

Міністерство освіти і науки України  
Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення  
Дисципліна: Базові методології та технології програмування

**Лабораторна робота №10**

**Тема: «РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ ОБРОБЛЕННЯ  
ДАНИХ СКЛАДОВИХ ТИПІВ З ФАЙЛОВИМ  
ВВЕДЕННЯМ/ВИВЕДЕННЯМ»**

Виконав: ст. гр. КН-24

Куріщенко П. В.

Перевірив: викладач

Коваленко А.С.

Кропивницький 2025

## Варіант - 10

*Мета роботи* - полягає у набутті ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою програмування C++ 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 символічної інформації.

### ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

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

### ПОРЯДОК ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

#### Аналіз задач (10.1-10.3).

Користувач вводить довільне слово українською у вхідному текстовому файлі (input.txt). Далі програмний модуль:

- читає цей файл;
- виконує обробку даних (текст, числа);
- формує **вихідний текстовий файл** (output.txt) з результатами.

#### Завдання 1 (10.1).

Записати до вихідного файлу:

1. **Авторську інформацію:** Ім'я й прізвище розробника, установа/організація, місто, країна, рік розробки.

2. **Кількість символів нижнього регістру** у слові з вхідного файлу.

3. **Повідомлення, чи є це слово** у наведеному уривку з вірша Василя Симоненка «Матері»:

<i>В</i>	<i>хаті</i>	<i>сонячний</i>	<i>промінь</i>	<i>косо</i>
<i>На</i>	<i>долівку</i>	<i>ляга</i>	<i>з</i>	<i>вікна...</i>
<i>Твої</i>	<i>чорні</i>	<i>шовкові</i>	<i>коси</i>	
<i>Припорошила</i>		<i>вже</i>	<i>сивина.</i>	

#### **Аналіз вимог:**

Програма має: Зчитати слово з вхідного .txt файлу, порахувати кількість символів `islower(ch)` в цьому слові, перевірити наявність слова в уривку (можна привести до нижнього регістру).

#### **Завдання 2 (10.2).**

Дописати до вихідного файлу:

1. Слово, у якому **всі великі літери** замінені на малі, а **малі — на великі** (інверсія регістру).
2. **Поточну дату й час** додавання інформації.

#### **Аналіз вимог:**

1. Реалізувати інверсію символів.
2. Використати функції з `<ctime>`.

#### **Завдання 3 (10.3).**

**Вхідні дані:** Числові значення  $x$ ,  $y$ ,  $z$  і натуральне число  $b$ .

Дописати до вихідного файлу:

1. Результат функції `s_calculation(x, y, z)`.
2. Подати число  $b$  у двійковому коді.

### **Аналіз вимог:**

Виведення результату функції `s_calculation(x, y, z)`;

Подання `b` у двійковій формі: `std::bitset<8>(b)`.

### **Вміст вхідного текстового файлу:**

Повинен містити: 1 слово (українською мовою) — для 10.1 і 10.2;

### **Вихідний код ModulesКуріщенко:**

```
#include "sample_module.h"

float s_calculator(float x, float y, float z) {
    float S;
    if (x <= y){
        cout << "Error: log(x - y) is undefined (x must be greater
        than y).\n";
    }
    else if (x + (z / 2) * pow(y, 2) == 0){
        cout << "Error: Division by zero.\n";
    }
    else{
        S = log(x - y) + sqrt((M_PI * pow(x, 2))/(x + (z / 2 *
        pow(y, 2))));
        cout << "Your result is " << S;
    }
    return S;
}

void ffun_cal_discount(float purc_sum){...}

void sfun_male_sizes(int uw_size){...}

void tfun_bin_count(int number){...}

int low_letters_count(const string& word) { // 10.1(2)
```

```

int count = 0;

for (char ch : word) {

    string lowercase = "абвггдеежзийіклмнопрстуфхцчшщьюя";

    if (lowercase.find(ch) != string::npos) count++;

}

return (count / 2);

}

void ffun_word_searching(const string& inputFile, const string&
outputFile) { // 10.1
    ifstream fin(inputFile);
    ofstream fout(outputFile);
    if (!fin.is_open() || !fout.is_open()) {
        cerr << "Не вдалося відкрити файл." << endl;
        return;
    }

    string inputWord;
    fin >> inputWord;

    fout << "Автор: Павло Куріщенко\n";
    fout << "Організація: ЦНТУ\n";
    fout << "Місто/Країна: Кропивницький/Україна\n";
    fout << "Рік розробки: 2025\n\n";

    int fin_low_count = low_letters_count(inputWord);
    fout << "Кількість символів нижнього регістру у слові \"" <<
inputWord << "\": " << fin_low_count << "\n";

    string poem =
        "В хаті сонячний промінь косо\n"
        "На долівку ляга з вікна...\n"
        "Твої чорні шовкові коси\n"
        "Припорошила вже сивина.";

```

```

bool found = is_word_in_poem(inputWord, poem);
fout << "Слово \"" << inputWord << "\" " << (found ? "є" : "не
є") << " у вірші Василя Симоненка.\n";

fin.close();
fout.close();

}

void sfun_swap_case(const string& inputFile, const string&
outputFile) { // 10.2
    ifstream fin(inputFile);
    ofstream fout(outputFile, ios::app);

    string swappedWord;
    fin >> swappedWord;

    string lower = "абвгдеєжзиіїйклмнопрстуфхцчшщьюя";
    string upper = "АБВГДЕЄЖЗИІІЙКЛМНОПРСТУФХЦЧШЩЬЮЯ";

    for (char& ch : swappedWord) {
        size_t pos = lower.find(ch);
        if (pos != string::npos) {
            ch = upper[pos];
        }
        else {
            pos = upper.find(ch);
            if (pos != string::npos) {
                ch = lower[pos];
            }
        }
    }
}

time_t now = time(0);
char* dt = ctime(&now);

```

```

        fout << "\nІнверсія перістру: " << swappedWord << endl;
        fout << "Дата та час додавання: " << dt << endl;

        fout.close();

    }

    void tfun_write_calculations(const string& outputFile){
        ofstream fout(outputFile, ios::app);

        float x, y, z;
        cout << "Введи три дійсних числа(через пробіли): ";
        cin >> x >> y >> z;

        int b = 0;
        cout << "Введи натуральне число для подання у двійковому коді: ";
        cin >> b;

        fout << "\nРезультат обчислення s_calculator: " <<
        s_calculator(x, y, z) << endl;
        fout << "Число b у бінарному представленні: " << bitset<8>(b)
        << endl;

    }

    void Dev_info(){...}

```

### **Вихідний код TestDriver:**

```

#include "../.../lab08/prj/ModulesKurishchenko/sample_module.h"

int main() {

    SetConsoleOutputCP(65001);
    setlocale(LC_ALL, "uk_UA.UTF-8");

```

```

Dev_info();

ffun_word_searching("input.txt", "output.txt");
sfun_swap_case("input.txt", "output.txt");
tfun_write_calculations("output.txt");
return 0;

}

```

### Результати модульного тестування:

```

+-----+
| Pavlo Kurishchenko(c) |
| github.com/movavok    |
+-----+
Введи три дійсних числа(через пробіли): 34 8 10
Введи натуральне число для подання у двійковому коді: 2
Your result is 6.46106

```

### Висновок:

У процесі виконання лабораторної роботи №10 я закріпив теоретичні знання та практичні навички, пов'язані з обробкою текстових файлів, аналізом рядкових даних, модульним програмуванням та роботою з вхідними/вихідними потоками у мові програмування C++. Завдання вимагали уважності, логічного мислення та вміння працювати з текстом і числами одночасно. Під час реалізації кожного з етапів роботи я набув низку важливих умінь, які можуть стати у нагоді як у навчанні, так і в подальшій професійній діяльності. Зокрема, я навчився:

1. Закріпив навички роботи з вхідними та вихідними текстовими файлами.
2. Навчився створювати програму, яка аналізує текстовий файл.
3. Засвоїв принципи побудови логіки аналізу вхідного тексту.
4. Зрозумів, як використовувати символи нижнього та верхнього регістрів.
5. Освоїв перетворення регістрів символів у словах.



6. Навчився знаходити конкретні слова у заданому фрагменті тексту.
7. Закріпив знання роботи з функціями рядків (`string`) у мові програмування C++.
8. Ознайомився з основами обробки символів.
9. Дізнався, як перевіряти відкриття та закриття файлів.
10. Зрозумів важливість обробки помилок під час роботи з файлами.
11. Навчився використовувати `ofstream` та `ifstream` для запису та читання файлів.
12. Практикував навички форматowanego виведення інформації.
13. Навчився додавати інформацію до наявного файлу, не затираючи попередній вміст.
14. Засвоїв, як працювати з числовими типами даних при зчитуванні з файлу.
15. Навчився виконувати бітові перетворення (переведення числа у двійковий код).
16. Поглибив знання щодо передачі аргументів у функції.
17. Навчився створювати заголовкові файли.
18. Освоїв базові принципи модульного програмування.
19. Навчився організовувати структуру модуля з логічним поділом обов'язків.
20. Засвоїв принципи реєстрації та документування даних про автора модуля.
21. Відпрацював навички роботи з датою та часом у C++.
22. Ознайомився з функціями `time`, `ctime`, `localtime`.
23. Навчився реалізовувати контроль часу виконання програми.
24. Закріпив знання форматів представлення дати.
25. Дізнався, як реалізовувати українськомовне повідомлення у програмі.

26. Навчився автоматично формувати повноцінні повідомлення з аналізу.
27. Засвоїв навички зчитування символів у циклі до кінця файлу.
28. Навчився фільтрувати символи згідно з їхнім регістром.
29. Освоїв методи підрахунку символів у файлі.
30. Вивчив методи пошуку слова у рядку.
31. Навчився змінювати регістр усіх символів у слові.
32. Поглибив знання про структуру проєктів на C++.
33. Навчився створювати трасувальні таблиці.
34. Навчився створювати тестові кейси для перевірки правильності алгоритму.
35. Відпрацював уміння декомпозиції задачі на підзадачі.
36. Засвоїв послідовність обробки вхідних та вихідних даних.
37. Навчився використовувати коментарі у коді для кращого розуміння.
38. Ознайомився з принципами оформлення звітної документації.
39. Практикував написання структурованого та читабельного коду.
40. Навчився тестувати програму з різними вхідними даними.
41. Зрозумів необхідність валідації вхідних даних.
42. Навчився працювати з умовними операторами у задачах аналізу.
43. Освоїв базові алгоритми опрацювання рядкових даних.
44. Навчився створювати змінні для зберігання результатів аналізу.
45. Навчився уникати дублювання коду шляхом виділення окремих функцій.
46. Поглибив вміння логічного мислення при побудові алгоритмів.
47. Засвоїв принципи акуратної організації текстового файлу результатів.
48. Навчився враховувати локалізацію при роботі з датами та мовою.

49. Ознайомився з методами документування власного програмного коду.

50. Навчився генерувати вихідні повідомлення, зрозумілі користувачу.

51. Освоїв поняття “вхідний/вихідний” файл у контексті проєктування програми.

52. Навчився впроваджувати логіку контролю потоку виконання через файли.

53. Засвоїв правила створення назв файлів відповідно до завдання.

54. Отримав досвід виконання завдання, близького до реальних прикладних задач.