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-23

Гребенюк Д. О.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Дрєєва Г. М.

Кропивницький – 2024

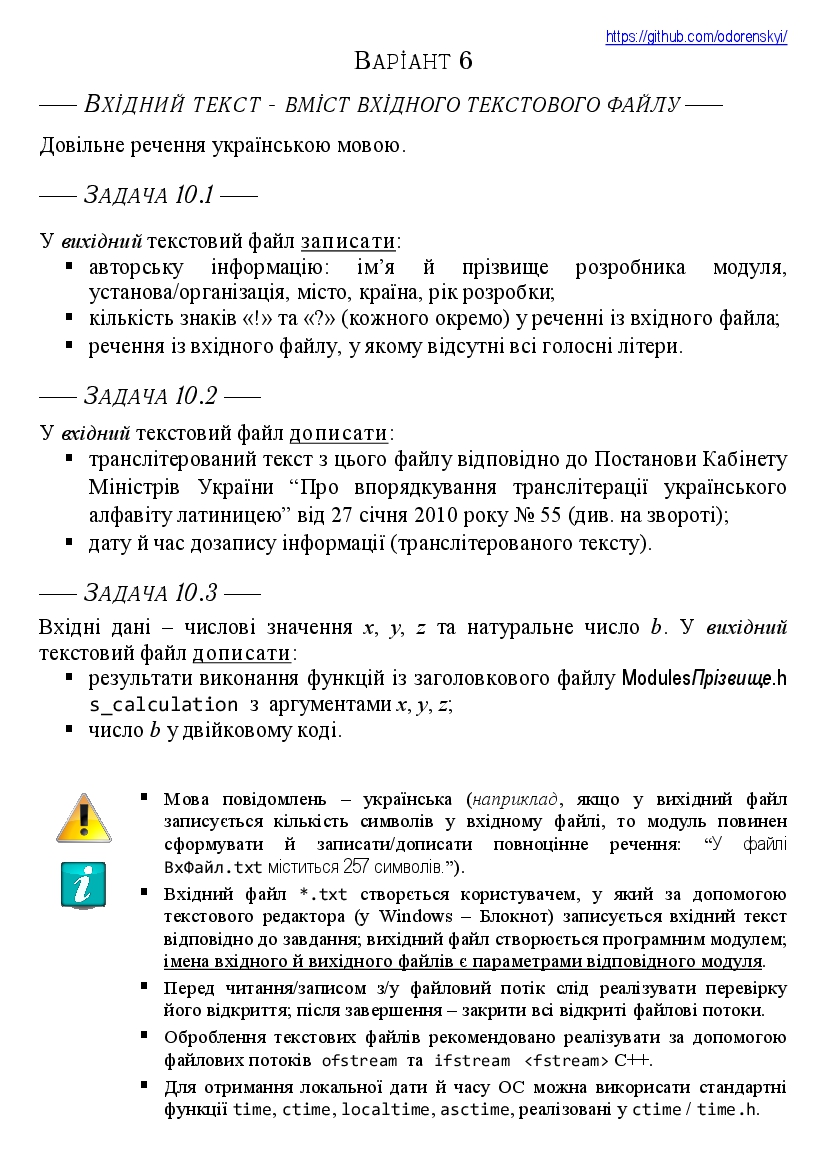
**Тема:** Реалізація програмних модулів оброблення даних складових типів з файловим введенням/виведенням

**Завдання до лабораторної роботи:**

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.

**Варіант 6**



**Задача 10.1**

Вхідні дані: inputFileName, outputFileName – назва вхідного та вихідного файлу

Вихідні дані: вихідний файл з вказаною назвою та його зміст: авторська інформація; кількість знаків «!» та «?»; речення, у якому відсутні всі голосні літери.

Для роботи з файлами потрібно використати файлові потоки ofstream та ifstream з <fstream>. Зміст вхідного файлу зчитується в змінну за допомогою функції getline, а потім обробляється відповідно до завдань. Спочатку обраховується кількість знаків за допомогою циклу, а потім в тому ж циклі перевіряється літера з переліку голосних літер. У випадку, якщо літера дійсно голосна – не додавати її в змінну з новим змістом файлу. В кінці подати інформацію українською мовою (за вимогою завдання), закрити файлові потоки.

**Задача 10.2**

Вхідні дані: inputFileName – назва вхідного файлу

Вихідні дані: транслітерований текст з файлу, дата й час дозапису тексту

В цій задачі потрібно дописати дані у вхідний файл, а не окремо у вихідний. Для реалізації транслітерації можна використати рядковий масив, індекси якого будуть відповідати коду символу в кодуванні Windows-1251. Відповідно, аналог на латині записуєтсья під індексом (кодом) кожної букви українського алфавіту. Для визначення дати можна застосувати <ctime>, а саме функції time та ctime.

**Задача 10.3**

Вхідні дані: x, y, z – числові значення; b – натуральне число; outputFileName – назва вихідного файлу

Вихідні дані: результат виконання функції s\_calculation з модуля ModulesHrebeniuk та число b у двійковому коді

Так само, як в попередніх задачах, використовується файловий потік ofstream для дозапису у файл. З модуля ModulesHrebeniuk використовується функція s\_calculation. Для переводу числа b у двійковий код використовується клас bitset, що надається заголовковим файлом <bitset>.

*Результат виконання тестового драйвера:*

=== TS\_10\_1 ===

Тест-кейс №1

Вхідні дані:

inputFileName: input\_1.txt

outputFileName: output\_1.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №2

Вхідні дані:

inputFileName: input\_2.txt

outputFileName: output\_2.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №3

Вхідні дані:

inputFileName: input\_3.txt

outputFileName: output\_3.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №4

Вхідні дані:

inputFileName: input\_1.txt

outputFileName: output\_4.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №5

Вхідні дані:

inputFileName: input\_4.txt

outputFileName: output\_1.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

=== TS\_10\_2 ===

Тест-кейс №1

Вхідні дані:

inputFileName: input\_1.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №2

Вхідні дані:

inputFileName: input\_2.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №3

Вхідні дані:

inputFileName: input\_3.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №4

Вхідні дані:

inputFileName: input\_1.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №5

Вхідні дані:

inputFileName: input\_4.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

=== TS\_10\_3 ===

Тест-кейс №1

Вхідні дані:

x: 5

y: -27

z: -21

b: 90

outputFileName: output\_1.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №2

Вхідні дані:

x: -20

y: 28

z: -24

b: 2024

outputFileName: output\_2.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №3

Вхідні дані:

x: 27

y: 21

z: 15

b: 912012

outputFileName: output\_3.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №4

Вхідні дані:

x: 15

y: 4

z: 14

b: 1007

outputFileName: output\_1.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed

Тест-кейс №5

Вхідні дані:

x: 27

y: -7

z: -25

b: 128

outputFileName: output\_4.txt

Відкрийте модифікований файл та порівняйте з текстом із поля Expected Result. Співпадає? (y/n):

Статус тест-кейса: passed