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 I/O

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 КІ - 24

Балан. М.С.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

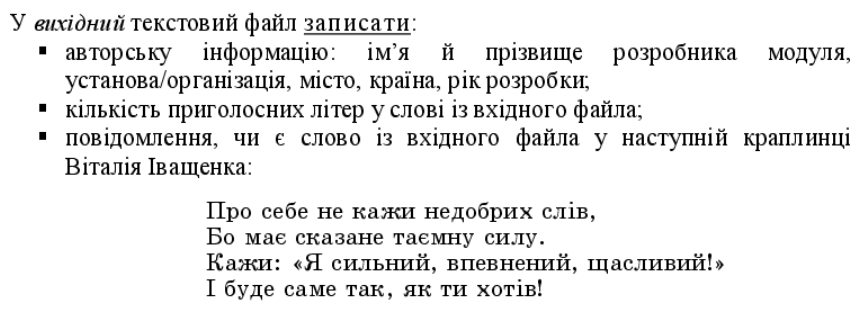
Коваленко А.С

Кропивницький – 2025

**ТЕМА:** Реалізація програмних модулів оброблення даних складових типів з файловим I/O

**МЕТА** Полягає у набутті ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою 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

**ЗАВДАННЯ №10.1**



**ПОСТАНОВА ЗАДАЧІ**  
Потрібно розробити програму, яка аналізує слово, зчитане з вхідного файлу. Програма має визначати кількість приголосних літер у слові та перевіряти, чи міститься це слово у заданому вірші Віталія Іващенка. Результати аналізу повинні виводитися у вихідний файл разом з інформацією про розробника, установу, місто, країну та рік створення програми.

**АНАЛІЗ ЗАДАЧІ**  
Задача передбачає роботу з файлами, рядками та символами. Потрібно зчитати слово з вхідного файлу, перевірити кожну його літеру на належність до списку приголосних, який включає як українські, так і англійські приголосні. Далі слід здійснити пошук цього слова у наперед заданому тексті вірша. Якщо слово знайдено у вірші, це фіксується. Всі результати обробки мають бути записані у вихідний файл у визначеному форматі.

**РЕАЛІЗАЦІЯ БІБЛІОТЕКИ**

Заголовковий файл

void analyzeWord(std::ifstream& inFile, std::ofstream& outFile)

Вихідний файл

void analyzeWord(std::ifstream& inFile, std::ofstream& outFile) {

std::string word;

inFile >> word;

// Підрахунок приголосних літер

std::string consonants = "bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZбвгґджзйклмнпрстфхцчшщБВГҐДЖЗЙКЛМНПРСТФХЦЧШЩ";

int consonantCount = 0;

for (char c : word) {

if (consonants.find(c) != std::string::npos) {

consonantCount++;

}

}

// Вірш Віталія Іващенка

std::string poem = "Про себе не кажи недобрих слів Бо має сказане таємну силу Кажи Я сильний впевнений щасливий І буде саме так як ти хотів";

std::stringstream ss(poem);

std::string poemWord;

bool found = false;

while (ss >> poemWord) {

if (poemWord == word) {

found = true;

break;

}

}

// Запис результату у вихідний файл

outFile << "Розробник: Балан Микола\n";

outFile << "Установа: ЦНТУ\n";

outFile << "Місто: Кропивницький\n";

outFile << "Країна: Україна\n";

outFile << "Рік розробки: 2025\n\n";

outFile << "Кількість приголосних літер у слові "" << word << "": " << consonantCount << "\n";

outFile << "Слово "" << word << "" " << (found ? "присутнє" : "відсутнє") << " у вірші Віталія Іващенка.\n";

}

**РЕЗУЛЬТАТ ВИКОНАННЯ ТЕСТОВОГО ДРАЙВЕРА:**

Тестування функції analyzeWord (Завдання 10.1):

Тест кейс 1 | Вхід: "Кажи"

Очікуваний результат:

Розробник: Балан Микола

Установа: ЦНТУ

Місто: Кропивницький

Країна: Україна

Рік розробки: 2025

Кількість приголосних літер у слові "Кажи": 2

Слово "Кажи" присутнє у вірші Віталія Іващенка.

Фактичний результат:

Розробник: Балан Микола

Установа: ЦНТУ

Місто: Кропивницький

Країна: Україна

Рік розробки: 2025

Кількість приголосних літер у слові "Кажи": 2

Слово "Кажи" присутнє у вірші Віталія Іващенка.

Результат: PASSED

--------------------------------------

Тест кейс 2 | Вхід: "сильний"

Очікуваний результат:

Розробник: Балан Микола

Установа: ЦНТУ

Місто: Кропивницький

Країна: Україна

Рік розробки: 2025

Кількість приголосних літер у слові "сильний": 4

Слово "сильний" присутнє у вірші Віталія Іващенка.

Фактичний результат:

Розробник: Балан Микола

Установа: ЦНТУ

Місто: Кропивницький

Країна: Україна

Рік розробки: 2025

Кількість приголосних літер у слові "сильний": 4

Слово "сильний" присутнє у вірші Віталія Іващенка.

Результат: PASSED

--------------------------------------

Тест кейс 3 | Вхід: "таємну"

Очікуваний результат:

Розробник: Балан Микола

Установа: ЦНТУ

Місто: Кропивницький

Країна: Україна

Рік розробки: 2025

Кількість приголосних літер у слові "таємну": 3

Слово "таємну" присутнє у вірші Віталія Іващенка.

Фактичний результат:

Розробник: Балан Микола

Установа: ЦНТУ

Місто: Кропивницький

Країна: Україна

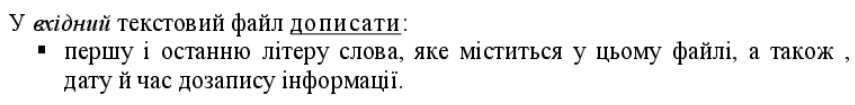
Рік розробки: 2025

Кількість приголосних літер у слові "таємну": 3

Слово "таємну" присутнє у вірші Віталія Іващенка.

Результат: PASSED

**ЗАВДАННЯ №10.2**



**ПОСТАНОВА ЗАДАЧІ**  
Необхідно створити програму, яка зчитує рядок тексту з вхідного файлу, виділяє з нього перше слово та аналізує його. Програма повинна визначити першу і останню літери цього слова та записати ці дані у вихідний файл разом із поточною датою й часом обробки.

**АНАЛІЗ ЗАДАЧІ**  
Задача зосереджена на обробці текстових файлів і роботі з рядками. Після зчитування тексту з вхідного файлу програма відокремлює перше слово. Якщо текст порожній, програма повідомляє про помилку. Далі вона визначає першу і останню літери цього слова, використовуючи доступ до першого й останнього символів. Крім цього, виводиться поточна дата та час за допомогою функцій роботи з часом. Усі отримані результати виводяться у вихідний файл у зручному для читання форматі.

**РЕАЛІЗАЦІЯ БІБЛІОТЕКИ**

Заголовковий файл

void appendWordInfo(std::ifstream& inFile, std::ofstream& outFile, const std::string& inputFileName)

Вихідний файл

void appendWordInfo(std::ifstream& inFile, std::ofstream& outFile, const std::string& inputFileName) {

std::string text;

std::getline(inFile, text);

std::stringstream ss(text);

std::string word;

ss >> word;

if (word.empty()) {

outFile << "Помилка: вхідний текст порожній.\n";

return;

}

char firstLetter = word.front();

char lastLetter = word.back();

// Безпечна версія отримання часу

time\_t now = time(nullptr);

char dt[100];

tm timeInfo;

localtime\_s(&timeInfo, &now);

std::strftime(dt, sizeof(dt), "%Y-%m-%d %H:%M:%S", &timeInfo);

// Записуємо результат у вихідний файл (outFile)

outFile << word << "\n";

outFile << "Перша літера слова: " << firstLetter << "\n";

outFile << "Остання літера слова: " << lastLetter << "\n";

outFile << "Дата і час допису: " << dt << "\n";

}

Тестування функції appendWordInfo (Завдання 10.2):

Тест кейс 1 | Вхід: "Привіт"

Очікуваний результат (вихідний файл):

Привіт

Перша літера слова: П

Остання літера слова: т

Дата і час допису: 2025-05-04 15:37:44

Фактичний результат (вихідний файл):

Привіт

Перша літера слова: П

Остання літера слова: т

Дата і час допису: 2025-05-04 16:15:38

Результат: PASSED

--------------------------------------

Тест кейс 2 | Вхід: "Слово"

Очікуваний результат (вихідний файл):

Слово

Перша літера слова: С

Остання літера слова: о

Дата і час допису: 2025-05-04 15:37:44

Фактичний результат (вихідний файл):

Слово

Перша літера слова: С

Остання літера слова: о

Дата і час допису: 2025-05-04 16:15:38

Результат: PASSED

--------------------------------------

Тест кейс 3 | Вхід: "Тест"

Очікуваний результат (вихідний файл):

Тест

Перша літера слова: Т

Остання літера слова: т

Дата і час допису: 2025-05-04 15:37:44

Фактичний результат (вихідний файл):

Тест

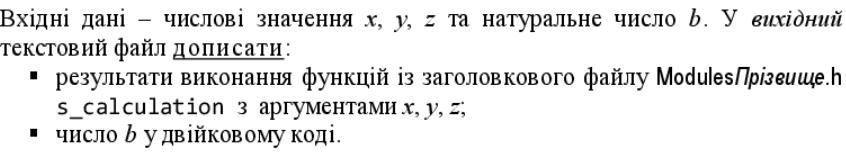
Перша літера слова: Т

Остання літера слова: т

Дата і час допису: 2025-05-04 16:15:38

Результат: PASSED

**ЗАВДАННЯ №10.3**



**ПОСТАНОВА ЗАДАЧІ**  
Потрібно розробити програму, яка зчитує з вхідного файлу три дійсні числа та одне ціле число, обчислює значення функції за заданою формулою, а також перетворює ціле число у двійкову систему числення. Результати мають бути виведені у вихідний файл з відповідним форматуванням.

**АНАЛІЗ ЗАДАЧІ**  
Завдання включає читання числових даних із файлу, виконання математичних обчислень через виклик функції s\_calculation, яка приймає три дійсні аргументи, та переведення цілого числа у двійковий код шляхом поділу на два з накопиченням залишків. При цьому реалізовано перевірку на нульове значення, що обробляється окремо. Результати обчислень форматуються з точністю до шести знаків після коми та виводяться у вихідний файл у зрозумілому для користувача вигляді.

**РЕАЛІЗАЦІЯ БІБЛІОТЕКИ**

Заголовковий файл

void computeAndConvert(std::ifstream& inFile, std::ofstream& outFile)

Вихідний файл

void computeAndConvert(std::ifstream& inFile, std::ofstream& outFile) {

double x, y, z;

int n;

inFile >> x >> y >> z >> n;

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

std::string binary = "";

int temp = n;

if (temp == 0) {

binary = "0";

}

else {

while (temp > 0) {

binary = (temp % 2 == 0 ? "0" : "1") + binary;

temp /= 2;

}

}

outFile << "Результат s\_calculation(" << x << ", " << y << ", " << z << "): "

<< std::fixed << std::setprecision(6) << s << "\n";

outFile << "Число " << n << " у двійковому коді: " << binary << "\n";

}

Тестування функції computeAndConvert (Завдання 10.3):

Тест кейс 1 | Вхід: "1 2 3 5"

Очікуваний результат:

Результат s\_calculation(1, 2, 3): -5.477823

Число 5 у двійковому коді: 101

Фактичний результат:

Результат s\_calculation(1, 2, 3): -5.477823

Число 5 у двійковому коді: 101

Результат: PASSED

--------------------------------------

Тест кейс 2 | Вхід: "5 1 2 13"

Очікуваний результат:

Результат s\_calculation(5, 1, 2): 27.239833

Число 13 у двійковому коді: 1101

Фактичний результат:

Результат s\_calculation(5, 1, 2): 27.239833

Число 13 у двійковому коді: 1101

Результат: PASSED

--------------------------------------

Тест кейс 3 | Вхід: "12 0 1 10"

Очікуваний результат:

Результат s\_calculation(12, 0, 1): 177.991905

Число 10 у двійковому коді: 1010

Фактичний результат:

Результат s\_calculation(12, 0, 1): 177.991905

Число 10 у двійковому коді: 1010

Результат: PASSED

**ВІДПОВІДЬ НА КОНТРОЛЬНІ ЗАПИТАННЯ**

1. **Яке призначення конструкції try-throw-catch у C++ і як вона працює?**  
   У C++ блок try використовується для розміщення коду, де можлива помилка. Якщо виникає виняток, він передається через throw, а catch ловить його і дозволяє обробити ситуацію без аварійного завершення програми.
2. **Як у C++ оголошується та використовується міжмодульна змінна?**  
   Міжмодульна змінна — це змінна, яка визначається в одному модулі, але використовується в іншому. Для цього її оголошують як extern у заголовковому файлі, а визначають в основному файлі.
3. **Яка область видимості змінних, оголошених у тілі функції main()?**  
   Змінні, оголошені в main, мають локальну область дії — вони доступні лише всередині цієї функції.
4. **У чому полягає різниця між типом enum і масивом у C++?**  
   Тип enum задає перелік сталих з іменами, тоді як масив дозволяє зберігати множину значень одного типу. enum менш гнучкий, але економить пам’ять, масив — більш універсальний для обробки даних.
5. **Що таке потік і чим файловий потік відрізняється від інших у C++?**  
   Потік — це засіб передавання інформації. Файловий потік призначений для взаємодії з файлами і відрізняється тим, що працює з постійним сховищем даних, а не з пристроями введення/виведення.
6. **Як виглядає загальний алгоритм роботи з файлами у C++?**  
   Потрібно відкрити файл у необхідному режимі, переконатися в успішному відкритті, виконати читання або запис і закрити файл по завершенні.
7. **Який буде розмір текстового файлу з числом π з 106 цифрами після крапки?**  
   Якщо записати число π з 106 знаками після крапки, файл матиме приблизно 107 байтів, оскільки кожен символ, включаючи крапку, займає 1 байт.
8. **Як перевірити успішність відкриття файлу у C++ і для чого це потрібно?**  
   Після відкриття файлу перевіряють стан потоку. Це дає змогу переконатися, що файл успішно відкрито і уникнути спроби роботи з недоступним файлом.
9. **У чому різниця між текстовим і двійковим потоками у C++?**  
   Текстові потоки працюють із символами і враховують формат, тоді як у двійкових передаються дані в такому вигляді, як вони представлені в пам’яті.
10. **Які режими відкриття файлів існують у C++?**  
    У C++ є режими відкриття для читання, запису, додавання, а також комбінації з текстовим або двійковим форматами.
11. **Які класи для роботи з файлами реалізовано у fstream та як їх використовувати?**  
    Бібліотека fstream включає класи ifstream, ofstream і fstream для відповідної роботи з файлами на читання, запис і комбіновано.
12. **Які існують константи режимів відкриття файлів у ios і яке їх призначення?**  
    Серед основних констант: in — читання, out — запис, app — додавання в кінець, trunc — очищення, binary — двійковий режим, ate — позиціонування в кінець при відкритті.
13. **Які функції використовуються для керування файлами у C++?**  
    Файли відкриваються через метод open(), закриваються — close(), а eof() дозволяє визначити, чи досягнуто кінця файлу.
14. **Як реалізувати видалення певного слова з текстового файлу?**  
    Спочатку вміст файлу зчитується в пам’ять, потім виконується пошук і видалення потрібного слова, після чого файл оновлюється.
15. **Які типи даних вважаються складовими в C++ і як із ними працювати?**  
    До складових типів належать масиви, структури, об’єднання та класи. Вони складніші за прості типи, вимагають більше уваги до ініціалізації й доступу до елементів.
16. **Як у C++ перевірити наявність елемента в множині та змінити її?**  
    Для цього використовують логічні або бітові масиви, а також STL-контейнери на кшталт set — елемент можна знайти, додати або видалити відповідною функцією.
17. **Як реалізуються явне і неявне перетворення типів у C++?**  
    Неявне перетворення типів виконується автоматично, коли типи сумісні. Явне вимагає використання оператора приведення типу програмістом.
18. **Коли доцільно використовувати тип union у C++ і як це виглядає на практиці?**  
    union використовують, коли потрібно економити пам’ять і працювати з різними типами даних, що не використовуються одночасно.
19. **Як знайти задане натуральне число у текстовому файлі засобами C++?**  
    Сканується вміст файлу поелементно (рядками чи словами), кожне значення порівнюється з заданим числом, при збігу пошук завершується.
20. **Що відбувається при використанні виразу sizeof short('R') у C++?**  
    Вираз sizeof short('R') є некоректним. 'R' інтерпретується як int, тому sizeof 'R' повертає розмір типу int. Щоб дізнатися розмір short, слід писати sizeof(short).

**ВИСНОВКИ**

1. Опрацьовано принципи роботи з масивами в C++.
2. Набуто навичок створення та ініціалізації структур.
3. Ознайомлено з особливостями використання об'єднань.
4. Досліджено можливості використання перерахувань (enum).
5. Розглянуто концепцію множин і способи її реалізації.
6. Вивчено способи приведення типів даних.
7. Оволодіто обробкою символьної інформації.
8. Опановано основи файлового введення та виведення.
9. Здобуто досвід відкриття й закриття файлів.
10. Навченося ефективно працювати з текстовими файлами.
11. Застосовано функції для збереження даних у файл.
12. Здійснено читання даних із файлів у різних форматах.
13. Опановано методи виявлення та обробки помилок при роботі з файлами.
14. Реалізовано використання динамічних масивів.
15. Вивчено обробку рядків за допомогою бібліотеки <string>.
16. Освоєно структурну організацію програмного коду.
17. Розвинено аналітичне мислення при вирішенні задач.
18. Створено та протестовано власні функції.
19. Закріплено знання щодо передачі параметрів у функції.
20. Відпрацьовано механізм повернення значень з функцій.
21. Застосовано заголовкові файли для оптимізації структури коду.
22. Опановано принципи інкапсуляції даних у структурах.
23. Вивчено методи форматування результатів виведення.
24. Поглиблено розуміння умовних операторів.
25. Застосовано цикли для реалізації повторюваних обчислень.
26. Опрацьовано вкладені цикли при роботі з багатовимірними масивами.
27. Реалізовано алгоритми сортування елементів масиву.
28. Здійснено пошук елементів у масивах.
29. Опановано копіювання та порівняння символьних рядків.
30. Отримано практику у виявленні та виправленні помилок.
31. Оволодіто середовищем розробки Code::Blocks.
32. Виконано компіляцію, запуск і налагодження програм.
33. Організовано програмний код у кількох файлах.
34. Створено власні заголовкові файли для повторного використання коду.
35. Проведено інтеграцію декількох модулів у спільному проєкті.
36. Вивчено застосування покажчиків у роботі з масивами.
37. Здійснено передачу структур як параметрів функцій.
38. Опановано використання символьних потоків.
39. Поглиблено знання про різновиди типів даних у C++.
40. Розглянуто особливості оголошення змінних різних типів.
41. Наголошено на важливості ініціалізації змінних.
42. Проаналізовано механізми ручного та автоматичного приведення типів.
43. Ознайомлено з використанням typedef та enum.
44. Використано об’єднання типів на практиці.
45. Закріплено знання про роботу з циклами for, while, do-while.
46. Опановано умовні конструкції if та switch.
47. Застосовано бібліотеки <fstream>, <iostream>, <cstring> у прикладних задачах.
48. Практично використано логічні оператори.
49. Отримано досвід обробки даних за допомогою програмних засобів.
50. Сформовано глибше розуміння концепцій модульного програмування.