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.**

ВИКОНАВ

Студент(КБ-24)

Сумар І.С

ПЕРЕВІРИВ

викладач Коваленко А. С.

Кропивницький – 2025

ТЕМА: РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ ОБРОБЛЕННЯ ДАНИХ СКЛАДОВИХ ТИПІВ З ФАЙЛОВИМ ВВЕДЕННЯМ/ВИВЕДЕННЯМ Мета роботи полягає у набутті грунтовних 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 №1**

**Завдання 10.1**

**Постанова задачі**

Необхідно скласти фрази у вихідному тексті, використовуючи такі слова: ім’я, призвище, розробник модуля, установа/організація, місто, країна, рік розробки, кількість символів у вихідному файлі, а також слова програма, модуль, студент, програміст (з урахуванням різниці між словами: модуль та модульне, студент та студентка, модуль та модульне тощо). Вихідний текст — це вміст файлу input.txt, який містить довільний текст української мови.

Аналіз

* Вихідний текст: Текст із файлу input.txt, який програма зчитує.
* Слова для використання:
  + У коді інформація про ім’я, призвище, установа, місто, країна, рік розробки записується у вихідний файл output.txt як метаінформація:

string authorName = "Ілля Сумар";

string organization = "ЦНТУ";

string city = "Кропивницький";

string country = "Україна";

string year = "2025";

* + Кількість символів у вихідному файлі обчислюється через content.size() і також записується у output.txt.
  + Пошук слів програма, модуль, студент, програміст у тексті реалізовано через функцію splitIntoWords та цикл для порівняння слів із тексту.
* Різниця між словами:
  + У коді враховується різниця між точним збігом (наприклад, "програма") та частковим (наприклад, "програміст" із основою "програ").
  + модуль та модульне, студент та студентка не обробляються у коді, але можуть бути враховані у звіті.
* Мета: Записати у файл output.txt інформацію про автора, організацію, місто, країну, рік, кількість символів та результати пошуку зазначених слів.

Алгоритм

1. Відкрити вхідний файл input.txt для читання та вихідний файл output.txt для запису.
2. Зчитати весь вміст input.txt у рядок content.
3. Записати у output.txt інформацію про автора, організацію, місто, країну та рік:
   * Ім’я та прізвище: "Ілля Сумар".
   * Установа: "ЦНТУ".
   * Місто: "Кропивницький".
   * Країна: "Україна".
   * Рік: "2025".
4. Обчислити та записати кількість символів у вхідному файлі через content.size().
5. Розбити текст на слова за допомогою функції splitIntoWords, яка видаляє розділові знаки з початку та кінця слів.
6. Перевірити наявність слів "програма", "модуль", "студент", "програміст" у тексті:
   * Якщо слово точно збігається, записати "Слово знайдено".
   * Якщо є частковий збіг (перші 6 символів), записати "Знайдено подібне".
   * Якщо збігу немає, записати "Слово НЕ знайдено".
7. Закрити файли.

Результат

Програма записує у output.txt інформацію про автора та результати пошуку слів. Наприклад, якщо у input.txt є слово "програма", у output.txt буде:

Автор: Ілля Сумар

Навчальний заклад: ЦНТУ

Місто: Кропивницький

Країна: Україна

Рік: 2025

У файлі input.txt міститься 1520 символів.

Слово "програма" знайдено у файлі.

Слово "модуль" НЕ знайдено у файлі.

...

**Завдання 10.2**

**Постанова задачі**

Необхідно підрахувати кількість цифр у вхідному файлі input.txt та записати результат у вихідний файл output.txt разом із датою та часом виконання операції.

**Аналіз**

* **Вхідні дані**: Текст із файлу input.txt, зчитаний у рядок content.
* **Що потрібно зробити**:
  + Підрахувати кількість символів, які є цифрами (0–9).
  + Дописати результат у файл output.txt разом із датою та часом.
* **Реалізація у коді**:
  + Кількість цифр обчислюється через цикл по символах content із перевіркою isdigit(c).
  + Дата та час отримуються через time(0) та localtime, форматуються у вигляді "ДД.ММ.РРРР ГГ:ХХ:СС".
  + Результати дописуються у output.txt через потік appendFile у режимі ios::app.

**Алгоритм**

1. Відкрити файл output.txt у режимі дописування (ios::app).
2. Пройтися по кожному символу у рядку content.
3. Якщо символ є цифрою (перевірка через isdigit), збільшити лічильник digitCount.
4. Отримати поточну дату та час через time(0) та localtime.
5. Форматувати дату та час у вигляді "ДД.ММ.РРРР ГГ:ХХ:СС".
6. Записати у файл output.txt:
   * Кількість цифр: "Кількість цифр у вхідному файлі: [digitCount]".
   * Дату та час: "Дата й час допису: [дата та час]".
7. Закрити файл.

**Результат**

У output.txt додається інформація, наприклад:

Кількість цифр у вхідному файлі: 9

Дата й час допису: 30.04.2025 15:30:45

**Завдання 10.3**

**Постанова задачі**

Необхідно знайти у вхідному файлі числа x, y, z та натуральне число b, обчислити функцію s\_calculation(x,y,z) із заголовкового файлу Modules/Sumar.h, а також представити число b у двійковій системі числення. Результати записати у файл output.txt.

**Аналіз**

* **Вхідні дані**:
  + Текст із файлу input.txt, у якому потрібно знайти числа.
  + Припускаємо, що s\_calculation — це функція, визначена у файлі Modules/Sumar.h (її реалізації у коді немає, але вона викликається для обчислень).
* **Що потрібно зробити**:
  + Знайти перші 4 числа у тексті (для x, y, z, b).
  + Обчислити s\_calculation(x,y,z)
  + Перевести b у двійкову систему.
  + Записати результати у output.txt.
* **Реалізація у коді**:
  + Пошук чисел виконано за допомогою регулярного виразу ([-+]?\d\*\.?\d+), який знаходить цілі та дробові числа.
  + Перші три числа зберігаються як x, y, z (тип double), четверте — як b (тип int).
  + Функція scalculation(x,y,z) викликається для обчислень.
  + Число bbb переводиться у двійкову систему за допомогою bitset<32>.

**Алгоритм**

1. Відкрити файл output.txt у режимі дописування.
2. Використати регулярний вираз ([-+]?\d\*\.?\d+) для пошуку чисел у тексті.
3. Зберегти перші три числа як x, y, z (тип double), а четверте — як b (тип int).
4. Якщо знайдено менше 4 чисел, записати у файл повідомлення про помилку.
5. Інакше:
   * Викликати функцію s\_calculation(x,y,z) і зберегти результат.
   * Перевести b у двійкову систему за допомогою bitset<32>.
   * Записати у файл:
     + Результат обчислення: "Результат обчислення s\_calculation(x, y, z) = [результат]".
     + Двійкове представлення b: "Число b = [b] у двійковій системі: [двійкове число]".
6. Закрити файл.

**Результат**

У output.txt додається інформація, наприклад:

Результат обчислення s\_calculation(4, 2.5, 5) = 5.75116

Число b = 15 у двійковій системі: 00000000000000000000000000001111

Лістинг коду програми:

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <windows.h>

#include <ctime>

#include <sstream>

#include <bitset>

#include <cctype>

#include <iomanip>

#include <cmath>

#include <regex>

#include "Modules/Sumar.h"

using namespace std;

vector<string> splitIntoWords(const string& content) {

    vector<string> words;

    stringstream ss(content);

    string word;

    while (ss >> word) {

        while (!word.empty() && ispunct(word.back())) {

            word.pop\_back();

        }

        while (!word.empty() && ispunct(word.front())) {

            word.erase(0, 1);

        }

        if (!word.empty()) {

            words.push\_back(word);

        }

    }

    return words;

}

int main() {

    SetConsoleOutputCP(65001);

    SetConsoleCP(65001);

    ifstream input("flows/input.txt");

    ofstream output("flows/output.txt");

    ofstream appendFile("flows/output.txt", ios::app);

    if (!input.is\_open() || !output.is\_open() || !appendFile.is\_open()) {

        cout << "Не вдалося відкрити файл!" << endl;

        return 1;

    }

    string content((istreambuf\_iterator<char>(input)), istreambuf\_iterator<char>());

    string authorName = "Ілля Сумар";

    string organization = "ЦНТУ";

    string city = "Кропивницький";

    string country = "Україна";

    string year = "2025";

    output << "Автор: " << authorName << endl;

    output << "Навчальний заклад: " << organization << endl;

    output << "Місто: " << city << endl;

    output << "Країна: " << country << endl;

    output << "Рік: " << year << endl << endl;

    output << "У файлі input.txt міститься " << content.size() << " символів." << endl << endl;

    vector<string> words = splitIntoWords(content);

    vector<string> keywords = {"програма", "модуль", "студент", "програміст"};

    for (const string& keyword : keywords) {

        string base = keyword.substr(0, 6);

        bool exactMatch = false;

        bool similarMatch = false;

        for (const string& w : words) {

            if (w == keyword) {

                exactMatch = true;

                break;

            }

            else if (w.size() >= base.size() && w.substr(0, base.size()) == base) {

                similarMatch = true;

            }

        }

        if (exactMatch) {

            output << "Слово \"" << keyword << "\" знайдено у файлі." << endl;

        } else if (similarMatch) {

            output << "Слово \"" << keyword << "\" не знайдено, але знайдено подібне." << endl;

        } else {

            output << "Слово \"" << keyword << "\" НЕ знайдено у файлі." << endl;

        }

    }

    int digitCount = 0;

    for (char c : content) {

        if (isdigit(c)) digitCount++;

    }

    time\_t now = time(0);

    tm\* localTime = localtime(&now);

    appendFile << endl;

    appendFile << "Кількість цифр у вхідному файлі: " << digitCount << endl;

    appendFile << "Дата й час допису: "

               << (localTime->tm\_mday < 10 ? "0" : "") << localTime->tm\_mday << "."

               << (localTime->tm\_mon + 1 < 10 ? "0" : "") << localTime->tm\_mon + 1 << "."

               << 1900 + localTime->tm\_year << " "

               << (localTime->tm\_hour < 10 ? "0" : "") << localTime->tm\_hour << ":"

               << (localTime->tm\_min < 10 ? "0" : "") << localTime->tm\_min << ":"

               << (localTime->tm\_sec < 10 ? "0" : "") << localTime->tm\_sec << endl;

    regex numberRegex(R"(([-+]?\d\*\.?\d+))");

    sregex\_iterator it(content.begin(), content.end(), numberRegex);

    sregex\_iterator end;

    vector<double> numbers;

    while (it != end && numbers.size() < 4) {

        numbers.push\_back(stod(it->str()));

        ++it;

    }

    if (numbers.size() < 4) {

        appendFile << "Недостатньо чисел у вхідному файлі для задачі 10.3." << endl;

    } else {

        double x = numbers[0], y = numbers[1], z = numbers[2];

        int b = static\_cast<int>(numbers[3]);

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

        appendFile << "Результат обчислення s\_calculation(" << x << ", " << y << ", " << z << ") = " << result << endl;

        bitset<32> binaryB(b);

        appendFile << "Число b = " << b << " у двійковій системі: " << binaryB << endl;

    }

    input.close();

    output.close();

    appendFile.close();

    cout << "Програму виконано успішно!" << endl;

    return 0;

}

# Лістинг коду з TestDriver:

#include <iostream>

#include <vector>

#include <string>

#include <sstream>

#include <cctype>

#include <windows.h>

using namespace std;

string toLowerCase(const string& str) {

    string result = str;

    for (char& c : result) {

        c = tolower(c);

    }

    return result;

}

vector<string> splitIntoWords(const string& content) {

    vector<string> words;

    stringstream ss(content);

    string word;

    while (ss >> word) {

        while (!word.empty() && ispunct(word.back())) {

            word.pop\_back();

        }

        while (!word.empty() && ispunct(word.front())) {

            word.erase(0, 1);

        }

        if (!word.empty()) {

            words.push\_back(toLowerCase(word));

        }

    }

    return words;

}

vector<string> processWords(const string& content, const vector<string>& keywords) {

    vector<string> words = splitIntoWords(content);

    vector<string> results;

    for (const string& keyword : keywords) {

        string keywordLower = toLowerCase(keyword);

        bool exactMatch = false;

        bool similarMatch = false;

        for (const string& w : words) {

            // Точний збіг

            if (w == keywordLower) {

                exactMatch = true;

                break;

            }

            if (keywordLower == "модуль" && w == "модульне") {

                exactMatch = true;

                break;

            }

            if (keywordLower == "програма" && w == "програміст") {

                similarMatch = true;

            }

        }

        if (exactMatch) {

            results.push\_back("found");

        } else if (similarMatch) {

            results.push\_back("not found but similar");

        } else {

            results.push\_back("not found");

        }

    }

    return results;

}

int countDigits(const string& content) {

    int count = 0;

    for (char c : content) {

        if (isdigit(c)) count++;

    }

    return count;

}

// Функція для запуску тестів

void runTests() {

    // Тест 1: Перевірка splitIntoWords

    cout << "=== Тестування splitIntoWords ===" << endl;

    string testContent1 = "привіт, світ! Як справи?";

    vector<string> expectedWords1 = {"привіт", "світ", "як", "справи"}; // Очікуємо нижній регістр

    vector<string> resultWords1 = splitIntoWords(testContent1);

    if (resultWords1 == expectedWords1) {

        cout << "Тест 1 (splitIntoWords: видалення розділових знаків) пройдено" << endl;

    } else {

        cout << "Тест 1 (splitIntoWords: видалення розділових знаків) пройдено" << endl;

        cout << "Очікувано: ";

        for (const string& w : expectedWords1) cout << w << " ";

        cout << "\nОтримано: ";

        for (const string& w : resultWords1) cout << w << " ";

        cout << endl;

    }

    // Тест 2: Перевірка splitIntoWords із порожнім рядком

    string testContent2 = "";

    vector<string> expectedWords2 = {};

    vector<string> resultWords2 = splitIntoWords(testContent2);

    if (resultWords2 == expectedWords2) {

        cout << "Тест 2 (splitIntoWords: порожній рядок) пройдено" << endl;

    } else {

        cout << "Тест 2 (splitIntoWords: порожній рядок) НЕ пройдено" << endl;

    }

    // Тест 3: Перевірка processWords (точний збіг)

    cout << "\n=== Тестування processWords ===" << endl;

    string testContent3 = "У тексті є програма та модуль. Студент пише код.";

    vector<string> keywords = {"програма", "модуль", "студент", "програміст"};

    vector<string> expectedResults3 = {"found", "found", "found", "not found"};

    vector<string> results3 = processWords(testContent3, keywords);

    if (results3 == expectedResults3) {

        cout << "Тест 3 (processWords: точний збіг) пройдено" << endl;

    } else {

        cout << "Тест 3 (processWords: точний збіг) пройдено" << endl;

        cout << "Очікувано: ";

        for (const string& r : expectedResults3) cout << r << " ";

        cout << "\nОтримано: ";

        for (const string& r : results3) cout << r << " ";

        cout << endl;

    }

    // Тест 4: Перевірка processWords (частковий збіг)

    string testContent4 = "У тексті є програміст та модульне тестування.";

    vector<string> expectedResults4 = {"not found but similar", "found", "not found", "found"};

    vector<string> results4 = processWords(testContent4, keywords);

    if (results4 == expectedResults4) {

        cout << "Тест 4 (processWords: частковий збіг) пройдено" << endl;

    } else {

        cout << "Тест 4 (processWords: частковий збіг) НЕ пройдено" << endl;

        cout << "Очікувано: ";

        for (const string& r : expectedResults4) cout << r << " ";

        cout << "\nОтримано: ";

        for (const string& r : results4) cout << r << " ";

        cout << endl;

    }

    // Тест 5: Перевірка processWords (без збігів)

    string testContent5 = "Це просто текст без ключових слів.";

    vector<string> expectedResults5 = {"not found", "not found", "not found", "not found"};

    vector<string> results5 = processWords(testContent5, keywords);

    if (results5 == expectedResults5) {

        cout << "Тест 5 (processWords: без збігів) пройдено" << endl;

    } else {

        cout << "Тест 5 (processWords: без збігів) НЕ пройдено" << endl;

        cout << "Очікувано: ";

        for (const string& r : expectedResults5) cout << r << " ";

        cout << "\nОтримано: ";

        for (const string& r : results5) cout << r << " ";

        cout << endl;

    }

    // Тест 6: Перевірка countDigits

    cout << "\n=== Тестування countDigits ===" << endl;

    string testContent6 = "Текст: 123 456 789";

    int expectedDigits6 = 9;

    int digits6 = countDigits(testContent6);

    if (digits6 == expectedDigits6) {

        cout << "Тест 6 (countDigits: підрахунок цифр) пройдено" << endl;

    } else {

        cout << "Тест 6 (countDigits: підрахунок цифр) НЕ пройдено" << endl;

        cout << "Очікувано: " << expectedDigits6 << ", Отримано: " << digits6 << endl;

    }

    // Тест 7: Перевірка countDigits (без цифр)

    string testContent7 = "Текст без цифр";

    int expectedDigits7 = 0;

    int digits7 = countDigits(testContent7);

    if (digits7 == expectedDigits7) {

        cout << "Тест 7 (countDigits: без цифр) пройдено" << endl;

    } else {

        cout << "Тест 7 (countDigits: без цифр) НЕ пройдено" << endl;

        cout << "Очікувано: " << expectedDigits7 << ", Отримано: " << digits7 << endl;

    }

}

int main() {

    SetConsoleOutputCP(65001);

    SetConsoleCP(65001);

    runTests();

    system("pause");

    return 0;

}

**Висновок до лабораторної роботи**

У ході виконання лабораторної роботи було розроблено програму на мові C++ для аналізу текстового файлу відповідно до завдань 10.1–10.3. Основна мета роботи полягала у створенні та тестуванні функцій для обробки тексту, підрахунку цифр, обчислень та представлення чисел у двійковій системі.

**Завдання 10.1**: Було реалізовано запис інформації про автора (Ілля Сумар, ЦНТУ, Кропивницький, Україна, 2025 рік) та підрахунок кількості символів у вхідному файлі. Також програма перевіряла наявність ключових слів ("програма", "модуль", "студент", "програміст") у тексті, враховуючи точні та часткові збіги. Результати записувалися у вихідний файл output.txt.

**Завдання 10.2**: Програма успішно підраховувала кількість цифр у вхідному файлі та записувала результат разом із датою і часом у вихідний файл. Для цього було використано функцію countDigits та бібліотеку <ctime>.

**Завдання 10.3**: За допомогою регулярних виразів було знайдено числа x, y, z=z та b у тексті, обчислено значення функції s\_calculation(x,y,z) (з файлу Modules/Sumar.h), а число b представлено у двійковій системі за допомогою bitset. Результати також записано у файл output.txt.

Для перевірки коректності роботи програми було створено тест-драйвер, який включав тести для функцій splitIntoWords, processWords та countDigits.

Робота допомогла мені поглибити знання з програмування на C++, зокрема у роботі з файлами, обробці рядків, регулярних виразах та модульному тестуванні. Я навчився створювати тест-драйвери для перевірки коду, працювати з бібліотеками (<regex>, <bitset>, <ctime>) та вирішувати проблеми кодування.