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

Федотов А. О.

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**

У вихідний текстовий файл записати:

* авторську інформацію: ім'я й прізвище розробника модуля,

установа/організація, місто, країна, рік розробки;

* кількість знаків «!» та «?» (кожного окремо) у реченні із вхідного файла;
* речення із вхідного файлу, у якому відсутні всі голосні літери.

**ПОСТАНОВА ЗАДАЧІ**  
Необхідно створити програму, яка читає вхідний текстовий файл, підраховує кількість знаків оклику (!) та питання (?) у кожному реченні, а також виявляє та записує речення, у яких немає жодної голосної літери. Отриману інформацію слід записати у вихідний текстовий файл разом з авторськими даними.

**АНАЛІЗ ЗАДАЧІ**  
Програма повинна працювати з текстовими файлами: зчитувати вхідний файл, обробляти його вміст та зберігати результати в окремий вихідний файл. Для виконання завдання потрібно розділити текст на речення, перевірити кожне з них на наявність голосних букв, а також підрахувати кількість символів «!» та «?». У вихідному файлі також мають бути вказані ім’я та прізвище розробника, навчальний заклад, місто, країна та рік розробки.

**РЕАЛІЗАЦІЯ БІБЛІОТЕКИ**

Вихідний файл ModulesFedotov.cpp

void processSentence(ifstream& input\_file, ofstream& output\_file) {

output\_file << "Розробник: Андрій Фкдотов\n"

<< "Установа: ЦНТУ\n"

<< "Місто: Кропивницький\n"

<< "Країна: Україна\n"

<< "Рік розробки: 2025\n\n";

string sentence;

if (getline(input\_file, sentence)) {

int exclamation = 0, question = 0;

for (char c : sentence) {

if (c == '!') exclamation++;

else if (c == '?') question++;

}

output\_file << "Кількість знаків '!': " << exclamation << endl;

output\_file << "Кількість знаків '?': " << question << endl << endl;

string no\_vowels;

for (char c : sentence) {

char lower = tolower(c);

if (!(lower == 'a' || lower == 'e' || lower == 'i' || lower == 'o' || lower == 'u' ||

lower == 'а' || lower == 'е' || lower == 'и' || lower == 'і' || lower == 'о' || lower == 'у')) {

no\_vowels += c;

}

}

output\_file << "Речення без голосних літер:\n" << no\_vowels << endl;

}

else {

output\_file << "Помилка: вхідний файл порожній або не містить речення." << endl;

}

}

Заголовковий файл ModulesFedotov.h

void processSentence(ifstream& input\_file, ofstream& output\_file);

**РЕЗУЛЬТАТ ВИКОНАННЯ ТЕСТОВОГО ДРАЙВЕРА:**

Тест кейс 1 | Вхід: "Hello! How are you? Great!!"

Очікуваний результат:

Розробник: Андрій Фкдотов

Установа: ЦНТУ

Місто: Кропивницький

Країна: Україна

Рік розробки: 2025

Кількість знаків '!': 3

Кількість знаків '?': 1

Речення без голосних літер:

Hll! Hw r y? Grt!!

Фактичний результат:

Розробник: Андрій Фкдотов

Установа: ЦНТУ

Місто: Кропивницький

Країна: Україна

Рік розробки: 2025

Кількість знаків '!': 3

Кількість знаків '?': 1

Речення без голосних літер:

Hll! Hw r y? Grt!!

Результат: PASSED

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

Тест кейс 2 | Вхід: "Це тестове речення."

Очікуваний результат:

Розробник: Андрій Фкдотов

Установа: ЦНТУ

Місто: Кропивницький

Країна: Україна

Рік розробки: 2025

Кількість знаків '!': 0

Кількість знаків '?': 0

Речення без голосних літер:

Ц тств рчння.

Фактичний результат:

Розробник: Андрій Фкдотов

Установа: ЦНТУ

Місто: Кропивницький

Країна: Україна

Рік розробки: 2025

Кількість знаків '!': 0

Кількість знаків '?': 0

Речення без голосних літер:

Ц тств рчння.

Результат: PASSED

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

Тест кейс 3 | Вхід: "aeiouAEIOUаеиіоу"

Очікуваний результат:

Розробник: Андрій Фкдотов

Установа: ЦНТУ

Місто: Кропивницький

Країна: Україна

Рік розробки: 2025

Кількість знаків '!': 0

Кількість знаків '?': 0

Речення без голосних літер:

Фактичний результат:

Розробник: Андрій Фкдотов

Установа: ЦНТУ

Місто: Кропивницький

Країна: Україна

Рік розробки: 2025

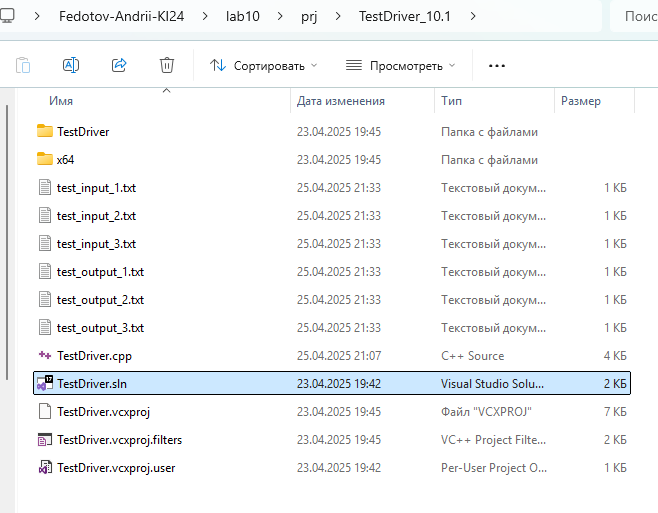
Кількість знаків '!': 0

Кількість знаків '?': 0

Речення без голосних літер:

Результат: PASSED

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



Малюнок 1 - Файли програми після запуску

**ЗАВДАННЯ №10.2**

У вхідний текстовий файл дописати:

* транслітерований текст з цього файлу відповідно до Постанови Кабінету

Міністрів України “Про впорядкування транслітерації українського

алфавіту латиницею” від 27 січня 2010 року № 55 (див. на звороті);

* дату й час дозапису інформації (транслітерованого тексту).

**ПОСТАНОВА ЗАДАЧІ**  
Потрібно дописати до вхідного текстового файлу транслітерований текст згідно з Постановою Кабінету Міністрів України “Про впорядкування транслітерації українського алфавіту латиницею” від 27 січня 2010 року № 55. Після транслітерованого тексту потрібно дописати поточну дату та час дозапису.

**АНАЛІЗ ЗАДАЧІ**  
Задача передбачає роботу з текстовим файлом, зчитування українського тексту, транслітерацію згідно з офіційними правилами, а також роботу з системним часом. Для реалізації необхідно мати транслітераційний словник або відповідну функцію, яка перетворює українські символи на латинські. Також потрібно отримати поточну дату і час у форматі, зручному для читання, і дописати ці дані разом із транслітерованим текстом у кінець файлу без втрати існуючої інформації.

**РЕАЛІЗАЦІЯ БІБЛІОТЕКИ**

Вихідний файл ModulesFedotov.cpp

void transliterateAndAppend(ifstream& input\_file, ofstream& output\_file) {

string content;

string line;

while (getline(input\_file, line)) {

if (!line.empty() && line.back() == '\r') {

line.pop\_back();

}

content += line + "\n";

}

string transliterated;

bool isWordStart = true;

for (size\_t i = 0; i < content.length(); ++i) {

char c = content[i];

string lowerC(1, tolower(c));

string upperC(1, toupper(c));

if (c == 'А' || c == 'а') transliterated += (c == 'А' ? "A" : "a");

else if (c == 'Б' || c == 'б') transliterated += (c == 'Б' ? "B" : "b");

else if (c == 'В' || c == 'в') transliterated += (c == 'В' ? "V" : "v");

else if (c == 'Г' || c == 'г') transliterated += (c == 'Г' ? "H" : "h");

else if (c == 'Ґ' || c == 'ґ') transliterated += (c == 'Ґ' ? "G" : "g");

else if (c == 'Д' || c == 'д') transliterated += (c == 'Д' ? "D" : "d");

else if (c == 'Е' || c == 'е') transliterated += (c == 'Е' ? "E" : "e");

else if (c == 'Є' || c == 'є') transliterated += (isWordStart ? (c == 'Є' ? "Ye" : "ye") : (c == 'Є' ? "Ie" : "ie"));

else if (c == 'Ж' || c == 'ж') transliterated += (c == 'Ж' ? "Zh" : "zh");

else if (c == 'З' || c == 'з') transliterated += (c == 'З' ? "Z" : "z");

else if (c == 'И' || c == 'и') transliterated += (c == 'И' ? "Y" : "y");

else if (c == 'І' || c == 'і') transliterated += (c == 'І' ? "I" : "i");

else if (c == 'Ї' || c == 'ї') transliterated += (isWordStart ? (c == 'Ї' ? "Yi" : "yi") : (c == 'Ї' ? "I" : "i"));

else if (c == 'Й' || c == 'й') transliterated += (isWordStart ? (c == 'Й' ? "Y" : "y") : (c == 'Й' ? "I" : "i"));

else if (c == 'К' || c == 'к') transliterated += (c == 'К' ? "K" : "k");

else if (c == 'Л' || c == 'л') transliterated += (c == 'Л' ? "L" : "l");

else if (c == 'М' || c == 'м') transliterated += (c == 'М' ? "M" : "m");

else if (c == 'Н' || c == 'н') transliterated += (c == 'Н' ? "N" : "n");

else if (c == 'О' || c == 'о') transliterated += (c == 'О' ? "O" : "o");

else if (c == 'П' || c == 'п') transliterated += (c == 'П' ? "P" : "p");

else if (c == 'Р' || c == 'р') transliterated += (c == 'Р' ? "R" : "r");

else if (c == 'С' || c == 'с') transliterated += (c == 'С' ? "S" : "s");

else if (c == 'Т' || c == 'т') transliterated += (c == 'Т' ? "T" : "t");

else if (c == 'У' || c == 'у') transliterated += (c == 'У' ? "U" : "u");

else if (c == 'Ф' || c == 'ф') transliterated += (c == 'Ф' ? "F" : "f");

else if (c == 'Х' || c == 'х') transliterated += (c == 'Х' ? "Kh" : "kh");

else if (c == 'Ц' || c == 'ц') transliterated += (c == 'Ц' ? "Ts" : "ts");

else if (c == 'Ч' || c == 'ч') transliterated += (c == 'Ч' ? "Ch" : "ch");

else if (c == 'Ш' || c == 'ш') transliterated += (c == 'Ш' ? "Sh" : "sh");

else if (c == 'Щ' || c == 'щ') transliterated += (c == 'Щ' ? "Shch" : "shch");

else if (c == 'Ь' || c == 'ь') transliterated += "";

else if (c == 'Ю' || c == 'ю') transliterated += (isWordStart ? (c == 'Ю' ? "Yu" : "yu") : (c == 'Ю' ? "Iu" : "iu"));

else if (c == 'Я' || c == 'я') transliterated += (isWordStart ? (c == 'Я' ? "Ya" : "ya") : (c == 'Я' ? "Ia" : "ia"));

else if (c == '\'') transliterated += "";

else transliterated += c;

isWordStart = !isalpha(static\_cast<unsigned char>(c)) && c != '\'';

}

time\_t now = time(nullptr);

char timeStr[100];

struct tm timeInfo;

localtime\_s(&timeInfo, &now);

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

output\_file << "Транслітерований текст:\n" << transliterated;

output\_file << "Дата й час дозапису: " << timeStr << "\n";

}

Заголовковий файл ModulesFedotov.h

void transliterateAndAppend(ifstream& input\_file, ofstream& output\_file);

**РЕЗУЛЬТАТ ВИКОНАННЯ ТЕСТОВОГО ДРАЙВЕРА:**

Тест кейс 1 | Файл: test1.txt

Вхідний вміст:

Привіт, як справи? Їжа та ґудзики!

Очікуваний вміст (префікс):

Привіт, як справи? Їжа та ґудзики!

Транслітерований текст:

Pryvit, yak spravy? Yizha ta gudzyky!

<дата й час>

Фактичний вміст:

Привіт, як справи? Їжа та ґудзики!

Транслітерований текст:

Pryvit, yak spravy? Yizha ta gudzyky!

Дата й час дозапису: 2025-04-25 21:56:11

Результат: PASSED

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

Тест кейс 2 | Файл: test2.txt

Вхідний вміст:

Євген Їв Юлія Яна

Очікуваний вміст (префікс):

Євген Їв Юлія Яна

Транслітерований текст:

Yevhen Yiv Yuliya Yana

<дата й час>

Фактичний вміст:

Євген Їв Юлія Яна

Транслітерований текст:

Yevhen Yiv Yuliya Yana

Дата й час дозапису: 2025-04-25 21:56:11

Результат: PASSED

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

Тест кейс 3 | Файл: test3.txt

Вхідний вміст:

Привіт, друже як справи?

Очікуваний вміст (префікс):

Привіт, друже як справи?

Транслітерований текст:

Pryvit, druzhe yak spravy?

<дата й час>

Фактичний вміст:

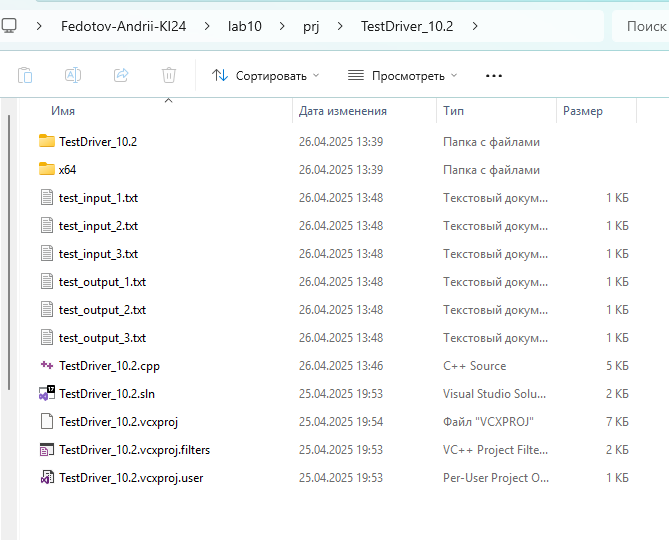
Привіт, друже як справи?

Транслітерований текст:

Pryvit, druzhe yak spravy?

Дата й час дозапису: 2025-04-25 21:56:11

Результат: PASSED



Малюнок 2 - Файли програми після запуску

**ЗАВДАННЯ №10.3**

Вхідні дані - числові значення х, у, z та натуральне число Ь. У вихідний

текстовий файл дописати:

* результати виконання функцій із заголовкового файлу ModulesПрiзeuщe.h

s\_calculation з аргументами х, у, z;

* число b у двійковому коді.

**ПОСТАНОВА ЗАДАЧІ**  
Потрібно дописати до вихідного текстового файлу результати виконання функції s\_calculation із заголовкового файлу ModulesПрiзeuщe.h, використовуючи вхідні числові значення x, y, z як аргументи. Крім того, необхідно подати натуральне число b у двійковому представленні.

**АНАЛІЗ ЗАДАЧІ**  
Задача включає обробку числових значень і роботу з зовнішньою функцією, що підключається з окремого заголовкового файлу. Для її виконання потрібно викликати функцію s\_calculation з трьома аргументами типу double або float. Одержаний результат слід записати у файл. Також потрібно реалізувати перетворення натурального числа b у двійковий код, що передбачає використання стандартних алгоритмів поділу на 2 з фіксацією залишку.

**РЕАЛІЗАЦІЯ БІБЛІОТЕКИ**

Вихідний файл ModulesFedotov.cpp

void writeCalculationAndBinary(ifstream& input\_file, ofstream& output\_file) {

double x, y, z;

unsigned int b;

if (!(input\_file >> x >> y >> z >> b)) {

output\_file << "Помилка: не вдалося прочитати вхідні дані.\n";

return;

}

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

string binary = "";

unsigned int original\_b = b;

if (b == 0) {

binary = "0";

}

else {

while (b > 0) {

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

b /= 2;

}

}

output\_file << "s\_calculation(" << fixed << setprecision(6) << x << ", "

<< y << ", " << z << ") = " << result << "\n";

output\_file << "Число " << original\_b << " у двійковому коді: " << binary << "\n";

}

Заголовковий файл ModulesFedotov.h

void writeCalculationAndBinary(ifstream& input\_file, ofstream& output\_file);

**РЕЗУЛЬТАТ ВИКОНАННЯ ТЕСТОВОГО ДРАЙВЕРА:**

Тест кейс 1 | Файл: test1.txt

Вхідні дані: x=1, y=2, z=3, b=5

Очікуваний вміст:

Початковий вміст

s\_calculation(1.000000, 2.000000, 3.000000) = 0.090703

Число 5 у двійковому коді: 101

Фактичний вміст:

Початковий вміст

s\_calculation(1.000000, 2.000000, 3.000000) = 0.090703

Число 5 у двійковому коді: 101

Результат: PASSED

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

Тест кейс 2 | Файл: test2.txt

Вхідні дані: x=-1.5, y=0.5, z=-2, b=0

Очікуваний вміст:

s\_calculation(-1.500000, 0.500000, -2.000000) = 1.147907

Число 0 у двійковому коді: 0

Фактичний вміст:

s\_calculation(-1.500000, 0.500000, -2.000000) = 1.147907

Число 0 у двійковому коді: 0

Результат: PASSED

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

Тест кейс 3 | Файл: test3.txt

Вхідні дані: x=0.75, y=-1.25, z=1.5, b=13

Очікуваний вміст:

Інший текст

s\_calculation(0.750000, -1.250000, 1.500000) = 0.377430

Число 13 у двійковому коді: 1101

Фактичний вміст:

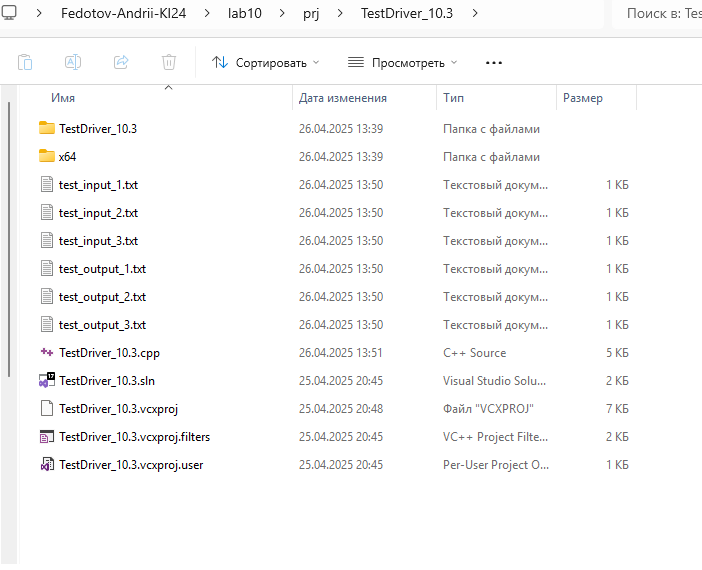
Інший текст

s\_calculation(0.750000, -1.250000, 1.500000) = 0.377430

Число 13 у двійковому коді: 1101

Результат: PASSED

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



Малюнок 3 - Файли програми після запуску

**ВІДПОВІДЬ НА КОНТРОЛЬНІ ЗАПИТАННЯ**

**1. Яке призначення та синтаксис запису блоку-контроля try - throw - catch у мові програмування С++?**  
Блок try використовується для написання коду, який потенційно може викликати помилку. Якщо виникає виключна ситуація, вона передається за допомогою оператора throw. Відповідний catch-блок перехоплює цю помилку і виконує обробку. Це дозволяє уникати аварійного завершення програми і грамотно керувати помилками.

**2. Наведіть приклад опису й використання міжмодульної змінної.**  
Міжмодульна змінна — це змінна, оголошена в одному модулі, але доступна для використання в інших. Для цього вона оголошується за допомогою ключового слова extern в заголовковому файлі, а визначається в одному з модулів програми.

**3. Яку область видимості матимуть об’єкти (змінні, типи, константи тощо), описані в тілі функції main С++?**  
Об’єкти, описані в тілі функції main, мають локальну область видимості, тобто вони доступні лише в межах цієї функції. Інші функції не мають до них доступу напряму.

**4. Здійсніть порівняльний аналіз змінної типу enum та масиву.**  
enum визначає перелік іменованих констант з числовими значеннями і призначений для роботи з фіксованим набором значень. Масив же є послідовністю елементів одного типу, доступних за індексом, і дозволяє зберігати змінні дані. enum економніший по пам’яті, масив — більш гнучкий для зберігання великих обсягів даних.

**5. Дайте визначення потоку й файлового потоку. Чим файловий потік відрізняється від інших потоків введення/виведення?**  
Потік — це абстракція, через яку відбувається передача даних. Файловий потік призначений для обміну інформацією з файлами. Відрізняється він від потоків, наприклад, вводу з клавіатури чи виводу на екран тим, що працює з постійним носієм інформації — файлами.

**6. Запропонуйте універсальний алгоритм реалізації мовою програмування С++ процесу читання/запису даних з/у файл за допомогою файлового потоку.**  
Відкрити файл у потрібному режимі, перевірити чи файл відкрився успішно, зчитувати або записувати дані відповідно до мети, закрити файл після завершення операцій.

**7. Який об’єм (у байтах) матиме текстовий файл, якщо в нього записано значення фундаментальної математичної константи π з точністю 106 десяткових цифр після коми?**  
Оскільки кожна цифра та крапка займає один байт у текстовому файлі, об’єм становитиме приблизно 107 байтів (1 байт на символ і 1 — на крапку).

**8. Яким чином у програмному забезпеченні можливо реалізувати перевірку статуса відкриття файлового потоку (відкрито/не відкрито) і в чому полягає її мета?**  
Після спроби відкриття файла перевіряється стан потоку. Це дозволяє переконатись, що файл існує або доступний для читання/запису. Якщо потік не відкрився, можна уникнути помилок при доступі до неіснуючого файла.

**9. Чим текстовий потік відрізняється від двійкового з погляду реалізації процесу оброблення даних із них?**  
У текстовому потоці дані обробляються у вигляді символів, з урахуванням форматування. У двійковому потоці дані зберігаються в тому вигляді, як вони представлені в пам’яті, що дозволяє зберігати структуровану інформацію ефективніше.

**10. Які розрізняють режими відкриття файлових потоків у С++?**  
Режими включають: для читання, для запису, для додавання, для читання і запису, у текстовому чи двійковому форматі. Їх можна комбінувати для досягнення потрібного результату.

**11. Класи яких файлових потоків реалізовано у fstream С++? Мовою програмування С++ наведіть приклад створення файлових об’єктів та відкриття відповідних потоків.**  
У fstream реалізовано три класи: ifstream для читання, ofstream для запису, та fstream для читання і запису. Вони дозволяють працювати з файлами відповідно до потреб програми.

**12. Перелічіть реалізовані в ios С++ константи режимів відкриття файлових потоків та призначення кожної з них.**  
Основні константи: in — відкриття для читання, out — для запису, app — додавання в кінець, trunc — очищення файла при відкритті, binary — відкриття в двійковому режимі, ate — позиціювання на кінець при відкритті.

**13. За допомогою яких функцій-членів файлових об’єктів fstream С++ реалізовується відкриття і закриття потоку, визначення кінця файлу?**  
Функції open() і close() відповідають за відкриття і закриття. Функція eof() визначає, чи досягнуто кінця файлу.

**14. Запропонуйте алгоритм видалення заданої послідовності символів (наприклад, слова) з текстового файла.**  
Прочитати вміст файла в оперативну пам’ять, замінити або видалити потрібну послідовність символів, перезаписати файл з оновленими даними.

**15. Перелічіть складові типи даних С/С++ та сформулюйте особливості оголошення й оброблення їх об’єктів (змінних) порівняно зі змінними простих типів.**  
Складові типи — це масиви, структури, об’єднання, класи. Вони можуть містити багато елементів різних типів, і вимагають більш складного підходу до ініціалізації, доступу та обробки, ніж прості типи.

**16. Яким чином у С/С++ здійснюється перевірка наявності елемента у множині, його додавання, вилучення?**  
Зазвичай реалізується за допомогою бітових масивів, логічних масивів або STL-контейнерів (наприклад, set). Для перевірки використовуються пошук, для додавання — вставка, для вилучення — видалення елемента.

**17. Як мовою програмування С/С++ реалізовується явне та неявне перетворення типів даних?**  
Неявне відбувається автоматично, коли типи сумісні. Явне виконується вручну, коли програміст вказує бажаний тип, наприклад, через приведення типів.

**18. Перелічіть випадки (задачі), у яких є доцільним використання змінних типу union. Наведіть приклад їх реалізації мовою С++.**  
Union доцільно використовувати, коли потрібно зберігати змінні різних типів, але не одночасно. Це економить пам’ять, наприклад, у структурах з параметрами різних типів, які використовуються по черзі.

**19. Запропонуйте алгоритм програмного модуля перевірки наявності у текстовому файлі заданого натурального числа.**  
Відкрити файл для читання, прочитати кожен рядок або слово, перевірити, чи дорівнює воно шуканому числу, при виявленні — завершити пошук з позитивним результатом.

**20. Що міститиме змінна symbol\_transaction при: short symbol\_transaction = sizeof short('R'); Відповідь поясніть та доведіть експериментально.**  
Змінна міститиме розмір типу short, а не символу 'R'. Запис некоректний, тому що sizeof short('R') не є допустимим синтаксисом. Правильний варіант: sizeof(short) або sizeof 'R', але останнє поверне розмір типу int, бо 'R' у C++ — це int-літерал.

**ВИСНОВКИ**

1. Засвоєно роботу з масивами в C++.
2. Навченося створювати та заповнювати структури.
3. Освоєно використання об’єднань.
4. Вивчено типи даних «перелік».
5. Опрацьовано поняття множин.
6. Засвоєно перетворення типів даних.
7. Набуто вміння роботи з символьними даними.
8. Вивчено основи файлового введення/виведення.
9. Освоєно відкриття і закриття файлів.
10. Навченося обробляти текстові файли.
11. Використано функції для запису у файли.
12. Навченося читати з файлів у різних форматах.
13. Засвоєно обробку помилок при роботі з файлами.
14. Освоєно роботу з динамічними масивами.
15. Вивчено обробку рядків з бібліотеки <string>.
16. Освоєно структурування коду в модулях.
17. Розвинуто навички логічного аналізу.
18. Навченося створювати користувацькі функції.
19. Закріплено вміння передавати аргументи у функції.
20. Практикували повернення значень з функцій.
21. Освоєно використання заголовкових файлів.
22. Засвоєно інкапсуляцію даних у структурах.
23. Навченося форматувати виведення.
24. Практично застосовано умовні оператори.
25. Використано цикли для обробки даних.
26. Вивчено вкладені цикли в масивах.
27. Освоєно сортування масивів.
28. Засвоєно пошук в масивах.
29. Вивчено копіювання та порівняння рядків.
30. Отримано досвід налагодження програм.
31. Засвоєно використання IDE Code::Blocks.
32. Освоєно компіляцію та запуск програм.
33. Навченося організовувати код за файлами.
34. Отримано навички створення заголовкових файлів.
35. Практикували інтеграцію декількох файлів у проєкті.
36. Засвоєно використання покажчиків у масивах.
37. Вивчено передачу структур у функції.
38. Освоєно роботу з символьними потоками.
39. Розширено розуміння типів даних у С++.
40. Практично вивчено оголошення змінних різних типів.
41. Засвоєно значення ініціалізації даних.
42. Вивчено механізм перетворення типів вручну та автоматично.
43. Набуто досвід роботи з typedef та enum.
44. Практично застосовано об’єднання типів.
45. Засвоєно структуру циклів for, while, do-while.
46. Навченося використовувати умовні оператори if, switch.
47. Вивчено бібліотеки <fstream>, <iostream>, <cstring>.
48. Практикували застосування логічних операторів.
49. Отримано досвід програмної обробки даних.
50. Поглиблено розуміння принципів модульного програмування.