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 ТИІВ 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 КБ-24

Заріцкий В. А.

ПЕРЕВІРИВ

викладач кафедри кібербезпеки   
та програмного забезпечення

Коваленко А. С.

Кропивницький – 2025

Мета роботи полягає у набутті ґрунтовних 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 №12**

**Завдання до лабораторної роботи**

1. Реалізувати програмні модулі розв’язування задач 10.1–10.3 як складові статичної бібліотеки libModulesZaritskyi.а (проект ModulesZaritskyi лабораторних робіт №8–9).
2. Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв’язування задач 10.1–10.3.

# 1. Аналіз задачі 10.1

## 1.1. Вимоги та постановка задачі

* **Мета**: Створити вихідний текстовий файл із записом авторської інформації, інформації про мову вхідного речення та обробленого речення.
* **Вхідні дані**: Текстовий файл, який містить довільне речення (як українською, так і англійською).
* **Обробка речення**:
  + **Визначення мови**: Проаналізувати вміст файла для визначення, чи переважають латинські символи (англійська) чи кириличні (українська).
  + **Обробка англійського речення**: Перетворити всі літери у верхній регістр.
  + **Обробка українського речення**: Видалити слова «лінощі», «сесія», «академзаборгованість» (з урахуванням можливих варіацій регістру).
* **Запис вихідних даних**:
  + Запис авторської інформації:

"Інформація про автора: Віктор Заріцкий, Центральноукраїнський Національний Технічний Університет, Кропивницький, Україна, 2025."

* + Запис інформації про мову вхідного речення.
  + Запис обробленого речення відповідно до визначеної логіки.
* **Особливості реалізації**:
  + Файл для запису створюється, якщо не існує, або очищається (перезаписується) перед записом.
  + Перевірка відкриття файлового потоку як для читання, так і для запису.

## 1.2. Архітектурне проектування (10.1)

* **Модуль читання файлу**:
  + Читає вхідне речення з файлу.
  + Перевірка успішного відкриття файлу.
* **Модуль аналізу тексту**:
  + Функція для аналізу символів: якщо більшість символів – латиниця, визначається англійська мова, інакше – українська.
  + Якщо англійський текст – застосовується функція конвертації до верхнього регістру.
  + Якщо український текст – застосовується функція пошуку та виключення слів "лінощі", "сесія", "академзаборгованість".
* **Модуль запису файлу**:
  + Створює або очищує вихідний файл.
  + Записує послідовно: авторські дані, інформацію про мову, оброблене речення.
  + Закриває файловий потік після запису.

## 1.3. Детальний псевдокод для 10.1

function processTask10\_1(inputFileName, outputFileName):

// Відкриття вхідного файлу для читання

open inputFileName as inFile for reading

if inFile не відкрився:

вивести повідомлення про помилку та завершити роботу

read sentence from inFile

close inFile

// Визначення мови речення

if sentence містить переважно латинські символи:

language = "англійська"

processedSentence = convertToUppercase(sentence)

else:

language = "українська"

processedSentence = removeForbiddenWords(sentence, ["лінощі", "сесія", "академзаборгованість"])

// Відкриття вихідного файлу для запису (створення/очищення)

open outputFileName as outFile for writing

if outFile не відкрився:

вивести повідомлення про помилку та завершити роботу

// Запис даних до вихідного файлу

write to outFile: "Розробник: Віктор Заріцкий, Центральноукраїнський Національний Технічний УніверсИТЕТ, Кропивницький, Україна, 2025."

write to outFile: "Мова речення: " + language

write to outFile: "Оброблене речення: " + processedSentence

close outFile

# 2. Аналіз задачі 10.2

## 2.1. Вимоги та постановка задачі

* **Мета**: Доповнити (дозаписати) вхідний текстовий файл додатковою інформацією.
* **Додаткова інформація**:
  + Текст статті 62 Закону України "Про вищу освіту". Цей текст може бути збережений як константа або завантажуватись із зовнішнього джерела.
  + Дата та час моменту дозапису (отримання локальної дати та часу через стандартні функції з <ctime>/<time.h>).
* **Режим роботи з файлом**:
  + Вхідний файл відкривається в режимі допису (append).
  + Потрібна перевірка відкриття потоку для дозапису.
  + Після операцій потік слід закрити.

## 2.2. Архітектурне проектування (10.2)

* **Модуль дозапису інформації**:
  + Функція для відкриття файлу в режимі append.
  + Функція отримання поточного часу у вигляді рядка.
  + Запис тексту статті 62 та рядка з датою й часом.
* **Обробка файлу**:
  + Відкриття файлу, перевірка успішного відкриття, дозапис та закриття потоку.

## 2.3. Детальний псевдокод для 10.2

function processTask10\_2(inputFileName):

// Відкриття файлу у режимі дозапису

open inputFileName as file in append mode

if file не відкрився:

вивести повідомлення про помилку та завершити роботу

// Завантаження тексту статті 62 Закону "Про вищу освіту"

textArticle = "Текст статті 62 Закону України 'Про вищу освіту'..."

// Отримання поточного часу

currentTime = getLocalTime()

// Запис інформації

write to file: textArticle

write to file: "Дата та час дозапису: " + currentTime

close file

# 3. Аналіз задачі 10.3

## 3.1. Вимоги та постановка задачі

* **Мета**: Доповнити вихідний текстовий файл результатами числових обчислень.
* **Вхідні дані**:
  + Три числові значення: **x**, **y**, **z**.
  + Натуральне число **b**.
* **Обчислення**:
  + Виклик функції s\_calculation(x, y, z) з заголовкового файлу ModulesZaritskyi для отримання певного результату.
  + Перетворення числа **y** до двійкового формату (через алгоритм послідовного ділення на 2 з накопиченням остач).
* **Вивід**:
  + Дописування до вихідного файлу результату виконання функції s\_calculation.
  + Дописування числа **y** у вигляді рядка, що є двійковим представленням.
* **Режим роботи з файлом**:
  + Вихідний файл відкривається в режимі дозапису (append) із відповідною перевіркою відкриття.

## 3.2. Архітектурне проектування (10.3)

* **Модуль числових обчислень**:
  + Функція для виклику s\_calculation з параметрами **x**, **y**, **z**.
  + Функція, яка приймає значення **y** і повертає рядок у двійковій формі.
* **Модуль запису числових результатів**:
  + Відкриття вихідного файлу (режим дозапису).
  + Запис результатів обчислень у форматовані повідомлення (наприклад, "Результат виконання s\_calculation(x, y, z) = ...", "Число y у двійковій системі числення: ...").

## 3.3. Детальний псевдокод для 10.3

function processTask10\_3(outputFileName, x, y, z, b):

// Виклик функції s\_calculation з ModulesZaritsky

resultCalculation = s\_calculation(x, y, z)

// Перетворення числа y до двійкового формату

binaryY = convertToBinary(y)

// Відкриття вихідного файлу для дозапису

open outputFileName as outFile in append mode

if outFile не відкрився:

вивести повідомлення про помилку та завершити роботу

// Запис результатів числових обчислень

write to outFile: "Результат виконання s\_calculation(" + x + ", " + y + ", " + z + ") = " + resultCalculation

write to outFile: "Число " + y + " у двійковій системі числення: " + binaryY

close outFile

## Алгоритм перетворення десяткового числа в двійкове:

function convertToBinary(num):

if num == 0:

return "0"

binaryRepresentation = ""

while num > 0:

remainder = num % 2

binaryRepresentation = remainder + binaryRepresentation // накопичення символів зліва направо

num = num // 2

return binaryRepresentation

# Результати автоматизованого модульного тестування:

Test: Task10\_1 Test 1

Вхідні дані: inputFileName = input\_en.txt, outputFileName = output.txt

Отриманий результат:

Розробник: Віктор Заріцкий, Центральноукраїнський Національний Технічний Університет, Кропивницький, Україна, 2025.

Мова речення: англійська

Оброблене речення: HELLO WORLD!

Статус тесту: passed

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

Test: Task10\_1 Test 2

Вхідні дані: inputFileName = input\_ua.txt, outputFileName = output.txt

Отриманий результат:

Розробник: Віктор Заріцкий, Центральноукраїнський Національний Технічний Університет, Кропивницький, Україна, 2025.

Мова речення: українська

Оброблене речення: Привіт студенте!

Статус тесту: passed

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

Test: Task10\_2 Test 1

Вхідні дані: fileName = input\_ua.txt

Отриманий результат:

Привіт студенте!

Стаття 62. Права осіб, які навчаються у закладах вищої освіти

1. Особи, які навчаються у закладах вищої освіти, мають право на:

1) вибір форми навчання під час вступу до закладу вищої освіти;

2) безпечні і нешкідливі умови навчання, праці та побуту;

3) трудову діяльність у позанавчальний час;

4) додаткову оплачувану відпустку у зв’язку з навчанням за основним місцем роботи, скорочений робочий час та інші пільги, передбачені законодавством для осіб, які поєднують роботу з навчанням;

5) безоплатне користування бібліотеками, інформаційними фондами, навчальною, науковою та спортивною базами закладу вищої освіти;

6) безоплатне забезпечення інформацією для навчання у доступних форматах з використанням технологій, що враховують обмеження життєдіяльності, зумовлені станом здоров’я (для осіб з особливими освітніми потребами);

7) користування виробничою, культурно-освітньою, побутовою, оздоровчою базами закладу вищої освіти у порядку, передбаченому статутом закладу вищої освіти;

8) забезпечення гуртожитком та цілодобовим доступом до нього на строк навчання у порядку, встановленому законодавством;

9) участь у науково-дослідних, дослідно-конструкторських роботах, конференціях, симпозіумах, виставках, конкурсах, представлення своїх робіт для публікації;

10) участь у заходах з освітньої, наукової, науково-дослідної, спортивної, мистецької, громадської діяльності, що проводяться в Україні та за кордоном, у встановленому законодавством порядку;

11) участь в обговоренні та вирішенні питань удосконалення навчального процесу, науково-дослідної роботи, призначення стипендій, організації дозвілля, побуту, оздоровлення;

12) внесення пропозицій щодо умов і розміру плати за навчання;

13) участь у громадських об’єднаннях;

14) участь у діяльності органів громадського самоврядування закладу вищої освіти, інститутів, факультетів, відділень, вченої ради закладу вищої освіти, органів студентського самоврядування;

15) вибір навчальних дисциплін у межах, передбачених відповідною освітньою програмою та навчальним планом, в обсязі, що становить не менш як 25 відсотків загальної кількості кредитів ЄКТС, передбачених для даного рівня вищої освіти. При цьому здобувачі певного рівня вищої освіти мають право вибирати навчальні дисципліни, що пропонуються для інших рівнів вищої освіти, за погодженням з керівником відповідного факультету чи підрозділу;

16) навчання одночасно за декількома освітніми програмами, а також у декількох закладах вищої освіти, за умови отримання тільки однієї вищої освіти за кожним ступенем за кошти державного (місцевого) бюджету;

17) академічну мобільність, у тому числі міжнародну;

18) отримання соціальної допомоги у випадках, встановлених законодавством;

19) зарахування до страхового стажу відповідно до Закону України "Про загальнообов’язкове державне пенсійне страхування" періодів навчання на денній формі навчання у закладах вищої освіти, аспірантурі, докторантурі, інтернатурі, резидентурі, за умови добровільної сплати страхових внесків;

20) академічну відпустку або перерву в навчанні із збереженням окремих прав здобувача вищої освіти, а також на поновлення навчання у порядку, встановленому центральним органом виконавчої влади у сфері освіти і науки;

21) участь у формуванні індивідуального навчального плану;

22) моральне та/або матеріальне заохочення за успіхи у навчанні, науково-дослідній і громадській роботі, за мистецькі та спортивні досягнення тощо;

23) захист від будь-яких форм експлуатації, фізичного та психічного насильства;

24) безоплатне проходження практики на підприємствах, в установах, закладах та організаціях, а також на оплату праці під час виконання виробничих функцій згідно із законодавством;

25) канікулярну відпустку тривалістю не менш як вісім календарних тижнів на навчальний рік;

26) отримання цільових пільгових державних кредитів для здобуття вищої освіти у порядку, визначеному Кабінетом Міністрів України;

27) оскарження дій органів управління закладу вищої освіти та їх посадових осіб, педагогічних і науково-педагогічних працівників;

28) спеціальний навчально-реабілітаційний супровід та вільний доступ до інфраструктури закладу вищої освіти відповідно до медико-соціальних показань за наявності обмежень життєдіяльності, зумовлених станом здоров’я.

2. Особи, які навчаються у закладах вищої освіти за денною формою навчання за рахунок коштів державного або місцевих бюджетів, мають право на отримання академічних та соціальних стипендій у встановленому законодавством порядку.

3. Особи, які навчаються у закладах вищої освіти за денною формою навчання, можуть отримувати інші стипендії, призначені фізичними (юридичними) особами.

4. Соціальні стипендії призначаються студентам (курсантам) закладу вищої освіти в порядку, встановленому Кабінетом Міністрів України.

Студенти (курсанти) закладу вищої освіти з числа дітей-сиріт та дітей, позбавлених батьківського піклування, а також студенти (курсанти) закладу вищої освіти, які в період навчання у віці від 18 до 23 років залишилися без батьків, мають гарантоване право на отримання соціальної стипендії, у тому числі у разі отримання академічної стипендії.

Академічні стипендії призначаються особам, які досягли значних успіхів у навчанні та/або науковій діяльності згідно з критеріями, встановленими Кабінетом Міністрів України. Частка студентів (курсантів), які мають право на отримання академічних стипендій, встановлюється вченою радою закладу вищої освіти у межах визначеного Кабінетом Міністрів України загального відсотка студентів (курсантів), які мають право на отримання академічних стипендій, та стипендіального фонду.

Студентам (курсантам) закладів вищої освіти, які мають право на отримання соціальної стипендії і набувають право на отримання академічної стипендії, надається один вид стипендії за їхнім вибором.

5. Розмір академічної та соціальної стипендій, порядок їх призначення і виплати встановлюються Кабінетом Міністрів України.

6. Для студентів (курсантів), які навчаються за гостродефіцитними спеціальностями (спеціалізаціями) (у галузях знань освіта, математичні, природничі, технічні науки), встановлюється підвищений розмір академічної стипендії. Перелік таких спеціальностей (спеціалізацій) та розмір підвищення визначаються Кабінетом Міністрів України.

{Частину сьому статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

{Частину восьму статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

9. Здобувачі вищої освіти, які навчаються у закладах вищої освіти за денною формою навчання, мають право на пільговий проїзд у транспорті у порядку, встановленому Кабінетом Міністрів України.

10. Студенти, курсанти закладів вищої освіти мають право на отримання студентського квитка, зразок якого затверджується центральним органом виконавчої влади у сфері освіти і науки.Дата та час дозапису: Wed Apr 9 09:31:11 2025

Статус тесту: passed

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

Test: Task10\_2 Test 2

Вхідні дані: fileName = input\_en.txt

Отриманий результат:

HELLO WORLD!

Стаття 62. Права осіб, які навчаються у закладах вищої освіти

1. Особи, які навчаються у закладах вищої освіти, мають право на:

1) вибір форми навчання під час вступу до закладу вищої освіти;

2) безпечні і нешкідливі умови навчання, праці та побуту;

3) трудову діяльність у позанавчальний час;

4) додаткову оплачувану відпустку у зв’язку з навчанням за основним місцем роботи, скорочений робочий час та інші пільги, передбачені законодавством для осіб, які поєднують роботу з навчанням;

5) безоплатне користування бібліотеками, інформаційними фондами, навчальною, науковою та спортивною базами закладу вищої освіти;

6) безоплатне забезпечення інформацією для навчання у доступних форматах з використанням технологій, що враховують обмеження життєдіяльності, зумовлені станом здоров’я (для осіб з особливими освітніми потребами);

7) користування виробничою, культурно-освітньою, побутовою, оздоровчою базами закладу вищої освіти у порядку, передбаченому статутом закладу вищої освіти;

8) забезпечення гуртожитком та цілодобовим доступом до нього на строк навчання у порядку, встановленому законодавством;

9) участь у науково-дослідних, дослідно-конструкторських роботах, конференціях, симпозіумах, виставках, конкурсах, представлення своїх робіт для публікації;

10) участь у заходах з освітньої, наукової, науково-дослідної, спортивної, мистецької, громадської діяльності, що проводяться в Україні та за кордоном, у встановленому законодавством порядку;

11) участь в обговоренні та вирішенні питань удосконалення навчального процесу, науково-дослідної роботи, призначення стипендій, організації дозвілля, побуту, оздоровлення;

12) внесення пропозицій щодо умов і розміру плати за навчання;

13) участь у громадських об’єднаннях;

14) участь у діяльності органів громадського самоврядування закладу вищої освіти, інститутів, факультетів, відділень, вченої ради закладу вищої освіти, органів студентського самоврядування;

15) вибір навчальних дисциплін у межах, передбачених відповідною освітньою програмою та навчальним планом, в обсязі, що становить не менш як 25 відсотків загальної кількості кредитів ЄКТС, передбачених для даного рівня вищої освіти. При цьому здобувачі певного рівня вищої освіти мають право вибирати навчальні дисципліни, що пропонуються для інших рівнів вищої освіти, за погодженням з керівником відповідного факультету чи підрозділу;

16) навчання одночасно за декількома освітніми програмами, а також у декількох закладах вищої освіти, за умови отримання тільки однієї вищої освіти за кожним ступенем за кошти державного (місцевого) бюджету;

17) академічну мобільність, у тому числі міжнародну;

18) отримання соціальної допомоги у випадках, встановлених законодавством;

19) зарахування до страхового стажу відповідно до Закону України "Про загальнообов’язкове державне пенсійне страхування" періодів навчання на денній формі навчання у закладах вищої освіти, аспірантурі, докторантурі, інтернатурі, резидентурі, за умови добровільної сплати страхових внесків;

20) академічну відпустку або перерву в навчанні із збереженням окремих прав здобувача вищої освіти, а також на поновлення навчання у порядку, встановленому центральним органом виконавчої влади у сфері освіти і науки;

21) участь у формуванні індивідуального навчального плану;

22) моральне та/або матеріальне заохочення за успіхи у навчанні, науково-дослідній і громадській роботі, за мистецькі та спортивні досягнення тощо;

23) захист від будь-яких форм експлуатації, фізичного та психічного насильства;

24) безоплатне проходження практики на підприємствах, в установах, закладах та організаціях, а також на оплату праці під час виконання виробничих функцій згідно із законодавством;

25) канікулярну відпустку тривалістю не менш як вісім календарних тижнів на навчальний рік;

26) отримання цільових пільгових державних кредитів для здобуття вищої освіти у порядку, визначеному Кабінетом Міністрів України;

27) оскарження дій органів управління закладу вищої освіти та їх посадових осіб, педагогічних і науково-педагогічних працівників;

28) спеціальний навчально-реабілітаційний супровід та вільний доступ до інфраструктури закладу вищої освіти відповідно до медико-соціальних показань за наявності обмежень життєдіяльності, зумовлених станом здоров’я.

2. Особи, які навчаються у закладах вищої освіти за денною формою навчання за рахунок коштів державного або місцевих бюджетів, мають право на отримання академічних та соціальних стипендій у встановленому законодавством порядку.

3. Особи, які навчаються у закладах вищої освіти за денною формою навчання, можуть отримувати інші стипендії, призначені фізичними (юридичними) особами.

4. Соціальні стипендії призначаються студентам (курсантам) закладу вищої освіти в порядку, встановленому Кабінетом Міністрів України.

Студенти (курсанти) закладу вищої освіти з числа дітей-сиріт та дітей, позбавлених батьківського піклування, а також студенти (курсанти) закладу вищої освіти, які в період навчання у віці від 18 до 23 років залишилися без батьків, мають гарантоване право на отримання соціальної стипендії, у тому числі у разі отримання академічної стипендії.

Академічні стипендії призначаються особам, які досягли значних успіхів у навчанні та/або науковій діяльності згідно з критеріями, встановленими Кабінетом Міністрів України. Частка студентів (курсантів), які мають право на отримання академічних стипендій, встановлюється вченою радою закладу вищої освіти у межах визначеного Кабінетом Міністрів України загального відсотка студентів (курсантів), які мають право на отримання академічних стипендій, та стипендіального фонду.

Студентам (курсантам) закладів вищої освіти, які мають право на отримання соціальної стипендії і набувають право на отримання академічної стипендії, надається один вид стипендії за їхнім вибором.

5. Розмір академічної та соціальної стипендій, порядок їх призначення і виплати встановлюються Кабінетом Міністрів України.

6. Для студентів (курсантів), які навчаються за гостродефіцитними спеціальностями (спеціалізаціями) (у галузях знань освіта, математичні, природничі, технічні науки), встановлюється підвищений розмір академічної стипендії. Перелік таких спеціальностей (спеціалізацій) та розмір підвищення визначаються Кабінетом Міністрів України.

{Частину сьому статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

{Частину восьму статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

9. Здобувачі вищої освіти, які навчаються у закладах вищої освіти за денною формою навчання, мають право на пільговий проїзд у транспорті у порядку, встановленому Кабінетом Міністрів України.

10. Студенти, курсанти закладів вищої освіти мають право на отримання студентського квитка, зразок якого затверджується центральним органом виконавчої влади у сфері освіти і науки.Дата та час дозапису: Wed Apr 9 09:31:11 2025

Статус тесту: passed

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

Test: Task10\_3 Test 1

Вхідні дані: outputFileName = output.txt, x = 5, y = 1, z = 2, b = 5

Отриманий результат:

Попередній текст (як результат задач 10.1/10.2)

Результат виконання s\_calculation(5, 1, 2) = 1.20824

Число 1 у двійковій системі числення: 1

Статус тесту: passed

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

Test: Task10\_3 Test 2

Вхідні дані: outputFileName = output.txt, x = 10, y = 0, z = -4, b = 1

Отриманий результат:

Попередній текст (як результат задач 10.1/10.2)

Результат виконання s\_calculation(10, 0, -4) = -1.99745

Число 0 у двійковій системі числення: 0

Статус тесту: passed

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

# Вихідний код проекту ModulesZaritskyi:

#include <iostream>

#include <fstream>

#include <sstream>

#include <string>

#include <ctime>

#include <algorithm>

#include <cctype>

#include <cmath>

#include "ModulesZaritskyi.h"

using namespace std;

double s\_calculation(double x, double y, double z) {

// Перевірка області визначення: x має бути більше 4, щоб логарифм був додатнім.

if (x <= 4.0) {

return NAN; // або можна сигналізувати про помилку іншим способом

}

double logValue = log10(x - 3.0);

// Якщо логарифм не додатній, обчислення неможливе

if (logValue <= 0.0) {

return NAN;

}

// Обчислення виразу під квадратним коренем

double underSqrt = (12.0 \* x \* x) / logValue;

// Перевірка, що підкореневий вираз не від'ємний

if (underSqrt < 0.0) {

return NAN;

}

// Обчислення S згідно з формулою

double result = pow(3.0 \* sin(sqrt(underSqrt)), 2.0) + 0.5 \* z;

return result;

}

double calculateElectricityBill(double kWh) {

double rate;

if (kWh <= 150)

rate = 0.130843;

else if (kWh <= 800)

rate = 0.241945;

else

rate = 0.534047;

return kWh \* rate;

}

Temperature calculateAverageTemperature(const double temps[12]) {

double sum = 0.0;

for (int i = 0; i < 12; i++) {

sum += temps[i];

}

double avgC = sum / 12.0;

Temperature result;

result.celsius = avgC;

result.fahrenheit = 32 + (9.0 / 5.0) \* avgC;

return result;

}

int countBits(int N) {

if (N == 0) {

return 1;

}

bool bit10 = (N & (1 << 10)) != 0;

int count = 0;

while (N > 0) {

int bit = N & 1;

// Якщо bit10 == false – підраховуємо нулі, інакше одиниці.

count += (bit10 ? (bit == 1 ? 1 : 0) : (bit == 0 ? 1 : 0));

N >>= 1;

}

return count;

}

// Допоміжна функція: перетворення рядка у верхній регістр (англійське речення)

std::string convertToUppercase(const std::string &s) {

std::string result = s;

std::transform(result.begin(), result.end(), result.begin(), ::toupper);

return result;

}

// Допоміжна функція: перевірка, чи символ є латинською буквою

bool isLatinLetter(char c) {

return ( (c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') );

}

// Допоміжна функція: визначення мови речення

// Якщо кількість латинських символів перевищує кількість інших алфавітних символів,

// вважаємо, що мова – англійська; інакше – українська.

std::string determineLanguage(const std::string &s) {

int latinCount = 0;

int otherLetterCount = 0;

for (char c : s) {

if (std::isalpha(static\_cast<unsigned char>(c))) {

if (isLatinLetter(c))

latinCount++;

else

otherLetterCount++;

}

}

return (latinCount > otherLetterCount) ? "англійська" : "українська";

}

// Допоміжна функція: видалення заборонених слів із українського речення

// Заборонені слова: "лінощі", "сесія", "академзаборгованість"

std::string removeForbiddenWords(const std::string &s) {

std::string result = s;

// Перелік заборонених слів (можна розширювати за потребою; регістр порівнюємо у нижньому)

const std::string forbiddenWords[] = {"лінощі", "сесія", "академзаборгованість"};

// Для спрощення обробки переводимо текст у нижній регістр для пошуку,

// але при видаленні з оригінального тексту працюємо зі знайденими позиціями.

std::string lowerResult = s;

std::transform(lowerResult.begin(), lowerResult.end(), lowerResult.begin(), ::tolower);

// Для кожного забороненого слова виконуємо пошук і видалення

for (const std::string &word : forbiddenWords) {

size\_t pos = lowerResult.find(word);

while (pos != std::string::npos) {

// Видалення слова з оригінального рядка

result.erase(pos, word.length());

// Також з нижнього рядка – для коректності пошуку наступного входження

lowerResult.erase(pos, word.length());

pos = lowerResult.find(word);

}

}

return result;

}

// Допоміжна функція: перетворення десяткового числа у двійковий рядок

std::string convertToBinary(int num) {

if (num == 0)

return "0";

std::string binaryRepresentation;

while (num > 0) {

binaryRepresentation.insert(binaryRepresentation.begin(), (num % 2) + '0');

num /= 2;

}

return binaryRepresentation;

}

// Реалізація задачі 10.1

// Функція читає вхідний файл, визначає мову тексту, обробляє речення та записує результати у вихідний файл.

void processTask10\_1(const std::string &inputFileName, const std::string &outputFileName) {

std::ifstream inFile(inputFileName);

if (!inFile) {

std::cerr << "Помилка: не вдалося відкрити файл для читання: " << inputFileName << std::endl;

return;

}

std::string sentence;

std::getline(inFile, sentence);

inFile.close();

// Визначення мови речення

std::string language = determineLanguage(sentence);

std::string processedSentence;

if (language == "англійська") {

processedSentence = convertToUppercase(sentence);

} else {

processedSentence = removeForbiddenWords(sentence);

}

// Відкриття вихідного файлу для запису (створення/очищення)

std::ofstream outFile(outputFileName);

if (!outFile) {

std::cerr << "Помилка: не вдалося відкрити файл для запису: " << outputFileName << std::endl;

return;

}

// Запис даних: авторська інформація, мова речення, оброблене речення

outFile << "Розробник: Віктор Заріцкий, Центральноукраїнський Національний Технічний Університет, Кропивницький, Україна, 2025.\n";

outFile << "Мова речення: " << language << "\n";

outFile << "Оброблене речення: " << processedSentence << "\n";

outFile.close();

}

// Реалізація задачі 10.2

// Функція дозаписує до заданого файлу текст статті 62 Закону України "Про вищу освіту" та поточну дату й час.

void processTask10\_2(const std::string &inputFileName) {

// Відкриття файлу у режимі дозапису

std::ofstream file(inputFileName, std::ios::app);

if (!file) {

std::cerr << "Помилка: не вдалося відкрити файл для дозапису: " << inputFileName << std::endl;

return;

}

// Текст статті 62 Закону України "Про вищу освіту"

const std::string textArticle = R"(Стаття 62. Права осіб, які навчаються у закладах вищої освіти

1. Особи, які навчаються у закладах вищої освіти, мають право на:

1) вибір форми навчання під час вступу до закладу вищої освіти;

2) безпечні і нешкідливі умови навчання, праці та побуту;

3) трудову діяльність у позанавчальний час;

4) додаткову оплачувану відпустку у зв’язку з навчанням за основним місцем роботи, скорочений робочий час та інші пільги, передбачені законодавством для осіб, які поєднують роботу з навчанням;

5) безоплатне користування бібліотеками, інформаційними фондами, навчальною, науковою та спортивною базами закладу вищої освіти;

6) безоплатне забезпечення інформацією для навчання у доступних форматах з використанням технологій, що враховують обмеження життєдіяльності, зумовлені станом здоров’я (для осіб з особливими освітніми потребами);

7) користування виробничою, культурно-освітньою, побутовою, оздоровчою базами закладу вищої освіти у порядку, передбаченому статутом закладу вищої освіти;

8) забезпечення гуртожитком та цілодобовим доступом до нього на строк навчання у порядку, встановленому законодавством;

9) участь у науково-дослідних, дослідно-конструкторських роботах, конференціях, симпозіумах, виставках, конкурсах, представлення своїх робіт для публікації;

10) участь у заходах з освітньої, наукової, науково-дослідної, спортивної, мистецької, громадської діяльності, що проводяться в Україні та за кордоном, у встановленому законодавством порядку;

11) участь в обговоренні та вирішенні питань удосконалення навчального процесу, науково-дослідної роботи, призначення стипендій, організації дозвілля, побуту, оздоровлення;

12) внесення пропозицій щодо умов і розміру плати за навчання;

13) участь у громадських об’єднаннях;

14) участь у діяльності органів громадського самоврядування закладу вищої освіти, інститутів, факультетів, відділень, вченої ради закладу вищої освіти, органів студентського самоврядування;

15) вибір навчальних дисциплін у межах, передбачених відповідною освітньою програмою та навчальним планом, в обсязі, що становить не менш як 25 відсотків загальної кількості кредитів ЄКТС, передбачених для даного рівня вищої освіти. При цьому здобувачі певного рівня вищої освіти мають право вибирати навчальні дисципліни, що пропонуються для інших рівнів вищої освіти, за погодженням з керівником відповідного факультету чи підрозділу;

16) навчання одночасно за декількома освітніми програмами, а також у декількох закладах вищої освіти, за умови отримання тільки однієї вищої освіти за кожним ступенем за кошти державного (місцевого) бюджету;

17) академічну мобільність, у тому числі міжнародну;

18) отримання соціальної допомоги у випадках, встановлених законодавством;

19) зарахування до страхового стажу відповідно до Закону України "Про загальнообов’язкове державне пенсійне страхування" періодів навчання на денній формі навчання у закладах вищої освіти, аспірантурі, докторантурі, інтернатурі, резидентурі, за умови добровільної сплати страхових внесків;

20) академічну відпустку або перерву в навчанні із збереженням окремих прав здобувача вищої освіти, а також на поновлення навчання у порядку, встановленому центральним органом виконавчої влади у сфері освіти і науки;

21) участь у формуванні індивідуального навчального плану;

22) моральне та/або матеріальне заохочення за успіхи у навчанні, науково-дослідній і громадській роботі, за мистецькі та спортивні досягнення тощо;

23) захист від будь-яких форм експлуатації, фізичного та психічного насильства;

24) безоплатне проходження практики на підприємствах, в установах, закладах та організаціях, а також на оплату праці під час виконання виробничих функцій згідно із законодавством;

25) канікулярну відпустку тривалістю не менш як вісім календарних тижнів на навчальний рік;

26) отримання цільових пільгових державних кредитів для здобуття вищої освіти у порядку, визначеному Кабінетом Міністрів України;

27) оскарження дій органів управління закладу вищої освіти та їх посадових осіб, педагогічних і науково-педагогічних працівників;

28) спеціальний навчально-реабілітаційний супровід та вільний доступ до інфраструктури закладу вищої освіти відповідно до медико-соціальних показань за наявності обмежень життєдіяльності, зумовлених станом здоров’я.

2. Особи, які навчаються у закладах вищої освіти за денною формою навчання за рахунок коштів державного або місцевих бюджетів, мають право на отримання академічних та соціальних стипендій у встановленому законодавством порядку.

3. Особи, які навчаються у закладах вищої освіти за денною формою навчання, можуть отримувати інші стипендії, призначені фізичними (юридичними) особами.

4. Соціальні стипендії призначаються студентам (курсантам) закладу вищої освіти в порядку, встановленому Кабінетом Міністрів України.

Студенти (курсанти) закладу вищої освіти з числа дітей-сиріт та дітей, позбавлених батьківського піклування, а також студенти (курсанти) закладу вищої освіти, які в період навчання у віці від 18 до 23 років залишилися без батьків, мають гарантоване право на отримання соціальної стипендії, у тому числі у разі отримання академічної стипендії.

Академічні стипендії призначаються особам, які досягли значних успіхів у навчанні та/або науковій діяльності згідно з критеріями, встановленими Кабінетом Міністрів України. Частка студентів (курсантів), які мають право на отримання академічних стипендій, встановлюється вченою радою закладу вищої освіти у межах визначеного Кабінетом Міністрів України загального відсотка студентів (курсантів), які мають право на отримання академічних стипендій, та стипендіального фонду.

Студентам (курсантам) закладів вищої освіти, які мають право на отримання соціальної стипендії і набувають право на отримання академічної стипендії, надається один вид стипендії за їхнім вибором.

5. Розмір академічної та соціальної стипендій, порядок їх призначення і виплати встановлюються Кабінетом Міністрів України.

6. Для студентів (курсантів), які навчаються за гостродефіцитними спеціальностями (спеціалізаціями) (у галузях знань освіта, математичні, природничі, технічні науки), встановлюється підвищений розмір академічної стипендії. Перелік таких спеціальностей (спеціалізацій) та розмір підвищення визначаються Кабінетом Міністрів України.

{Частину сьому статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

{Частину восьму статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

9. Здобувачі вищої освіти, які навчаються у закладах вищої освіти за денною формою навчання, мають право на пільговий проїзд у транспорті у порядку, встановленому Кабінетом Міністрів України.

10. Студенти, курсанти закладів вищої освіти мають право на отримання студентського квитка, зразок якого затверджується центральним органом виконавчої влади у сфері освіти і науки.)";

// Отримання поточного часу

std::time\_t now = std::time(nullptr);

std::string currentTime = std::ctime(&now); // ctime повертає рядок з символом нового рядка в кінці

// Дописування інформації до файлу

file << textArticle;

file << "Дата та час дозапису: " << currentTime;

file.close();

}

// Реалізація задачі 10.3

// Функція викликає s\_calculation для заданих чисел, перетворює число y до двійкового формату та дозаписує результати до вихідного файлу.

void processTask10\_3(const std::string &outputFileName, double x, double y, double z, int b) {

// Виклик функції s\_calculation з модуля ModulesZaritsky

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

// Перетворення числа y до двійкового формату

std::string binaryY = convertToBinary(y);

// Відкриття вихідного файлу для дозапису

std::ofstream outFile(outputFileName, std::ios::app);

if (!outFile) {

std::cerr << "Помилка: не вдалося відкрити файл для дозапису: " << outputFileName << std::endl;

return;

}

// Запис результатів числових обчислень

outFile << "Результат виконання s\_calculation(" << x << ", " << y << ", " << z << ") = " << resultCalculation << "\n";

outFile << "Число " << y << " у двійковій системі числення: " << binaryY << "\n";

outFile.close();

}

# Вихідний код проекту Test Driver:

#include <iostream>

#include <fstream>

#include <sstream>

#include <string>

#include <ctime>

#include <vector>

#include <algorithm>

#include <cctype>

#include "ModulesZaritskyi.h"

using namespace std;

// Структура тест-кейсу для задач 10.1

struct TestCase10\_1 {

string inputFileName;

string outputFileName;

string inputContent;

string expectedOutput; // очікується точна відповідність тексту

};

// Структура тест-кейсу для задач 10.2 (порівнюємо наявність статті у тексті)

struct TestCase10\_2 {

string fileName;

string initialContent;

string expectedSubstring; // хоча б частина тексту, що має бути дописана

};

// Структура тест-кейсу для задач 10.3

struct TestCase10\_3 {

string outputFileName;

double x;

double y;

double z;

int b;

string expectedSubstring; // наприклад, результат виконання s\_calculation та двійкове представлення

};

// Функція зчитування вмісту файлу у рядок

string readFileContent(const string &fileName) {

ifstream file(fileName);

if (!file)

return "";

ostringstream ss;

ss << file.rdbuf();

return ss.str();

}

// Функція запису логів тестування

void logTestResult(ofstream &logFile, const string &testName, const string &inputArgs,

const string &obtainedResult, bool passed) {

logFile << "Test: " << testName << "\n";

logFile << "Вхідні дані: " << inputArgs << "\n";

logFile << "Отриманий результат:\n" << obtainedResult << "\n";

logFile << "Статус тесту: " << (passed ? "passed" : "failed") << "\n";

logFile << "---------------------------------------------\n";

}

int main() {

// Відкриваємо лог-файл для протоколювання тестування

ofstream logFile("test\_log.txt", ios::out);

if (!logFile) {

cerr << "Не вдалося відкрити файл test\_log.txt для запису логів.\n";

return 1;

}

// ========================

// ТЕСТИ ЗАДАЧІ 10.1

// ========================

vector<TestCase10\_1> tests10\_1 = {

{

"input\_en.txt",

"output.txt",

"Hello world!",

"Розробник: Віктор Заріцкий, Центральноукраїнський Національний Технічний Університет, Кропивницький, Україна, 2025.\n"

"Мова речення: англійська\n"

"Оброблене речення: HELLO WORLD!\n"

},

{

"input\_ua.txt",

"output.txt",

"Привіт студенте!",

"Розробник: Віктор Заріцкий, Центральноукраїнський Національний Технічний Університет, Кропивницький, Україна, 2025.\n"

"Мова речення: українська\n"

"Оброблене речення: Привіт студенте!\n"

}

// Додаткові приклади можна додати за потребою.

};

for (size\_t i = 0; i < tests10\_1.size(); i++) {

// Preliminary Steps: створення вхідного файлу з даними

{

ofstream inFile(tests10\_1[i].inputFileName);

inFile << tests10\_1[i].inputContent;

inFile.close();

}

// Очистка або створення вихідного файлу

{

ofstream outFile(tests10\_1[i].outputFileName);

outFile.close();

}

// Дії: виклик функції processTask10\_1

processTask10\_1(tests10\_1[i].inputFileName, tests10\_1[i].outputFileName);

// Отримання результату

string obtained = readFileContent(tests10\_1[i].outputFileName);

bool passed = (obtained == tests10\_1[i].expectedOutput);

// Протоколювання

stringstream inputArgs;

inputArgs << "inputFileName = " << tests10\_1[i].inputFileName

<< ", outputFileName = " << tests10\_1[i].outputFileName;

logTestResult(logFile, "Task10\_1 Test " + to\_string(i+1), inputArgs.str(), obtained, passed);

}

// ========================

// ТЕСТИ ЗАДАЧІ 10.2

// ========================

vector<TestCase10\_2> tests10\_2 = {

{

"input\_ua.txt",

"Привіт студенте!\n",

R"(Стаття 62. Права осіб, які навчаються у закладах вищої освіти

1. Особи, які навчаються у закладах вищої освіти, мають право на:

1) вибір форми навчання під час вступу до закладу вищої освіти;

2) безпечні і нешкідливі умови навчання, праці та побуту;

3) трудову діяльність у позанавчальний час;

4) додаткову оплачувану відпустку у зв’язку з навчанням за основним місцем роботи, скорочений робочий час та інші пільги, передбачені законодавством для осіб, які поєднують роботу з навчанням;

5) безоплатне користування бібліотеками, інформаційними фондами, навчальною, науковою та спортивною базами закладу вищої освіти;

6) безоплатне забезпечення інформацією для навчання у доступних форматах з використанням технологій, що враховують обмеження життєдіяльності, зумовлені станом здоров’я (для осіб з особливими освітніми потребами);

7) користування виробничою, культурно-освітньою, побутовою, оздоровчою базами закладу вищої освіти у порядку, передбаченому статутом закладу вищої освіти;

8) забезпечення гуртожитком та цілодобовим доступом до нього на строк навчання у порядку, встановленому законодавством;

9) участь у науково-дослідних, дослідно-конструкторських роботах, конференціях, симпозіумах, виставках, конкурсах, представлення своїх робіт для публікації;

10) участь у заходах з освітньої, наукової, науково-дослідної, спортивної, мистецької, громадської діяльності, що проводяться в Україні та за кордоном, у встановленому законодавством порядку;

11) участь в обговоренні та вирішенні питань удосконалення навчального процесу, науково-дослідної роботи, призначення стипендій, організації дозвілля, побуту, оздоровлення;

12) внесення пропозицій щодо умов і розміру плати за навчання;

13) участь у громадських об’єднаннях;

14) участь у діяльності органів громадського самоврядування закладу вищої освіти, інститутів, факультетів, відділень, вченої ради закладу вищої освіти, органів студентського самоврядування;

15) вибір навчальних дисциплін у межах, передбачених відповідною освітньою програмою та навчальним планом, в обсязі, що становить не менш як 25 відсотків загальної кількості кредитів ЄКТС, передбачених для даного рівня вищої освіти. При цьому здобувачі певного рівня вищої освіти мають право вибирати навчальні дисципліни, що пропонуються для інших рівнів вищої освіти, за погодженням з керівником відповідного факультету чи підрозділу;

16) навчання одночасно за декількома освітніми програмами, а також у декількох закладах вищої освіти, за умови отримання тільки однієї вищої освіти за кожним ступенем за кошти державного (місцевого) бюджету;

17) академічну мобільність, у тому числі міжнародну;

18) отримання соціальної допомоги у випадках, встановлених законодавством;

19) зарахування до страхового стажу відповідно до Закону України "Про загальнообов’язкове державне пенсійне страхування" періодів навчання на денній формі навчання у закладах вищої освіти, аспірантурі, докторантурі, інтернатурі, резидентурі, за умови добровільної сплати страхових внесків;

20) академічну відпустку або перерву в навчанні із збереженням окремих прав здобувача вищої освіти, а також на поновлення навчання у порядку, встановленому центральним органом виконавчої влади у сфері освіти і науки;

21) участь у формуванні індивідуального навчального плану;

22) моральне та/або матеріальне заохочення за успіхи у навчанні, науково-дослідній і громадській роботі, за мистецькі та спортивні досягнення тощо;

23) захист від будь-яких форм експлуатації, фізичного та психічного насильства;

24) безоплатне проходження практики на підприємствах, в установах, закладах та організаціях, а також на оплату праці під час виконання виробничих функцій згідно із законодавством;

25) канікулярну відпустку тривалістю не менш як вісім календарних тижнів на навчальний рік;

26) отримання цільових пільгових державних кредитів для здобуття вищої освіти у порядку, визначеному Кабінетом Міністрів України;

27) оскарження дій органів управління закладу вищої освіти та їх посадових осіб, педагогічних і науково-педагогічних працівників;

28) спеціальний навчально-реабілітаційний супровід та вільний доступ до інфраструктури закладу вищої освіти відповідно до медико-соціальних показань за наявності обмежень життєдіяльності, зумовлених станом здоров’я.

2. Особи, які навчаються у закладах вищої освіти за денною формою навчання за рахунок коштів державного або місцевих бюджетів, мають право на отримання академічних та соціальних стипендій у встановленому законодавством порядку.

3. Особи, які навчаються у закладах вищої освіти за денною формою навчання, можуть отримувати інші стипендії, призначені фізичними (юридичними) особами.

4. Соціальні стипендії призначаються студентам (курсантам) закладу вищої освіти в порядку, встановленому Кабінетом Міністрів України.

Студенти (курсанти) закладу вищої освіти з числа дітей-сиріт та дітей, позбавлених батьківського піклування, а також студенти (курсанти) закладу вищої освіти, які в період навчання у віці від 18 до 23 років залишилися без батьків, мають гарантоване право на отримання соціальної стипендії, у тому числі у разі отримання академічної стипендії.

Академічні стипендії призначаються особам, які досягли значних успіхів у навчанні та/або науковій діяльності згідно з критеріями, встановленими Кабінетом Міністрів України. Частка студентів (курсантів), які мають право на отримання академічних стипендій, встановлюється вченою радою закладу вищої освіти у межах визначеного Кабінетом Міністрів України загального відсотка студентів (курсантів), які мають право на отримання академічних стипендій, та стипендіального фонду.

Студентам (курсантам) закладів вищої освіти, які мають право на отримання соціальної стипендії і набувають право на отримання академічної стипендії, надається один вид стипендії за їхнім вибором.

5. Розмір академічної та соціальної стипендій, порядок їх призначення і виплати встановлюються Кабінетом Міністрів України.

6. Для студентів (курсантів), які навчаються за гостродефіцитними спеціальностями (спеціалізаціями) (у галузях знань освіта, математичні, природничі, технічні науки), встановлюється підвищений розмір академічної стипендії. Перелік таких спеціальностей (спеціалізацій) та розмір підвищення визначаються Кабінетом Міністрів України.

{Частину сьому статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

{Частину восьму статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

9. Здобувачі вищої освіти, які навчаються у закладах вищої освіти за денною формою навчання, мають право на пільговий проїзд у транспорті у порядку, встановленому Кабінетом Міністрів України.

10. Студенти, курсанти закладів вищої освіти мають право на отримання студентського квитка, зразок якого затверджується центральним органом виконавчої влади у сфері освіти і науки.)"

},

{

"input\_en.txt",

"HELLO WORLD!\n",

R"(Стаття 62. Права осіб, які навчаються у закладах вищої освіти

1. Особи, які навчаються у закладах вищої освіти, мають право на:

1) вибір форми навчання під час вступу до закладу вищої освіти;

2) безпечні і нешкідливі умови навчання, праці та побуту;

3) трудову діяльність у позанавчальний час;

4) додаткову оплачувану відпустку у зв’язку з навчанням за основним місцем роботи, скорочений робочий час та інші пільги, передбачені законодавством для осіб, які поєднують роботу з навчанням;

5) безоплатне користування бібліотеками, інформаційними фондами, навчальною, науковою та спортивною базами закладу вищої освіти;

6) безоплатне забезпечення інформацією для навчання у доступних форматах з використанням технологій, що враховують обмеження життєдіяльності, зумовлені станом здоров’я (для осіб з особливими освітніми потребами);

7) користування виробничою, культурно-освітньою, побутовою, оздоровчою базами закладу вищої освіти у порядку, передбаченому статутом закладу вищої освіти;

8) забезпечення гуртожитком та цілодобовим доступом до нього на строк навчання у порядку, встановленому законодавством;

9) участь у науково-дослідних, дослідно-конструкторських роботах, конференціях, симпозіумах, виставках, конкурсах, представлення своїх робіт для публікації;

10) участь у заходах з освітньої, наукової, науково-дослідної, спортивної, мистецької, громадської діяльності, що проводяться в Україні та за кордоном, у встановленому законодавством порядку;

11) участь в обговоренні та вирішенні питань удосконалення навчального процесу, науково-дослідної роботи, призначення стипендій, організації дозвілля, побуту, оздоровлення;

12) внесення пропозицій щодо умов і розміру плати за навчання;

13) участь у громадських об’єднаннях;

14) участь у діяльності органів громадського самоврядування закладу вищої освіти, інститутів, факультетів, відділень, вченої ради закладу вищої освіти, органів студентського самоврядування;

15) вибір навчальних дисциплін у межах, передбачених відповідною освітньою програмою та навчальним планом, в обсязі, що становить не менш як 25 відсотків загальної кількості кредитів ЄКТС, передбачених для даного рівня вищої освіти. При цьому здобувачі певного рівня вищої освіти мають право вибирати навчальні дисципліни, що пропонуються для інших рівнів вищої освіти, за погодженням з керівником відповідного факультету чи підрозділу;

16) навчання одночасно за декількома освітніми програмами, а також у декількох закладах вищої освіти, за умови отримання тільки однієї вищої освіти за кожним ступенем за кошти державного (місцевого) бюджету;

17) академічну мобільність, у тому числі міжнародну;

18) отримання соціальної допомоги у випадках, встановлених законодавством;

19) зарахування до страхового стажу відповідно до Закону України "Про загальнообов’язкове державне пенсійне страхування" періодів навчання на денній формі навчання у закладах вищої освіти, аспірантурі, докторантурі, інтернатурі, резидентурі, за умови добровільної сплати страхових внесків;

20) академічну відпустку або перерву в навчанні із збереженням окремих прав здобувача вищої освіти, а також на поновлення навчання у порядку, встановленому центральним органом виконавчої влади у сфері освіти і науки;

21) участь у формуванні індивідуального навчального плану;

22) моральне та/або матеріальне заохочення за успіхи у навчанні, науково-дослідній і громадській роботі, за мистецькі та спортивні досягнення тощо;

23) захист від будь-яких форм експлуатації, фізичного та психічного насильства;

24) безоплатне проходження практики на підприємствах, в установах, закладах та організаціях, а також на оплату праці під час виконання виробничих функцій згідно із законодавством;

25) канікулярну відпустку тривалістю не менш як вісім календарних тижнів на навчальний рік;

26) отримання цільових пільгових державних кредитів для здобуття вищої освіти у порядку, визначеному Кабінетом Міністрів України;

27) оскарження дій органів управління закладу вищої освіти та їх посадових осіб, педагогічних і науково-педагогічних працівників;

28) спеціальний навчально-реабілітаційний супровід та вільний доступ до інфраструктури закладу вищої освіти відповідно до медико-соціальних показань за наявності обмежень життєдіяльності, зумовлених станом здоров’я.

2. Особи, які навчаються у закладах вищої освіти за денною формою навчання за рахунок коштів державного або місцевих бюджетів, мають право на отримання академічних та соціальних стипендій у встановленому законодавством порядку.

3. Особи, які навчаються у закладах вищої освіти за денною формою навчання, можуть отримувати інші стипендії, призначені фізичними (юридичними) особами.

4. Соціальні стипендії призначаються студентам (курсантам) закладу вищої освіти в порядку, встановленому Кабінетом Міністрів України.

Студенти (курсанти) закладу вищої освіти з числа дітей-сиріт та дітей, позбавлених батьківського піклування, а також студенти (курсанти) закладу вищої освіти, які в період навчання у віці від 18 до 23 років залишилися без батьків, мають гарантоване право на отримання соціальної стипендії, у тому числі у разі отримання академічної стипендії.

Академічні стипендії призначаються особам, які досягли значних успіхів у навчанні та/або науковій діяльності згідно з критеріями, встановленими Кабінетом Міністрів України. Частка студентів (курсантів), які мають право на отримання академічних стипендій, встановлюється вченою радою закладу вищої освіти у межах визначеного Кабінетом Міністрів України загального відсотка студентів (курсантів), які мають право на отримання академічних стипендій, та стипендіального фонду.

Студентам (курсантам) закладів вищої освіти, які мають право на отримання соціальної стипендії і набувають право на отримання академічної стипендії, надається один вид стипендії за їхнім вибором.

5. Розмір академічної та соціальної стипендій, порядок їх призначення і виплати встановлюються Кабінетом Міністрів України.

6. Для студентів (курсантів), які навчаються за гостродефіцитними спеціальностями (спеціалізаціями) (у галузях знань освіта, математичні, природничі, технічні науки), встановлюється підвищений розмір академічної стипендії. Перелік таких спеціальностей (спеціалізацій) та розмір підвищення визначаються Кабінетом Міністрів України.

{Частину сьому статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

{Частину восьму статті 62 виключено на підставі Закону № 1774-VIII від 06.12.2016}

9. Здобувачі вищої освіти, які навчаються у закладах вищої освіти за денною формою навчання, мають право на пільговий проїзд у транспорті у порядку, встановленому Кабінетом Міністрів України.

10. Студенти, курсанти закладів вищої освіти мають право на отримання студентського квитка, зразок якого затверджується центральним органом виконавчої влади у сфері освіти і науки.)"

}

};

for (size\_t i = 0; i < tests10\_2.size(); i++) {

// Preliminary Steps: створення файлу з початковим вмістом

{

ofstream file(tests10\_2[i].fileName);

file << tests10\_2[i].initialContent;

file.close();

}

// Дія: виклик processTask10\_2

processTask10\_2(tests10\_2[i].fileName);

// Зчитування вмісту файлу

string obtained = readFileContent(tests10\_2[i].fileName);

// Перевіримо чи містить отриманий текст очікуваний підрядок

bool passed = (obtained.find(tests10\_2[i].expectedSubstring) != string::npos);

stringstream inputArgs;

inputArgs << "fileName = " << tests10\_2[i].fileName;

logTestResult(logFile, "Task10\_2 Test " + to\_string(i+1), inputArgs.str(), obtained, passed);

}

// ========================

// ТЕСТИ ЗАДАЧІ 10.3

// ========================

vector<TestCase10\_3> tests10\_3 = {

{

"output.txt", 5, 1, 2, 5,

"Результат виконання s\_calculation(5, 1, 2) = 1.20824"

// Очікуємо рядок з "Число 10 у двійковій системі числення: 1010"

},

{

"output.txt", 10, 0, -4, 1,

"Результат виконання s\_calculation(10, 0, -4) = -1.99745"

// Очікується, що "Число 0 у двійковій системі числення: 0"

}

};

for (size\_t i = 0; i < tests10\_3.size(); i++) {

// Preliminary Steps: створення або очищення вихідного файлу (якщо потрібно, з попередніми даними)

{

ofstream outFile(tests10\_3[i].outputFileName);

// Для тестування задач 10.3 можна записати попередню інформацію, якщо вона потрібна

outFile << "Попередній текст (як результат задач 10.1/10.2)\n";

outFile.close();

}

// Дія: виклик processTask10\_3

processTask10\_3(tests10\_3[i].outputFileName, tests10\_3[i].x, tests10\_3[i].y, tests10\_3[i].z, tests10\_3[i].b);

// Зчитування файлу

string obtained = readFileContent(tests10\_3[i].outputFileName);

bool passed = (obtained.find(tests10\_3[i].expectedSubstring) != string::npos);

stringstream inputArgs;

inputArgs << "outputFileName = " << tests10\_3[i].outputFileName

<< ", x = " << tests10\_3[i].x << ", y = " << tests10\_3[i].y

<< ", z = " << tests10\_3[i].z << ", b = " << tests10\_3[i].b;

logTestResult(logFile, "Task10\_3 Test " + to\_string(i+1), inputArgs.str(), obtained, passed);

}

cout << "Тестування завершено. Результати записано у файл test\_log.txt.\n";

logFile.close();

return 0;

}







# Аргументи досягнення мети:

1. Розбиття завдань на окремі модулі (10.1, 10.2, 10.3) забезпечило чіткий розподіл функціональних обов’язків. Це дозволяє легше тестувати та підтримувати код.
2. Правильне оголошення функцій у заголовковому файлі (ModulesZaritskyi.h) сприяло узгодженості інтерфейсу модулів. Це гарантує, що всі модулі використовують єдиний контракт обміну даними.
3. Побудова бібліотеки libModulesZaritskyi.a підтверджує здатність перетворювати функціональні модулі у повторно використовувані компоненти. Такий підхід спрощує інтеграцію та повторне використання коду.
4. Використання стандартних класів ifstream та ofstream забезпечило коректну обробку файлів введення/виведення. Забезпечується правильний доступ до даних із зовнішніх файлів.
5. Реалізація перевірки успішного відкриття файлових потоків запобігає виникненню критичних помилок у виконанні програми. Це дозволяє своєчасно виявити і обробити помилки I/O.
6. Функції для обробки тексту (перетворення у верхній регістр, видалення заборонених слів) дозволяють гнучко маніпулювати вхідними даними. Такі операції демонструють практичне використання стандартних алгоритмів обробки рядків.
7. Функція визначення мови, що порівнює кількість латинських та кириличних символів, дозволяє автоматично адаптувати подальшу обробку. Це сприяє коректній обробці англійського та українського тексту.
8. Функція перетворення тексту в upper-case для англомовних рядків підтверджує здатність програми модифікувати дані згідно специфікації завдання.  
   Такий підхід демонструє використання стандартних бібліотечних алгоритмів.
9. Розроблена функція видалення певних слів із українського речення демонструє наявність алгоритмічних рішень для фільтрації тексту.  
   Фільтрація здійснюється з урахуванням регістру, що підвищує точність обробки.
10. Виклик функції s\_calculation демонструє обчислювальні можливості розробленого модуля із застосуванням математичних операцій. Реалізація обчислювальної логіки перевіряється на прикладах з різними вхідними даними.
11. Функція convertToBinary показує вміння програми працювати з різними системами числення. Це демонструє математичну гнучкість рішення.
12. Програма активно використовує бібліотеки <fstream>, <string>, <ctime>, що показує володіння базовими методологіями С++. Стандартний функціонал спрощує розробку та підвищує портативність.
13. Реалізація ведення логів у файлі дозволяє аналізувати вхідні дані, отримані результати та статус тест-кейсів. Протоколювання допомагає відслідковувати якість роботи модулів.
14. Розробка тестового драйвера для автоматичного запуску тест-кейсів підтверджує ефективність модульного тестування. Це знижує людський фактор при перевірці якості коду.
15. Формалізовані тест-кейси у вигляді константних масивів демонструють системний підхід до перевірки роботи функцій. Завчасно сформовані набір тестів допомагають охопити різні сценарії використання.
16. Передача імен файлів та числових значень через параметри функцій демонструє здатність до параметризованого виконання. Це дозволяє легко адаптувати програму до змін вимог.
17. Наявність можливості налаштування параметрів компіляції та лінкування забезпечує інтеграцію з зовнішніми бібліотеками. Використання опцій build дозволяє оптимізувати проект під конкретну платформу.
18. Логіка створення/очищення файлів у залежності від їхньої наявності гарантує коректне зберігання даних. Це запобігає помилкам при повторному запуску програмного модуля.
19. Отримання поточного часу за допомогою функції ctime демонструє застосування системних ресурсів для ведення журналу подій. Це дозволяє використовувати дані часу для аналізу продуктивності.
20. Реалізація завдань безпосередньо відповідає умовам лабораторної роботи, що визначені у методичних рекомендаціях. Чіткий зв’язок між вимогами і кодом гарантує виконання мети.
21. Включення коментарів та пояснень у коді сприяє легшому розумінню логіки роботи програми. Детальна документація полегшує подальшу підтримку проєкту.
22. Перевірка коректності вхідних даних перед обробкою забезпечує безпечне виконання операцій. Це запобігає використанню некоректних або пошкоджених файлів.
23. Створення окремих файлів для збереження результатів роботи модулів дозволяє зберігати історію виконання. Це корисно для відлагодження та повторного аналізу даних.
24. Програмні модулі спроектовані так, щоб їх легко було доповнювати новою логікою або тест-кейсами. Такий підхід забезпечує масштабованість проєкту.
25. Використання єдиного стилю програмування, оголошень та форматування сприяє підтримці чіткості коду. Це зменшує помилки і збільшує продуктивність розробки.
26. Автоматичне порівняння отриманих результатів із заздалегідь підготовленими очікуваними значеннями підвищує надійність тестування. Це дозволяє виявити неточності та логічні помилки у реалізації модулів.
27. Відокремлення функціональності від інтерфейсу сприяє перевірці кожного модуля окремо. Це дозволяє локалізувати і виправляти помилки швидко.
28. Збереження тестових даних у масивах забезпечує їх повторне використання і однозначність вхідних даних. Це спрощує автоматизацію процесу тестування.
29. Логування статусу тест-кейсів (passed/failed) дозволяє швидко визначити якість роботи модулів. Це полегшує аналіз якості програмного забезпечення.
30. Застосування stringstream дозволяє формувати чисті і зрозумілі повідомлення про вхідні аргументи. Це сприяє однозначній інтерпретації даних у логах.
31. Використання результатів лабораторних робіт № 8–9 забезпечує безперервність навчання та використання перевірених рішень. Це підтверджує інтеграцію попередніх знань з новими задачами.
32. Розроблені модулі і тестовий драйвер можна адаптувати під реальні сценарії роботи з файлами. Практична орієнтація завдань розвиває вміння розробляти промислове ПЗ.
33. Виведення повідомлень про помилки у разі невдалого відкриття файлових потоків забезпечує стійкість застосунку. Це дозволяє користувачу одразу зрозуміти, яка частина програми потребує корегування.
34. Функції з бібліотеки <algorithm>, <ctime> та <cmath> демонструють використання сучасних підходів до програмування. Це сприяє підвищенню продуктивності коду та його зрозумілості.
35. Алгоритмічне прийняття рішень (if-else) використовується для обробки різних сценаріїв роботи з даними. Це гарантує адаптивність програми до різних умов виконання.
36. Використання цього інтегрованого середовища розвитку сприяє кращій організації проектів і налагодженню коду. Це дозволяє ефективно працювати у команді та підтримувати якість розробки.
37. Наявність Git-репозиторію забезпечує контроль версій і збереження історії змін проекту. Це критично важливо для сучасного програмування і співпраці в команді.
38. Регулярне проведення unit-тестування сприяє ранньому виявленню помилок. Це зменшує час виправлення дефектів і підвищує надійність ПЗ.
39. Всі завдання були розроблені відповідно до методичних рекомендацій, що свідчить про їх відповідність заданій меті. Дотримання стандартів підвищує якість підготовки звіту.
40. Ретельно оформлені коментарі допомагають зрозуміти логіку реалізації кожної функції. Це полегшує повторне використання і модифікацію коду.
41. Наявність окремих тек для проекту, тест-сьютів, звітів сприяє організації робочого простору. Зручна організація робочих файлів полегшує подальшу підтримку проекту.
42. Функції мають чітку і зрозумілу логіку, що легко перевіряється за допомогою тестів. Це дозволяє швидко оцінити відповідність реалізації поставленим завданням.
43. Обробка даних здійснюється із використанням параметрів, що дозволяє змінювати вхідні значення без зміни логіки. Це полегшує адаптацію програми до різних сценаріїв використання.
44. Використання стандартних потоків дозволяє обробляти як текстові, так і числові дані. Це свідчить про універсальність реалізованих модулів.
45. Система спроектована таким чином, що можна легко додати нові модулі обробки даних. Це гарантує довготривалу актуальність та розширюваність рішення.
46. Алгоритмічна логіка забезпечує динамічну обробку даних і перевірку умов безпеки. Це підвищує стабільність роботи програми в умовах змінних вхідних параметрів.
47. Кожна функція реалізована у вигляді окремої одиниці, що можна використовувати незалежно від інших. Це сприяє можливості повторного використання компонентів у інших проєктах.
48. Тестові драйвери охоплюють різноманітні вхідні дані (як англомовні, так і українські тексти, числові значення). Це гарантує повноту перевірки коректності роботи функцій.
49. Автоматичне створення вхідних файлів і порівняння результатів дозволяє скоротити ручну роботу тестувальника. Це підвищує ефективність роботи та знижує ризик людської помилки.
50. Повна реалізація вимог лабораторної роботи (створення модульних програм, тестування, протоколювання, інтеграція з Git) підтверджує досягнення мети роботи. Цей підсумковий аргумент свідчить, що усі технологічні та методичні вимоги виконані, а отримані результати відповідають завданням лабораторної роботи.

# Відповіді на контрольні запитання:

**1. Призначення та синтаксис блоку try–throw–catch у C++**

Блоки try–throw–catch використовуються для обробки виняткових ситуацій (exception handling). Вони дозволяють «кидати» (throw) винятки у місцях, де виникають проблеми, і «ловити» (catch) їх у спеціальних блоках для обробки помилок, не припиняючи роботу програми аварійно.  
**Синтаксис:**

try {

// Блок коду, де може статися помилка

if (/\* деяка умова \*/) {

throw std::runtime\_error("Виникла помилка!");

}

// Інший код, який виконується, якщо виняток не кинуто

} catch (const std::exception &ex) {

// Обробка винятку

std::cerr << "Помилка: " << ex.what() << std::endl;

}

Блок try містить код, який потенційно може згенерувати виняток. Оператор throw викликає виняток певного типу (наприклад, std::runtime\_error), а блок catch перехоплює його та здійснює обробку (визначення шляху відновлення, запис логу тощо).

**2. Приклад опису й використання міжмодульної змінної**

Міжмодульна змінна дозволяє обмінюватися даними між різними файлами (модулями) програми. Для цього її оголошують як extern у заголовковому файлі, а визначають у одному з cpp-файлів.

Приклад:  
У заголовковому файлі (Globals.h):

#ifndef GLOBALS\_H

#define GLOBALS\_H

extern int globalCounter;

#endif // GLOBALS\_H

У одному з модулів (Globals.cpp):

#include "Globals.h"

int globalCounter = 0;

У іншому модулі:

#include "Globals.h"

#include <iostream>

void incrementCounter() {

globalCounter++;

std::cout << "globalCounter: " << globalCounter << std::endl;

}

Оголошення за допомогою extern дозволяє декільком модулям бачити одну і ту ж змінну, визначену в одному місці.

**3. Область видимості об’єктів, оголошених в тілі функції main**

Об’єкти (змінні, типи, константи тощо), оголошені всередині функції main(), мають **локальну (блокову) область видимості**.

Такі об’єкти доступні тільки в межах функції main і після виходу з неї автоматично знищуються. Вони не видно в інших функціях чи модулях.

**4. Порівняльний аналіз змінної типу enum та масиву**

* **Enum (перелічення):**
  + Використовується для визначення символічних імен для константних цілих значень.
  + Значення зазвичай визначаються на етапі компіляції.
  + Наприклад:

enum Day { Monday, Tuesday, Wednesday, Thursday, Friday };

* + **Переваги:** зручність при читанні коду, підвищення безпеки типів.
* **Масив:**
  + Представляє собою послідовність елементів одного типу, організовану в пам’яті.
  + Забезпечує зберігання множини значень, доступ до яких можливий за індексом.
  + Наприклад:
  + int numbers[5] = {1, 2, 3, 4, 5};
  + **Переваги:** ефективна організація набору даних, можливість ітерації та маніпуляції.

Enum використовується для визначення фіксованого набору константних значень, тоді як масив – це контейнер для зберігання змінних значень, доступ до яких здійснюється за допомогою індексів.

**5. Визначення потоку й файлового потоку; відмінність файлового потоку**

* **Потік (stream):** Абстракція для послідовної передачі даних (якщо це потік вводу, виводу або як те, що використовується для обробки даних у пам’яті).
* **Файловий потік:** Тип потоку, що забезпечує зв’язок між програмою та файловою системою для читання з файлів або запису у файли.

Файловий потік працює з дисковими файлами (або іншими зовнішніми сховищами), тоді як інші потоки (наприклад, std::cin, std::cout) зазвичай працюють зі стандартним введенням/виведенням (клавіатурою, консолью). Файловий потік має специфічні режими відкриття (відкриття для читання, запису, дозапису, тощо).

**6. Універсальний алгоритм читання/запису з/у файл за допомогою файлового потоку**

1. **Відкрити файловий потік:**
   * Для читання використати std::ifstream.
   * Для запису – std::ofstream (або std::fstream з відповідним режимом).
2. **Перевірити, чи потік успішно відкритий:**

if (!file.is\_open()) { /\* обробка помилки \*/ }

1. **Виконати операції читання або запису:**
   * Наприклад, зчитати весь вміст файлу, використовуючи std::getline() або оператор >>.
2. **Закрити потік після завершення операцій:**

file.close();

Цей алгоритм гарантує, що файл буде відкритий коректно, дані – прочитані або записані, а після завершення робіт – звільнені ресурси.

**7. Об’єм текстового файлу, що містить π з точністю 106 десяткових цифр після коми**

Припускаючи, що записується число у форматі:

* Одна цифра перед десятковою крапкою (наприклад, «3»)
* Одна десяткова крапка
* 106 цифр після коми

Тоді загальна кількість символів = 1+1+1 000 000=1 000 002.

Оскільки кожен символ займає 1 байт (за типовою кодуванням ASCII), розмір файлу складе приблизно 1 000 002 байти.

**8. Перевірка статусу відкриття файлового потоку: реалізація та мета**

Перевірка здійснюється за допомогою методу .is\_open() або перевірки оператора приведення до логічного типу:

std::ifstream file("filename.txt");

if (!file) { /\* обробка помилки \*/ }

**Мета:** Переконатися, що файл відкрито успішно, перш ніж виконувати операції читання чи запису. Це допомагає уникнути аварійного завершення програми через спробу роботи з неіснуючим файлом.

**9. Відмінність текстового потоку від двійкового**

* **Текстовий потік:** Обробляє дані як послідовність символів із можливими конвертаціями (наприклад, перетворення символів нового рядка, локальні кодування).
* **Двійковий потік:** Обробляє дані як послідовність байтів без будь-якої конвертації.

У текстових потоках символи можуть бути змінені (наприклад, конвертація «\n» в залежності від ОС), тоді як у двійкових даних інформація зберігається точно так, як записано.

**10. Режими відкриття файлових потоків у C++**

Основні режими:

* ios::in – відкриття файлу для читання.
* ios::out – відкриття файлу для запису.
* ios::app – дозапис; записи додаються в кінець файлу.
* ios::ate – відкриття з переходом в кінець файлу відразу після відкриття.
* ios::trunc – обрізання файлу (видалення попереднього вмісту) при відкритті для запису.
* ios::binary – відкриття у двійковому режимі (без конвертації символів).

Ці режими дозволяють контролювати поведінку файлового потоку залежно від задачі.

**11. Класи файлових потоків у fstream і приклад створення об’єктів**

У C++ визначено три основні класи:

* std::ifstream – потік для читання з файлу.
* std::ofstream – потік для запису в файл.
* std::fstream – потік для читання і запису.

Приклад:

#include <fstream>

#include <string>

int main() {

std::ifstream inFile("input.txt", std::ios::in);

std::ofstream outFile("output.txt", std::ios::out);

std::fstream file("data.txt", std::ios::in | std::ios::out | std::ios::binary);

if (!inFile || !outFile || !file) {

// обробка помилки

}

// Використання потоків...

inFile.close();

outFile.close();

file.close();

return 0;

}

**12. Константи режимів відкриття файлових потоків та їх призначення (у просторі імен std::ios або std::ios\_base):**

* ios::in: Відкриття файлу для читання.
* ios::out: Відкриття файлу для запису.
* ios::app: Відкриття файлу для дозапису (дані додаються в кінець).
* ios::ate: Відкриття файлу з переходом у кінець (може використовуватися для одразу позиціювання).
* ios::trunc: Обрізання файлу при відкритті для запису (видалення попереднього вмісту).
* ios::binary: Відкриття файлу у двійковому режимі (без текстових конвертацій).

Кожна константа визначає, яким чином буде оброблятися файл при відкритті, що важливо для коректної роботи з даними.

**13. Функції-члени для відкриття, закриття та визначення кінця файлу у fstream**

* open(): Відкриття файлу.
* close(): Закриття відкритого файлу.
* eof(): Перевірка, чи досягнуто кінця файлу (повертає true, якщо кінець файлу досягнуто).

Ці функції використовуються для контролю над роботою з файловими потоками.

**14. Алгоритм видалення заданої послідовності символів (слова) з текстового файлу**

1. Відкрити файл для читання (за допомогою std::ifstream).
2. Зчитати весь вміст файлу у рядок або контейнер.
3. Використовуючи функції (наприклад, std::string::find і std::string::erase), знайти та видалити усі входження заданого слова.
4. Закрити вхідний потік.
5. Відкрити файл для запису (за допомогою std::ofstream з режимом ios::trunc).
6. Записати оновлений рядок назад у файл.
7. Закрити файловий потік.

Такий підхід дозволяє обробити весь вміст файлу і зберегти його без необхідних підрядків.

**15. Складові типи даних у C/C++ і особливості об’явлення/обробки їх об’єктів**

Основні типи:

* **Прості (базові) типи:** int, float, double, char, bool.
* **Агрегатні типи:**
  + **Масиви:** Контейнер фіксованої довжини, що містить елементи одного типу.
  + **Структури (struct):** Об’єднують декілька полів (можуть бути різних типів) в єдиний тип.
  + **Об’єднання (union):** Зберігають різні типи даних в одній області пам’яті; в один момент часу використовується лише одне з полів.
  + **Класи:** Подібні до структур, але з можливістю визначення методів та доступу (public, private).
* **Перелічення (enum):** Набір константних цілих значень із символічними іменами.
* **Вказівники та посилання:** Забезпечують роботу з адресами в пам’яті.

Агрегатні типи часто вимагають ініціалізації всіх компонентів, можуть мати вкладені типи, а їх обробка здійснюється через посилання чи цикли – на відміну від простих типів, що зберігаються як окреме значення.

**16. Перевірка наявності елемента у множині, його додавання та вилучення**

У C++ для роботи з множинами використовується стандартний контейнер std::set. Методи:

* **Перевірка:** set.find(element) повертає ітератор – якщо елемент знайдено.
* **Додавання:** set.insert(element).
* **Видалення:** set.erase(element).

Ці методи забезпечують швидкий та ефективний доступ до даних завдяки реалізації на базі дерев або хеш-таблиць.

**17. Явне та неявне перетворення типів у C/C++**

* **Неявне (implicit) перетворення:** Виконується автоматично компілятором, якщо можливе перетворення без втрати даних (наприклад, з int в float).
* **Явне (explicit) перетворення:** Використання синтаксису приведення типів, наприклад:
* double d = 3.14;
* int i = (int)d; // C-стиль
* int j = static\_cast<int>(d); // C++ стиль

Явне перетворення використовується для чіткого сигналу компілятору, що певна операція перетворення має бути виконана, незважаючи на можливі зміни значення.

**18. Випадки доцільного використання union та приклад реалізації**

**Union** дозволяє зберігати дані різних типів у спільній області пам’яті – корисно, коли потрібно економити пам’ять або працювати зі змінними, тип яких визначається динамічно.

Приклад:

#include <iostream>

using namespace std;

union Data {

int i;

float f;

char c;

};

int main() {

Data data;

data.i = 10;

cout << "Ціле: " << data.i << endl;

data.f = 3.14f;

cout << "Число з плавальною комою: " << data.f << endl;

data.c = 'A';

cout << "Символ: " << data.c << endl;

return 0;

}

Хоча всі дані зберігаються в одній області, використання union дозволяє економити пам’ять, проте водночас робить програму вразливою до некоректного доступу, якщо читати неактуальні значення.

**19. Алгоритм перевірки наявності заданого натурального числа у текстовому файлі**

1. **Відкрити файл** для читання (ifstream).
2. **Зчитати вміст** файлу (рядок або построчно).
3. **Парсити** рядки, використовуючи стандартні функції (наприклад, std::stoi або регулярні вирази) для пошуку натуральних чисел.
4. **Порівняти** знайдене число з заданим числом.
5. **Повернути результат:** якщо число знайдено – повернути true, інакше – false.

Такий модуль може бути використаний для пошуку шаблонів у тексті та забезпечення валідації даних.

**20. Що міститиме змінна symbol\_transaction при виразі:**

short symbol\_transaction = sizeof short('R');

**Аналіз:**

* Використання оператора sizeof визначає розмір типу. Тут бажано було написати:
* short symbol\_transaction = sizeof(short);
  + або з приведенням, проте форма sizeof short('R') може бути трактована як: 'R' – літерал типу char (або, за деякими стандартами, може мати тип int), але якщо його привести до типу short, то sizeof поверне розмір типу short.

У більшості систем тип short займає 2 байти. Тож:

short symbol\_transaction = sizeof(short); // ймовірно 2

Експериментально, якщо написати програму, що виводить значення sizeof(short), отримаємо (зазвичай) 2, що означає, що змінна symbol\_transaction міститиме число 2 (тобто 2 байти).  
Приклад:

#include <iostream>

using namespace std;

int main() {

std::cout << "Розмір short: " << sizeof(short) << " байт." << std::endl;

return 0;

}

При компіляції й запуску, скоріше за все, буде виведено:

Розмір short: 2 байт.