аналізувавши мету роботи для правильного виконання лабораторної роботи.
4. Завантаживши копію власного Git-репозиторію.
5. Повторивши застосування команд git fetch та git merge на практиці.
6. Виконавши аналіз умови задач 10.1, 10.2 та 10.3.
7. Виконавши постановку задач 10.1, 10.2 та 10.3.
8. Виконавши аналіз вимог для програмного модуля розв’язування задач 10.1, 10.2 та 10.3.
9. Виконавши аналіз вмісту текстових файлів задач 10.1, 10.2 та 10.3.
10. Виконавши проектування архітектури для програмного модуля розв’язування задач 10.1, 10.2 і 10.3.
11. Виконавши детальне проектування програмного модуля розв’язування задач 10.1, 10.2 і 10.3.
12. Розробивши набір контрольних прикладів для модульного тестування до задач 10.1, 10.2, 10.3.
13. Доповнивши проект статистичної бібліотеки новими функціями.
14. Доповнивши проект заголовкового файлу новими прототипами функцій.
15. Реалізувавши тестовий драйвер.
16. Пригадавши процес підключення до проекту власних бібліотек.
17. Пригадавши процес підключення простору імен std.
18. Пригадавши використання команд git add, git commit, git merge для Git.
19. Створювавши коміти на протязі всієї роботи.
20. Відправлявши коміти до репозиторію.
21. Не знайшовши в програмах помилок.
22. Задокументувавши вихідні коди отриманих програм.
23. Зберігши отримані програми в окремій папці.
24. Зберігши одержані результати виконання завдань на флеш-накопичувач з метою забезпечення можливості їх використання під час виконання наступних лабораторних робіт.
25. Ознайомившись з файловим вводом/виводом в мові C++.
26. Дізнавшись про призначення бібліотеки fstream.
27. Дізнавшись про використання класу ifstream для читання вмісту файлів.
28. Дізнавшись про використання класу ofstream для запису даних в файли.
29. Дізнавшись про режим відкриття файлу ios::app для додавання записів до файлу.
30. Дізнавшись про режим відкриття файлу ios::trunc для видалення даних з файлу.
31. Дізнавшись про метод close(), який явно закриває файл.
32. Дізнавшись про використання динамічних масивів vector.
33. Дізнавшись про використання метода push\_back() класу std::vector, ля додавання елементів у кінець вектора.
34. Дізнавшись про використання об'єкта стандартного потоку виведення cerr, для виведення повідомлень про помилки на консоль.
35. Пригадавши використання булевого типу даних.
36. Пригадавши використання маніпуляторів в C++.
37. Пригадавши використання контейнера bitset.
38. Пригадавши синтаксис рядків та роботу з рядками в C++.
39. Пригадавши використання оператора розгалуження if.
40. Пригадавши синтаксис масивів в C++.
41. Пригадавши як здійснюється доступ до елементів одно- та багатовимірних масивів.
42. Пригадавши виведення масивів.
43. Пригадавши використання циклу for.
44. Пригадавши використання символьного типу.
45. Пригадавши чутливість до реєстру мови програмування C++.
46. Пригадавши види функцій в мові програмування C++.
47. Пригадавши процес зміни кодування проекту.
48. Гарно організувавши свою роботу для ефективного засвоєння інформації.
49. Аналізувавши результати своєї роботи.
50. Підсумувавши виконану роботу.

Додаток А

Таблиця 1

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_10\_1 |
| Назва проекта / ПЗ  Name of Project / Software | libModulesBezkrovna.а |
| Рівень тестування  Level of Testing | Модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Вероніка Безкровна |
| Виконавець  Implementer | Вероніка Безкровна |

Таблиця 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Preliminary Steps | Action (test steps) | Expected Result | Test Result (passed/failed) |
| TC-1 | Вхідний файл: input.txt  Вміст input.txt:  Запис  Вихідний файл: output.txt | Виклик функції: input\_analysis(input, output);  input = "text\_files\\files\_for\_testing\\input.txt"  output = "text\_files\\files\_for\_testing\\output.txt" | Вміст output.txt:  Безкровна Вероніка  ЦНТУ  Кропивницький  Україна  2025 рік  Кількість літер у слові: 5  Слово НЕ міститься у наступній частині вірша Анатолія Тарана "Хай щастить у Вашій хаті, мамо":  Хай щастить у Вашій хаті, мамо,  Всім, хто переступить Ваш поріг:  Добрим людям, і птахам так само,  І котові, що в теплі приліг;  Хай щастить кожненькій деревині,  Що до хати віти притуля,  Хай щастить малесенькій травині  Й вітерцю, що приліта здаля. | passed |
| TC-2 | Вхідний файл: input.txt  Вміст input.txt:  переступить  Вихідний файл: output.txt | Виклик функції: input\_analysis(input, output);  input = "text\_files\\files\_for\_testing\\input.txt"  output = "text\_files\\files\_for\_testing\\output.txt" | Вміст output.txt:  Безкровна Вероніка  ЦНТУ  Кропивницький  Україна  2025 рік  Кількість літер у слові: 11  Слово міститься у наступній частині вірша Анатолія Тарана "Хай щастить у Вашій хаті, мамо":  Хай щастить у Вашій хаті, мамо,  Всім, хто переступить Ваш поріг:  Добрим людям, і птахам так само,  І котові, що в теплі приліг;  Хай щастить кожненькій деревині,  Що до хати віти притуля,  Хай щастить малесенькій травині  Й вітерцю, що приліта здаля. | passed |
| TC-3 | Вхідний файл: input.txt  Вміст input.txt:  ?  Вихідний файл: output.txt | Виклик функції: input\_analysis(input, output);  input = "text\_files\\files\_for\_testing\\input.txt"  output = "text\_files\\files\_for\_testing\\output.txt" | Вміст input.txt:  Хай щастить!  Вміст output.txt:  Безкровна Вероніка  ЦНТУ  Кропивницький  Україна  2025 рік  У файлі знак пунктуації | passed |
| TC-4 | Вхідний файл: input.txt  Вміст input.txt:  99  Вихідний файл: output.txt | Виклик функції: input\_analysis(input, output);  input = "text\_files\\files\_for\_testing\\input.txt"  output = "text\_files\\files\_for\_testing\\output.txt" | Вміст output.txt:  Безкровна Вероніка  ЦНТУ  Кропивницький  Україна  2025 рік  Леонід Каденюк | passed |

Додаток Б

Таблиця 1

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_10\_2 |
| Назва проекта / ПЗ  Name of Project / Software | libModulesBezkrovna.а |
| Рівень тестування  Level of Testing | Модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Вероніка Безкровна |
| Виконавець  Implementer | Вероніка Безкровна |

Таблиця 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Preliminary Steps | Action (test steps) | Expected Result | Test Result (passed/failed) |
| TC-1 | Вхідний файл: input.txt  Вміст input.txt:  Розетка  Вихідний файл: input.txt | Виклик функції: add\_poem\_with\_time(input)  input = "text\_files\\files\_for\_testing\\input.txt"  output = "text\_files\\files\_for\_testing\\output.txt" | Вміст input.txt:  Розетка  Хай щастить у Вашій хаті, мамо,  Всім, хто переступить Ваш поріг:  Добрим людям, і птахам так само,  І котові, що в теплі приліг;  Хай щастить кожненькій деревині,  Що до хати віти притуля,  Хай щастить малесенькій травині  Й вітерцю, що приліта здаля.  Ви завжди за всіх були дбайливі -  Бо така ж та доля у вдови.  Бо ж як будуть всі навкруг щасливі,  То щасливі будете і Ви.  2025-04-11 08:33:44 | passed |
| TC-2 | Вхідний файл: input.txt  Вміст input.txt:  ,  Вихідний файл: input.txt | Виклик функції: add\_poem\_with\_time(input)  input = "text\_files\\files\_for\_testing\\input.txt"  output = "text\_files\\files\_for\_testing\\output.txt" | Вміст input.txt:  ,  Хай щастить у Вашій хаті, мамо,  Всім, хто переступить Ваш поріг:  Добрим людям, і птахам так само,  І котові, що в теплі приліг;  Хай щастить кожненькій деревині,  Що до хати віти притуля,  Хай щастить малесенькій травині  Й вітерцю, що приліта здаля.  Ви завжди за всіх були дбайливі -  Бо така ж та доля у вдови.  Бо ж як будуть всі навкруг щасливі,  То щасливі будете і Ви.  2025-04-11 08:33:44 | passed |
| TC-3 | Вхідний файл: input.txt  Вміст input.txt:  777  Вихідний файл: input.txt | Виклик функції: add\_poem\_with\_time(input)  input = "text\_files\\files\_for\_testing\\input.txt" | Вміст input.txt:  777  Хай щастить у Вашій хаті, мамо,  Всім, хто переступить Ваш поріг:  Добрим людям, і птахам так само,  І котові, що в теплі приліг;  Хай щастить кожненькій деревині,  Що до хати віти притуля,  Хай щастить малесенькій травині  Й вітерцю, що приліта здаля.  Ви завжди за всіх були дбайливі -  Бо така ж та доля у вдови.  Бо ж як будуть всі навкруг щасливі,  То щасливі будете і Ви.  2025-04-11 08:33:44 | passed |

Додаток В

Таблиця 1

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_10\_3 |
| Назва проекта / ПЗ  Name of Project / Software | libModulesBezkrovna.а |
| Рівень тестування  Level of Testing | Модульний / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Вероніка Безкровна |
| Виконавець  Implementer | Вероніка Безкровна |

Таблиця 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Preliminary Steps | Action (test steps) | Expected Result | Test Result (passed/failed) |
| TC-1 | Вхідні файли: input.txt, output.txt  Вміст input.txt:  обчислення  Файл output.txt - пустий  Вихідний файл: output.txt | Виклик функції: add\_calculation\_and\_bits(x, y, b, output)  x = 3  y = 4  b = 3  output = "text\_files\\files\_for\_testing\\output.txt" | Вміст output.txt:  -47.1988  00000000000000000000000000000011 | passed |
| TC-2 | Вхідні файли: input.txt, output.txt  Вміст input.txt:  ;  Вміст output.txt:  -47.1988  00000000000000000000000000000011  Вихідний файл: output.txt | Виклик функції: add\_calculation\_and\_bits(x, y, b, output)  x = 4  y = 10  b = 11  output = "text\_files\\files\_for\_testing\\output.txt" | Вміст output.txt:  -47.1988  00000000000000000000000000000011  -884.016  00000000000000000000000000001011 | passed |
| TC-3 | Вхідні файли: input.txt, output.txt  Вміст input.txt:  82  Вміст output.txt:  -47.1988  00000000000000000000000000000011  -884.016  00000000000000000000000000001011  Вихідний файл: output.txt | Виклик функції: add\_calculation\_and\_bits(x, y, b, output)  x = 7  y = 3  b = 675  output = "text\_files\\files\_for\_testing\\output.txt" | Вміст output.txt:  -47.1988  00000000000000000000000000000011  -884.016  00000000000000000000000000001011  68.7318  00000000000000000000001010100011 | passed |

Додаток Г

Лістинг ModulesBezkrovna:

#include <cmath>

#include <iostream>

#include <iomanip>

#include <bitset>

#include <string>

#include <sstream>

#include <locale>

#include <windows.h>

#include <fstream>

#include <vector>

#include <algorithm>

#include <ctime>

#include <cstdlib>

#define \_USE\_MATH\_DEFINES

using namespace std;

float s\_calculation (float x, float y)

{

if ((pow(y,x) - M\_PI) < 0)

{

cout << "Помилка: вираз під квадратним коренем від'ємний" << endl;

}

float S = pow(x,2) - pow(y, 3) + sqrt(pow(y,x) - M\_PI);

return S;

}

string ECTS\_rating (int rating)

{

SetConsoleCP(CP\_UTF8);

SetConsoleOutputCP(CP\_UTF8);

if (rating<1 || rating>100)

{

return "Помилка...";

}

if (rating>=1 && rating<=34)

{

return "Оцінка: FX, Пояснення: з обов'язковим повторним курсом.";

}

if (rating>=35 && rating<=59)

{

return "Оцінка: F, Пояснення: з можливістю повторного складання.";

}

if (rating>=60 && rating<=66)

{

return "Оцінка: E, Пояснення: відповідає мінімальним критеріям.";

}

if (rating>=67 && rating<=74)

{

return "Оцінка: D, Пояснення: непогано, але зі значною кількістю недоліків.";

}

if (rating>=75 && rating<=81)

{

return "Оцінка: C, Пояснення: загалом правильне виконання із певною значною кількістю помилок.";

}

if (rating>=82 && rating<=89)

{

return "Оцінка: B, Пояснення: вище середнього, деякі помилки.";

}

if (rating>=90 && rating<=100)

{

return "Оцінка: A, Пояснення: відмінне виконання з незначною кількістю помилок.";

}

}

string format\_float(float value) {

ostringstream oss;

oss << fixed << setprecision(2) << value;

return oss.str();

}

string week\_temperature(float mon, float tue, float wed, float thu, float fri, float sat, float sun, float &week\_celsius, float &week\_fahrenheit)

{

week\_celsius = (mon+tue+wed+thu+fri+sat+sun)/7;

week\_fahrenheit = 32+((9.0/5)\*week\_celsius);

string result = "Середньотижнева температура за Цельсієм: " + format\_float(week\_celsius) + "\nСередньотижнева температура за Фаренгейтом: " + format\_float(week\_fahrenheit);

return result;

}

int bits\_analysis(int N)

{

if (N < 0 || N > 17948360)

{

return nan("");

}

bitset<32> bit\_N(N);

int check = bit\_N[12] == 0 ? 32 - bit\_N.count() : bit\_N.count();

return check;

}

bool is\_word(string text)

{

for (char c : text)

{

if (!isalpha(c))

{

return false;

}

}

return !text.empty();

}

int input\_analysis(string file\_input, string file\_output)

{

setlocale (LC\_CTYPE, "ukr");

ofstream output(file\_output);

if (!output)

{

cerr << "Помилка, файл не може бути відкритий для письма!" << endl;

exit(1);

}

ifstream input(file\_input);

if (!input)

{

cerr << "Помилка, файл не може бути відкритий для читання!" << endl;

exit(1);

}

output << "Безкровна Вероніка" << endl;

output << "ЦНТУ" << endl;

output << "Кропивницький" << endl;

output << "Україна" << endl;

output << "2025 рік" << endl;

string text;

string poem = "Хай щастить у Вашій хаті, мамо, "

"Всім, хто переступить Ваш поріг: "

"Добрим людям, і птахам так само, "

"І котові, що в теплі приліг; "

"Хай щастить кожненькій деревині, "

"Що до хати віти притуля, "

"Хай щастить малесенькій травині "

"Й вітерцю, що приліта здаля. ";

vector<string> words;

string word;

bool found = false;

for (char c : poem)

{

if (isalpha(c))

{

word += c;

}

else

{

if (!word.empty())

{

words.push\_back(word);

word.clear();

}

}

}

if (!word.empty())

{

words.push\_back(word);

}

input >> text;

input.close();

if (text.length() == 1 && ispunct(text[0]))

{

output << "\nУ файлі знак пунктуації" << endl;

output.close();

ofstream input(file\_input, ios::trunc);

if (!input)

{

cerr << "Помилка, файл не може бути відкритий для письма!" << endl;

exit(1);

}

input << "Хай щастить!" << endl;

input.close();

return 0;

}

else if (is\_word(text))

{

output << "\nКількість літер у слові: " << text.length()<< endl;

}

else

{

output << "\nЛеонід Каденюк" << endl;

output.close();

return 0;

}

transform(text.begin(), text.end(), text.begin(), ::tolower);

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

{

transform(words[i].begin(), words[i].end(), words[i].begin(), ::tolower);

if (words[i] == text)

{

found = true;

break;

}

}

if (found)

{

output << "\nСлово міститься у наступній частині вірша Анатолія Тарана \"Хай щастить у Вашій хаті, мамо\":" << endl;

output << "Хай щастить у Вашій хаті, мамо,\nВсім, хто переступить Ваш поріг:\nДобрим людям, і птахам так само,\nІ котові, що в теплі приліг;\nХай щастить кожненькій деревині,\nЩо до хати віти притуля,\nХай щастить малесенькій травині\nЙ вітерцю, що приліта здаля." << endl;

}

else

{

output << "\nСлово НЕ міститься у наступній частині вірша Анатолія Тарана \"Хай щастить у Вашій хаті, мамо\":" << endl;

output << "Хай щастить у Вашій хаті, мамо,\nВсім, хто переступить Ваш поріг:\nДобрим людям, і птахам так само,\nІ котові, що в теплі приліг;\nХай щастить кожненькій деревині,\nЩо до хати віти притуля,\nХай щастить малесенькій травині\nЙ вітерцю, що приліта здаля." << endl;

}

output.close();

return 0;

}

int add\_poem\_with\_time(string file\_input)

{

ofstream input(file\_input, ios::app);

if (!input)

{

cerr << "Помилка, файл не може бути відкритий для письма!" << endl;

exit(1);

}

input << "Хай щастить у Вашій хаті, мамо,\nВсім, хто переступить Ваш поріг:\nДобрим людям, і птахам так само,\nІ котові, що в теплі приліг;\nХай щастить кожненькій деревині,\nЩо до хати віти притуля,\nХай щастить малесенькій травині\nЙ вітерцю, що приліта здаля.\n Ви завжди за всіх були дбайливі -\nБо така ж та доля у вдови.\nБо ж як будуть всі навкруг щасливі,\nТо щасливі будете і Ви." << endl;

time\_t now = time(NULL);

char date\_time[100];

strftime(date\_time, 100, "%Y-%m-%d %H:%M:%S", localtime(&now));

input << date\_time << endl;

input.close();

return 0;

}

int add\_calculation\_and\_bits(float x, float y, int b, string file\_output)

{

ofstream output(file\_output, ios::app);

if (!output)

{

cerr << "Помилка, файл не може бути відкритий для письма!" << endl;

exit(1);

}

output << s\_calculation(x, y) << endl;

bitset<32> bit\_b(b);

output << bit\_b << endl;

output.close();

return 0;

}

string read(string path)

{

ifstream file(path);

if (!file)

{

return "Помилка, файл не може бути відкритий для читання!";

}

string text;

string line;

while (getline(file, line))

{

text += line + '\n';

}

file.close();

return text;

}

vector<string> test\_results(string exp\_result\_1, string exp\_result\_2, string exp\_result\_3)

{

vector<string> Test\_results;

string expected\_result\_1 = read(exp\_result\_1);

string expected\_result\_2 = read(exp\_result\_2);

string expected\_result\_3 = read(exp\_result\_3);

Test\_results.push\_back(expected\_result\_1);

Test\_results.push\_back(expected\_result\_2);

Test\_results.push\_back(expected\_result\_3);

return Test\_results;

}

vector<string> test\_results\_10\_1(string exp\_result\_1, string exp\_result\_2, string exp\_result\_3, string exp\_result\_4, string exp\_result\_5)

{

vector<string> Test\_results;

string expected\_result\_1 = read(exp\_result\_1);

string expected\_result\_2 = read(exp\_result\_2);

string expected\_result\_3 = read(exp\_result\_3);

string expected\_result\_4 = read(exp\_result\_4);

string expected\_result\_5 = read(exp\_result\_5);

Test\_results.push\_back(expected\_result\_1);

Test\_results.push\_back(expected\_result\_2);

Test\_results.push\_back(expected\_result\_3);

Test\_results.push\_back(expected\_result\_4);

Test\_results.push\_back(expected\_result\_5);

return Test\_results;

}

vector<string> input\_10\_1(string exp\_result\_1, string exp\_result\_2, string exp\_result\_3, string exp\_result\_4)

{

vector<string> Test\_results;

string expected\_result\_1 = read(exp\_result\_1);

string expected\_result\_2 = read(exp\_result\_2);

string expected\_result\_3 = read(exp\_result\_3);

string expected\_result\_4 = read(exp\_result\_4);

Test\_results.push\_back(expected\_result\_1);

Test\_results.push\_back(expected\_result\_2);

Test\_results.push\_back(expected\_result\_3);

Test\_results.push\_back(expected\_result\_4);

return Test\_results;

}

vector<string> test\_results\_10\_2(string exp\_result\_1, string exp\_result\_2, string exp\_result\_3)

{

time\_t now = time(NULL);

char date\_time[100];

strftime(date\_time, 100, "%Y-%m-%d %H:%M:%S", localtime(&now));

vector<string> Test\_results;

string expected\_result\_1 = read(exp\_result\_1);

expected\_result\_1 += date\_time;

expected\_result\_1 += '\n';

string expected\_result\_2 = read(exp\_result\_2);

expected\_result\_2 += date\_time;

expected\_result\_2 += '\n';

string expected\_result\_3 = read(exp\_result\_3);

expected\_result\_3 += date\_time;

expected\_result\_3 += '\n';

Test\_results.push\_back(expected\_result\_1);

Test\_results.push\_back(expected\_result\_2);

Test\_results.push\_back(expected\_result\_3);

return Test\_results;

}

Додаток Д

Лістинг TestDriver:

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <locale>

#include <windows.h>

#include "ModulesBezkrovna.h"

using namespace std;

int main()

{

SetConsoleCP(866);

SetConsoleOutputCP(866);

setlocale (LC\_CTYPE, "ukr");

string exp\_result\_1\_1 = "text\_files\\expected\_results\\for\_10\_1\\10\_1\_1.txt";

string exp\_result\_1\_2 = "text\_files\\expected\_results\\for\_10\_1\\10\_1\_2.txt";

string exp\_result\_1\_3 = "text\_files\\expected\_results\\for\_10\_1\\10\_1\_3\_output.txt";

string exp\_result\_1\_4 = "text\_files\\expected\_results\\for\_10\_1\\10\_1\_3\_input.txt";

string exp\_result\_1\_5 = "text\_files\\expected\_results\\for\_10\_1\\10\_1\_4.txt";

string exp\_result\_2\_1 = "text\_files\\expected\_results\\for\_10\_2\\10\_2\_1.txt";

string exp\_result\_2\_2 = "text\_files\\expected\_results\\for\_10\_2\\10\_2\_2.txt";

string exp\_result\_2\_3 = "text\_files\\expected\_results\\for\_10\_2\\10\_2\_3.txt";

string exp\_result\_3\_1 = "text\_files\\expected\_results\\for\_10\_3\\10\_3\_1.txt";

string exp\_result\_3\_2 = "text\_files\\expected\_results\\for\_10\_3\\10\_3\_2.txt";

string exp\_result\_3\_3 = "text\_files\\expected\_results\\for\_10\_3\\10\_3\_3.txt";

string input\_1\_1 = "text\_files\\input\\for\_10\_1\\input\_1.txt";

string input\_1\_2 = "text\_files\\input\\for\_10\_1\\input\_2.txt";

string input\_1\_3 = "text\_files\\input\\for\_10\_1\\input\_3.txt";

string input\_1\_4 = "text\_files\\input\\for\_10\_1\\input\_4.txt";

string input\_2\_1 = "text\_files\\input\\for\_10\_2\\input\_1.txt";

string input\_2\_2 = "text\_files\\input\\for\_10\_2\\input\_2.txt";

string input\_2\_3 = "text\_files\\input\\for\_10\_2\\input\_3.txt";

string input\_3\_1 = "text\_files\\input\\for\_10\_3\\input\_1.txt";

string input\_3\_2 = "text\_files\\input\\for\_10\_3\\input\_2.txt";

string input\_3\_3 = "text\_files\\input\\for\_10\_3\\input\_3.txt";

float x[] = {3, 4 ,7};

float y[] = {4, 10, 3};

int b[] = {3, 11, 675};

vector<string> test\_res\_1 = test\_results\_10\_1(exp\_result\_1\_1, exp\_result\_1\_2, exp\_result\_1\_3, exp\_result\_1\_4, exp\_result\_1\_5);

vector<string> test\_res\_2 = test\_results\_10\_2(exp\_result\_2\_1, exp\_result\_2\_2, exp\_result\_2\_3);

vector<string> test\_res\_3 = test\_results(exp\_result\_3\_1, exp\_result\_3\_2, exp\_result\_3\_3);

vector<string> input\_1 = input\_10\_1(input\_1\_1, input\_1\_2, input\_1\_3, input\_1\_4);

vector<string> input\_2 = test\_results(input\_2\_1, input\_2\_2, input\_2\_3);

vector<string> input\_3 = test\_results(input\_3\_1, input\_3\_2, input\_3\_3);

string input = "text\_files\\files\_for\_testing\\input.txt";

string output = "text\_files\\files\_for\_testing\\output.txt";

ofstream test\_results("text\_files\\test\_results\\test\_results.txt");

test\_results << "For task 10\_1" << endl;

ofstream input\_10\_1\_3\_result(input);

input\_10\_1\_3\_result << input\_1[2] << endl;

input\_10\_1\_3\_result.close();

ifstream input\_10\_1\_3\_input(input);

string result\_input\_10\_1\_3 = read(input);

input\_10\_1\_3\_input.close();

for(int i = 0; i < 2; i++)

{

ofstream input\_10\_1(input);

input\_10\_1 << input\_1[i] << endl;

input\_10\_1.close();

input\_analysis(input, output);

string result = read(output);

if(result == test\_res\_1[i])

{

test\_results << "Test case " << i + 1 << " PASSED" << endl;

}

else

{

test\_results << "Test case " << i + 1 << " FAILED" << endl;

}

}

ofstream input\_10\_1\_3(input);

input\_10\_1\_3 << input\_1[2] << endl;

input\_analysis(input, output);

string result\_1 = read(output);

string result\_2 = read(input);

if((result\_1 == test\_res\_1[2]) && (result\_2 == test\_res\_1[3]))

{

test\_results << "Test case 3 PASSED" << endl;

}

else

{

test\_results << "Test case 3 FAILED" << endl;

}

input\_10\_1\_3.close();

ofstream input\_10\_1\_4(input, ios::trunc);

input\_10\_1\_4 << input\_1[3] << endl;

input\_analysis(input, output);

string result = read(output);

if(result == test\_res\_1[4])

{

test\_results << "Test case 4 PASSED" << endl;

}

else

{

test\_results << "Test case 4 FAILED" << endl;

}

input\_10\_1\_4.close();

test\_results << "\nFor task 10\_2" << endl;

for(int i = 0; i < 3; i++)

{

ofstream input\_10\_2(input);

input\_10\_2 << input\_2[i] << endl;

input\_10\_2.close();

add\_poem\_with\_time(input);

string result = read(input);

if(result == test\_res\_2[i])

{

test\_results << "Test case " << i + 1 << " PASSED" << endl;

}

else

{

test\_results << "Test case " << i + 1 << " FAILED" << endl;

}

}

ofstream output\_cleaning(output);

output\_cleaning.close();

test\_results << "\nFor task 10\_3" << endl;

for(int i = 0; i < 3; i++)

{

ofstream input\_10\_3(input);

input\_10\_3 << input\_3[i] << endl;

input\_10\_3.close();

add\_calculation\_and\_bits(x[i], y[i], b[i], output);

string result = read(output);

if(result == test\_res\_3[i])

{

test\_results << "Test case " << i + 1 << " PASSED" << endl;

}

else

{

test\_results << "Test case " << i + 1 << " FAILED" << endl;

}

}

cout << "Тестування закінчено. Перегляньте результати в файлі \"test\_results.txt\" в директорії text\_files\test\_results" << endl;

return 0;

}