

**ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ФІЗИКО-ТЕХНІЧНИХ  
ТА КОМП'ЮТЕРНИХ НАУК  
КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ**

**Шумиляк Л.М., Газдюк К.П., Тарновецька О.Ю.**

**ОСНОВИ ПРОГРАМУВАННЯ,  
СТРУКТУРИ ДАНИХ ТА  
АЛГОРИТМИ  
НАВЧАЛЬНИЙ ПОСІБНИК  
(ЛАБОРАТОРНИЙ ПРАКТИКУМ)**

**ЧЕРНІВЦІ – 2024**

УДК 004.42:519.85  
О-76

*Друкується за ухвалою  
Вченої ради навчально-наукового інституту  
фізико-технічних та комп'ютерних наук  
Чернівецького національного університету імені Юрія Федьковича  
(протокол №2 від 22 лютого 2024 року)*

Шумиляк Л.М., Газдюк К.П., Тарновецька О.Ю.  
Основи програмування, структури даних та алгоритми (Лабораторний практикум) / Л.М. Шумиляк, К.П.Газдюк, О.Ю. Тарновецька – Чернівці: ЧНУ, 2024. – 76 с.

У пропонованому виданні викладено навчальний посібник дисципліни «Основи програмування, структури даних та алгоритми», який читається на першому році бакалаврату за освітньою програмою «Програмне забезпечення систем». Запропоновано навчальну програму курсу та лабораторний практикум.

Для студентів спеціальностей: 121 – «Інженерія програмного забезпечення», 122 – «Комп'ютерні науки» та 126 – «Інформаційні системи і технології» усіх форм навчання, інших спеціальностей, де вивчаються дисципліни, пов'язані з програмуванням та алгоритмізацією, а також для самостійного опанування основами курсу.

УДК 004.42:519.85

## ЗМІСТ

|   |    |
|---|----|
| ПЕРЕДМОВА .....   | 4  |
| СИЛАБУС.....  | 5  |
| МОДУЛЬ 1. ОСНОВИ ПРОГРАМУВАННЯ МОВОЮ C++<br>ТА АЛГОРИТМІЗАЦІЯ ..... | 11 |
| Лабораторна робота №1 .....   | 11 |
| Лабораторна робота № 2 .....  | 17 |
| Лабораторна робота № 3 .....  | 20 |
| Лабораторна робота № 4 .....  | 26 |
| Лабораторна робота № 5 .....  | 29 |
| Лабораторна робота № 6 .....  | 32 |
| Лабораторна робота № 7 .....  | 35 |
| Лабораторна робота № 8 .....  | 39 |
| МОДУЛЬ 2. СТРУКТУРИ ДАНИХ МОВОЮ C++.....                            | 45 |
| Лабораторна робота № 9 .....  | 45 |
| Лабораторна робота № 10 .....                                       | 52 |
| Лабораторна робота № 11 .....                                       | 55 |
| Лабораторна робота № 12 .....                                       | 63 |
| Лабораторна робота № 13 .....                                       | 69 |

## ПЕРЕДМОВА

Лабораторний практикум «Основи програмування, структури даних та алгоритми» є невід'ємною складовою академічного процесу, спрямованого на формування фундаментальних навичок у сфері інформаційних технологій. Цей практикум призначений для студентів, які вивчають основи програмування, структури даних та алгоритми, щоб надати їм практичні навички та глибше розуміння ключових концепцій цих областей.

Мета цього практикуму – не лише надати студентам можливість практичного застосування теоретичних знань, а й розвинути їхні аналітичні та проблемно-розв'язувальні навички. Завдяки виконанню лабораторних робіт, студенти матимуть можливість ефективно використовувати різноманітні структури даних, розробляти алгоритми та практично реалізовувати програмні рішення.

Практикум охоплює широкий спектр тем, починаючи з основних понять програмування, таких як змінні, типи даних, умови та цикли, і переходячи до більш глибокого розгляду принципів роботи з динамічною пам'яттю та структурами даних. Розгляд цих тем допоможе студентам не лише розвинути навички програмування, а й розуміти принципи ефективної організації даних та оптимізації алгоритмів.

Кожна лабораторна робота в цьому практикумі охоплює важливі аспекти програмування, від базових концепцій до більш складних задач. До кожного завдання додається відповідний теоретичний матеріал, що допомагає студентам засвоїти нові знання та зрозуміти їхню практичну вартість.

Виділяючи важливі поняття та надаючи практичні приклади, практикум створений таким чином, щоб сприяти активному вивченню. Він слугує як інструмент самоосвіти та може використовуватися в університетських курсах з програмування.

Завдяки цьому практикуму, автори сподіваються, що студенти зможуть не лише освоїти навички програмування та роботи зі структурами даних, а й розвинути критичне мислення та здатність ефективно вирішувати завдання в області інформаційних технологій.

# СИЛАБУС

## **1. Анотація дисципліни (призначення навчальної дисципліни).**

Навчальна дисципліна «Основи програмування, структури даних та алгоритми» є інструментальною основою для виконання аналітичної частини подальших спецкурсів, а також курсових і дипломних робіт. Вона забезпечує наступні дисципліни: «Об'єктно-орієнтоване програмування», «Операційні системи», і «Засоби програмування комп'ютерної графіки».

## **2. Мета навчальної дисципліни.**

Метою навчальної дисципліни є оволодіння основами сучасних мов та технологій програмування, набуття умінь та досвіду зі створення прикладних програм, програмних комплексів з врахуванням особливостей сучасних мов та технологій програмування при вирішенні задач у науковій, інженерно-технічній та економічній сферах діяльності та сприяння розвитку логічного та аналітичного мислення студентів.

## **3. Пререквізити.**

Основи інженерії програмного забезпечення

## **4. Результати навчання.**

Відповідно до освітньо-професійної програми підготовки магістрів галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення (освітня програма «Програмне забезпечення систем») вивчення дисципліни сприяє формуванню компетентностей та програмних результатів навчання:

### ***Загальні та фахові компетентності:***

- ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК03. Здатність проводити дослідження на відповідному рівні.
- ЗК05. Здатність генерувати нові ідеї (креативність).
- СК01. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення.
- СК02. Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення, готувати заявки на винаходи й промислові зразки, організувати роботи зі здійснення авторського нагляду при розробці, налагодженні, випробуваннях і здачі замовнику програмного продукту.
- СК03. Здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів; володіння знаннями щодо виконання циклу робіт по вдосконаленню, еволюції, модернізації, уніфікації та міграції програмних продуктів, їх компонентів та окремих елементів.
- СК11. Здатність ефективно проводити стратегічний аналіз, здійснювати вибір когнітивних моделей й методів візуалізації інформації та протидіяти інформаційним впливам, обробляти аналітичні матеріали для прийняття обґрунтованих рішень, розробляти інвестиційні проекти та стратегії розвитку підприємств, управляти економічними ризиками та оцінювати й забезпечувати якість виконуваних робіт.

### ***Програмні результати навчання:***

- РН05. Розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення.

- РН09. Обґрунтовано обирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення.
- РН16. Планувати, організовувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення.

## 5. Опис навчальної дисципліни

### 5.1. Загальна інформація

| Форма навчання | Рік підготовки | Семестр | Кількість |       | Кількість годин |           |             |             |                   |                        | Вид підсумкового контролю |
|----------------|----------------|---------|-----------|-------|-----------------|-----------|-------------|-------------|-------------------|------------------------|---------------------------|
|                |                |         | кредитів  | годин | лекції          | практичні | семінарські | лабораторні | самостійна робота | індивідуальні завдання |                           |
| Денна          | 1              | 1       | 6         | 180   | 30              | 30        | 0           | 30          | 90                | 0                      | іспит                     |
| Заочна         |                |         |           |       |                 |           |             |             |                   |                        |                           |

### 5.2. Дидактична карта навчальної дисципліни

| Назви змістових модулів і тем            | Кількість годин  |              |   |     |     |      |        |              |    |     |     |      |  |  |
|--|--|--------------|---|-----|-----|------|--------|--------------|----|-----|-----|------|--|--|
|  | денна форма  |              |   |     |     |      |        | заочна форма |    |     |     |      |  |  |
|  | усього   | у тому числі |   |     |     |      | усього | у тому числі |    |     |     |      |  |  |
|  |  | л            | п | лаб | інд | с.р. |        | л            | п  | лаб | інд | с.р. |  |  |
| 1  | 2  | 3            | 4 | 5   | 6   | 7    | 8      | 9            | 10 | 11  | 12  | 13   |  |  |
| <b>Теми лекційних занять</b>             | <b>Змістовий модуль 1 – Основи програмування мовою C++ та алгоритмізація</b> |              |   |     |     |      |        |              |    |     |     |      |  |  |
| Тема 1 Загальні поняття в програмуванні. | 14   | 2            | 2 | 4   |     | 6    |        |              |    |     |     |      |  |  |
| Тема 2. Елементи мови C++                | 14   | 2            | 2 | 4   |     | 6    |        |              |    |     |     |      |  |  |
| Тема 3. Умовні оператори мови C++        | 16   | 2            | 2 | 4   |     | 8    |        |              |    |     |     |      |  |  |
| Тема 4. Циклічні оператори               | 24   | 4            | 4 | 8   |     | 8    |        |              |    |     |     |      |  |  |
| Тема 5. Одновимірні масиви               | 24   | 4            | 4 | 8   |     | 8    |        |              |    |     |     |      |  |  |
| Тема 6. Багатовимірні масиви             | 16   | 2            | 2 | 4   |     | 8    |        |              |    |     |     |      |  |  |

|  |   |    |    |    |   |     |  |  |  |  |  |  |
|--|---|----|----|----|---|-----|--|--|--|--|--|--|
| Тема 7. Вказівники (показчики) та динамічна пам'ять.     | 16  | 2  | 2  | 4  |   | 8   |  |  |  |  |  |  |
| Тема 8. Рядки в C++ та символічні масиви                 | 16  | 2  | 2  | 4  |   | 8   |  |  |  |  |  |  |
| <i>Разом за ЗМ1</i>                                      | 140   | 20 | 20 | 40 |   | 60  |  |  |  |  |  |  |
| <b>Теми лекційних занять</b>                             | <b>Змістовий модуль 2 – Структури даних мовою C++</b> |    |    |    |   |     |  |  |  |  |  |  |
| Тема 9. Функції в C++                                    | 31  | 4  | 4  | 8  |   | 15  |  |  |  |  |  |  |
| Тема 10. Робота з текстовими файлами в C++.              | 23  | 2  | 2  | 4  |   | 15  |  |  |  |  |  |  |
| Тема 11 Структури в C++                                  | 23  | 2  | 2  | 4  |   | 15  |  |  |  |  |  |  |
| Тема 12. Бібліотека STL (стандартна бібліотека шаблонів) | 23  | 2  | 2  | 4  |   | 15  |  |  |  |  |  |  |
| <i>Разом за ЗМ 2</i>                                     | 100   | 10 | 10 | 20 |   | 60  |  |  |  |  |  |  |
| <b>Усього годин</b>                                      | 240   | 30 | 30 | 60 | 0 | 120 |  |  |  |  |  |  |

### 5.3. Зміст завдань для самостійної роботи

| №  | Назва теми                               |
|----|--|
| 1  | Побудова блок-схем                       |
| 2  | Лінійні алгоритми. Елементарні оператори |
| 3  | Умовні оператори та оператор вибору      |
| 4  | Циклічні оператори                       |
| 5  | Одновимірні масиви                       |
| 6  | Багатовимірні масиви                     |
| 7  | Вказівники. Робота з динамічною пам'яттю |
| 8  | Рядки                                    |
| 9  | Функції                                  |
| 10 | Шаблони функцій                          |
| 11 | Структури                                |
| 12 | Робота з файлами                         |
| 13 | Робота з STL                             |

### 6. Система контролю та оцінювання

#### Методи навчання

- словесні методи (лекція, дискусія, бесіда, консультація тощо).

- практичні методи (практичні або лабораторні роботи).
- наочні методи (презентації результатів виконаних завдань, ілюстрації, відеоматеріали, тощо).
- робота з інформаційними ресурсами: з навчально-методичною, науковою, нормативною літературою та Інтернет-ресурсами.
- самостійна робота над індивідуальним завданням або за програмою навчальної дисципліни.

### **Форми та засоби оцінювання**

- оцінювання завдань лабораторних робіт.
- контрольні роботи.
- стандартизовані тести.
- презентації результатів виконаних завдань та досліджень.

### **Види та форми контролю**

*Форми поточного контролю:*

- захист і презентації результатів виконаних лабораторних / практичних завдань.
- письмова (тестування).

*Форма підсумкового контролю – залік*

### **Критерії оцінювання результатів навчання з навчальної дисципліни**

***Політика щодо дедлайнів та перескладання:*** Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку (-50%). Перескладання модулів відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

***Політика щодо академічної доброчесності:*** Усі письмові роботи перевіряються на наявність плагіату і допускаються до захисту із коректними текстовими запозиченнями не більше 20%. Списування під час контрольних робіт, заліків або іспитів заборонені (в т.ч. із використанням мобільних девайсів).

***Політика щодо відвідування:*** Відвідування занять є обов'язковим компонентом оцінювання. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання може відбуватись в online формі за погодженням із керівником курсу.

#### ***Критеріями оцінювання є:***

- при усних відповідях: повнота розкриття питання; логіка викладання матеріалу; використання основної та додаткової літератури; аналітичні міркування, уміння робити порівняння, висновки; уміння аналізувати теоретичні проблеми з урахуванням світової і вітчизняної практики;
- при виконанні письмових завдань: повнота розкриття питання, аргументованість і логіка викладення матеріалу, використання літературних джерел, законодавчих актів, прикладів та фактичного матеріалу тощо; цілісність, системність, логічність, уміння формулювати висновки; акуратність оформлення письмової роботи.

Проведення підсумкового контролю здійснюється у формі передбаченою навчальним планом в обсязі навчального матеріалу, визначеного навчальною програмою дисципліни і в терміни, передбачені графіком навчального процесу.

Загальна підсумкова оцінка з дисципліни (максимум 100 балів) визначається як сума балів поточного і модульного контролю та результатів заліку/іспиту (як можливість



отримання додаткових балів, якщо набрані протягом семестру бали не влаштовують студентів). У випадку отримання менше 50 балів за результатами загального підсумкового контролю, студент обов'язково здійснює перескладання для ліквідації академічної заборгованості.

Загальні вимоги для одержання підсумкової оцінки:

– «відмінно» / «зараховано» – студент вільно володіє матеріалом дисципліни; може самостійно і грамотно провести всі необхідні розробки і викладки з усіх передбачених програмою питань, може розв'язувати нестандартні задачі, відповідь охоплює не менше 90% матеріалу питань в білеті.

– «добре» / «зараховано» – студент вільно орієнтується у матеріалі дисципліни; може грамотно відтворити лекційний матеріал; може розв'язувати всі стандартні задачі з матеріалу дисципліни; відповідь охоплює не менше 75% матеріалу питань в білеті.

– «задовільно» / «зараховано» – студент знає основні поняття і твердження, але не всі може відповідно обґрунтувати; може розв'язати прості стандартні задачі; відповідь охоплює не менше 60% матеріалу питань в білеті.

– «незадовільно» / «не зараховано» – вимоги позитивних оцінок не виконуються, відповідь містить менше 60% потрібного матеріалу питань білету.

### Шкала оцінювання знань студентів: національна та ECTS

| Сума балів за всі види навчальної діяльності | Оцінка ECTS | Оцінка за національною шкалою                              |   |
|--|-------------|--|---|
|  |             | для іспиту, курсового проєкту (роботи), практики           | для заліку  |
| 90-100                                       | <b>A</b>    | відмінно   | зараховано  |
| 80-89  | <b>B</b>    | добре  |   |
| 70-79  | <b>C</b>    |  |   |
| 60-69  | <b>D</b>    | задовільно   |   |
| 50-59  | <b>E</b>    |  |   |
| 35-49  | <b>FX</b>   | незадовільно з можливістю повторного складання             | не зараховано з можливістю повторного складання             |
| 0-34   | <b>F</b>    | незадовільно з обов'язковим повторним вивченням дисципліни | не зараховано з обов'язковим повторним вивченням дисципліни |

### Розподіл балів, які отримують студенти

| Поточне оцінювання<br>(аудиторна та самостійна робота) |    |    |                     |    |    | Кількість балів (залік) | Сумарна к-ть балів |
|--|----|----|---------------------|----|----|-------------------------|--------------------|
| Змістовий модуль №1                                    |    |    | Змістовий модуль №2 |    |    |                         |                    |
| T1   | T2 | T3 | T4                  | T5 | T6 | 30                      | 100                |
| 10   | 10 | 10 | 10                  | 15 | 15 |                         |                    |

## 7. Рекомендована література

- 1) Programming : principles and practice using C++ / Bjarne Stroustrup. – Second edition. – Addison-Wesley Professionalю – 2014. – 1312 p.
- 2) Skudaev S. Learn Critical Computer Programming Concepts: Three books in one with code examples and hands-on projects – Kindle Edition, 2021. – 722 p.
- 3) Пекарський Б.Г. Основи програмування: Навчальний посібник. Кондор, 2018. - 364 с.
- 4) Герберт Шилдт. С++: базовий курс. Вільямс, 2020. – 624с.
- 5) Джордж Хайнеман, Гері Полліс, Стенлі Селков. Алгоритми. Довідник з прикладами на С, С++, Java та Python, Вільямс, 2020. – 432 с.
- 6) Васильєв О.М. Програмування С++ в прикладах і задачах. Навчальний посібник, Ліра-К. – 2019. – 382 с.
- 7) Nyisztor K. Secure C and C++ Coding: Master Secure C/C++ Programming: Uncover Flaws, Learn Defensive Programming Techniques & Design Robust, Secure Software Kindle Edition, 2024 – 91 p.
- 8) Роберт Мартін. Чистий код. Фабула, 2019. – 368 с.

### Інформаційні (електронні) ресурси

- 1) International Standard ISO/IEC 14882:2014(E) – Programming Language C++ // <https://isocpp.org/std/the-standard>
- 2) C/C++ language and standard libraries reference // <https://msdn.microsoft.com/en-us/library/hh875057.aspx>
- 3) The C++ Programming Language (Bjarne Stroustrup's homepage) // <http://www2.research.att.com/~bs/C++.html>
- 4) ISO/IEC 14882:2003 Programming languages - C++ (International Standard) // <http://cs.nyu.edu/courses/summer12/CSCI-GA.2110-001/downloads/C++%20Standard%202003.pdf>
- 5) The C++ Resources Network // <http://www.cplusplus.com/>
- 6) The C++ Tutorial // <http://www.learncpp.com/>
- 7) C++ Tutorial - W3Schools // <https://www.w3schools.com/cpp/>
- 8) C++ Tutorial - Tutorialspoint // <https://www.tutorialspoint.com/cplusplus/index.htm>
- 9) C++ - Вікіпідручник // <http://uk.wikibooks.org/wiki/C++>
- 10) C++ - Вікіпедія // <http://uk.wikipedia.org/wiki/C++>

# МОДУЛЬ 1. ОСНОВИ ПРОГРАМУВАННЯ МОВОЮ C++ ТА АЛГОРИТМІЗАЦІЯ

## Лабораторна робота №1

### Тема : Побудова блок-схем

(3 завдання, оцінюється в 2 бали)

#### Теоретичні відомості

Алгоритм (algorithm) – це послідовність інструкцій чи кроків, спрямованих на виконання певного завдання або вирішення конкретної проблеми. Алгоритм визначає певні правила обробки інформації, тобто він визначає певну послідовність операцій опрацювання даних для розв'язку задачі.

Задавати алгоритми можна у різні способи:

- Написання у вигляді тексту
- Псевдокод
- Математичні формули
- Діаграми потоку
- Блок-схеми

Один з найпоширеніших є графічний спосіб представлення алгоритмів у вигляді схем алгоритмів (блок-схем), представлених на рис. 1.





| Назва      | Позначення  | Призначення                              |
|------------|---|--|
| Термінатор |  | Початок або кінець алгоритму             |
| Процес     |  | Виконання однієї або кількох команд      |
| Рішення    |  | Перевірка умови                          |
| Дані       |  | Уведення даних або виведення результатів |

Рис. 1 – Базові блоки блок-схеми

Розрізняють три базові алгоритмічні структури: лінійна (послідовність), розгалужена та циклічна (повторювання) (рис. 2). Алгоритмічні структури будь-якої складності утворюються поєднанням послідовності цих базових алгоритмів.

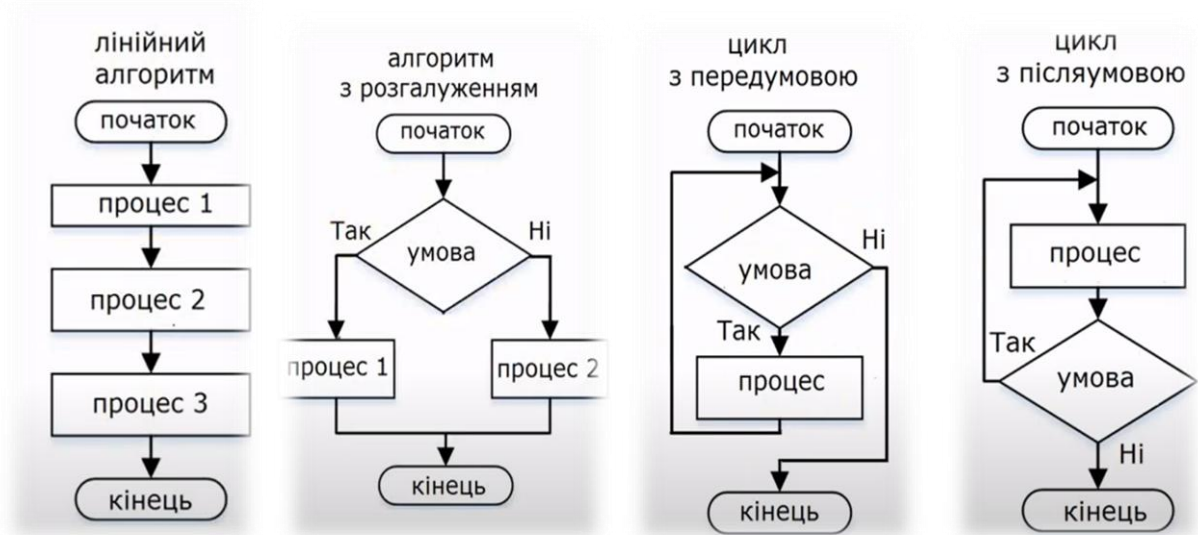


Рис. 2 – Базові алгоритмічні структури

**Завдання 1(0.5 бала)**

*Блок-схема лінійного алгоритму*

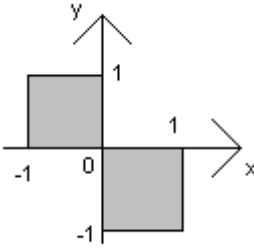
| Варіант | Завдання  |
|---------|---|
| 1       | Обчислити об'єм циліндра з радіусом основи $r$ і висотою $h$ .  |
| 2       | Визначити відстань на площині між двома точками з заданими координатами $M_1(x_1, y_1)$ і $M_2(x_2, y_2)$ |
| 3       | Дано довжину ребра куба. Знайти об'єм куба і площу його бічної поверхні.                                  |
| 4       | Три опори $R_1, R_2, R_3$ з'єднані паралельно. Знайти опір з'єднання.                                     |
| 5       | Визначити час падіння каменю на поверхню землі з висоти $h$ .   |
| 6       | Відома довжина кола. Знайти площу круга, обмеженого цим колом.  |
| 7       | Трикутник заданий координатами своїх вершин. Знайти периметр трикутника і його площу.                     |
| 8       | Обчислити висоти трикутника за відомим значенням довжин його сторін.                                      |

|    |   |
|----|---|
| 9  | Обчислити довжини сторін трикутника за відомим координатами вершин.   |
| 10 | Знайти значення виразу $z = x - 5y + 10$ , де $y = 3x + 4$  |
| 11 | Визначити відстань, пройдену фізичним тілом за час $t$ , якщо тіло рухається з постійним прискоренням $a$ і має в початковий момент часу швидкість $V_0$ .  |
| 12 | Обчислити площу трикутника, якщо задані довжини його сторін.  |
| 13 | Визначити координати вершини параболи $y = ax^2 + bx + c$ .   |
| 14 | За даними сторонам прямокутника обчислити його периметр, площу і довжину діагоналі.   |
| 15 | Студент почав вирішувати задачі даного уроку програмування, коли електронні годинники показували $h_1$ годин і $min_1$ хвилин, а закінчив, коли було $h_2$ годин і $min_2$ хвилин. Визначити, скільки часу студент вирішував ці задачі (вважати, що задачі розв'язувалися не довше доби).                             |
| 16 | Катерина купила собі у магазині 50 грам цукерок по $m$ гривень за 1 кг та $t$ булок хліба по $r$ гривень за булку. Мама дала Катерині із собою $w$ гривень. Скільки грошей витратила Катерина на усі покупки та яка решта грошей у неї залишилася?  |
| 17 | Олег пішов на рибалку разом із татом. Олег спіймав $n$ кг карасів, $m$ штук щук. Після цього вони поїхали на ринок та продали усі карасі по $f$ гривень за 1 кг, а також по $d$ гривень за кожну щуку. Скільки тато і син виручили за усіх карасів, за всі щуки окремо? Скільки всього грошей заробили тато із сином? |
| 18 | Задана довжина ребра куба. Знайти його об'єм та площу всієї поверхні.   |

### Завдання 2 (0.5 бала)

Блок-схема алгоритму з розгалуженням.

| Варіант | Завдання   |
|---------|--|
| 1       | $f(x) = \begin{cases} x - 1, & x \leq 0 \\ x^2, & 0 < x < 2 \\ 2x, & x \geq 2 \end{cases}$ |
| 2       | Обчислити функцію $Ln(X)$ , яка перевіряє допустимість значення аргументу $X$              |

|    |  |
|----|--|
| 3  | $f(x) = \begin{cases} \sqrt{1-x}, & x \leq 0 \\ 0, & 0 < x \leq 2 \\ x-2, & x > 2 \end{cases}$   |
| 4  | $f(x) = \begin{cases} -2(x+1), & x \leq -1 \\ (x+1)^3, & -1 < x < 0 \\ x, & x \geq 0 \end{cases}$  |
| 5  | $f(x) = \begin{cases} -x, & x \leq 0 \\ x^2, & 0 < x \leq 2 \\ x+1, & x > 2 \end{cases}$   |
| 6  | $f(x) = \begin{cases} x^2 + 1, & x \leq 1 \\ 2x, & 1 < x \leq 3 \\ x + 2, & x > 3 \end{cases}$   |
| 7  | <p>Дана заштрихована область і точка з координатами <math>x, y</math>. Визначити, чи попаде задана точка в заштриховану область.</p>  |
| 8  | $f(x) = \begin{cases} x-3, & x < 0 \\ x+1, & 0 \leq x \leq 4 \\ 3+x, & x > 4 \end{cases}$  |
| 9  | $f(x) = \begin{cases} 2x^2, & x \leq 0 \\ x, & 0 < x \leq 1 \\ 2+x, & x > 1 \end{cases}$   |
| 10 | <p>Дана заштрихована область і точка з координатами <math>x, y</math>. Визначити, чи попаде задана точка в заштриховану область.</p>   |

|    |  |
|----|--|
|    |  |
| 11 | $f(x) = \begin{cases} \cos x, & x \leq \pi/2 \\ 0, & \pi/2 < x < \pi \\ 2, & x \geq \pi \end{cases}$   |
| 12 | З'ясувати, чи існує трикутник з заданими довжинами сторін $x, y, z$ .  |
| 13 | $f(x) = \begin{cases} x+1, & x < 0 \\ x^2-1, & 0 \leq x < 1 \\ -x, & x \geq 1 \end{cases}$   |
| 14 | <p>Дана заштрихована область і точка з координатами <math>x, y</math>. Визначити, чи попаде задана точка в заштриховану область.</p>   |
| 15 | $f(x) = \begin{cases} -x, & x < 0 \\ x^2+1, & 0 \leq x < 2 \\ x+1, & x \geq 2 \end{cases}$   |
| 16 | Визначити, чи введений рік є високосним. Рік вважати високосним, якщо він кратний 4.   |
| 17 | Визначити, до якої чверті системи координат належить точка з введеними координатами.   |
| 18 | Обчислити площу круга, прямокутника і трикутника. Потрібно ввести з клавіатури який тип фігури обирає користувач і відповідна ввести радіус кола, довжини сторін прямокутника чи трикутника. |

### Завдання 3 (1 бал)

#### Блок-схема циклічного алгоритму

| Варіант | Завдання  |
|---------|---|
| 1       | Знайти добуток всіх двозначних чисел, сума цифр яких кратна 3.  |
| 2       | Визначити, скільки чисел є кратним 3 і скільки одночасно 3 і 5 на проміжку від $x$ до $y$ , де $x=4$ , $y=183$ .  |
| 3       | Знайти всі двозначні числа, сума цифр яких зменшується при множенні числа на 2.   |
| 4       | Знайти всі двозначні числа, сума цифр яких дорівнює даному числу.   |
| 5       | Визначити кількість дільників та самі дільники заданого числа   |
| 6       | Знайти всі тризначні числа, які можна представити різницею між квадратом числа, утвореного першими двома цифрами і квадратом третьої.                   |
| 7       | Знайти всі двозначні числа, сума квадратів цифр яких поділяється на 7.  |
| 8       | Знайти НСД двох натуральних чисел.  |
| 9       | Знайти двозначне число, що куб суми його цифр дорівнює квадрату самого числа.   |
| 10      | Знайти двозначне число, рівне потроєному добутку його цифр.   |
| 11      | Знайти двозначні числа, подвоєна сума цифр яких дорівнює їхньому добутку?   |
| 12      | Перший член геометричної прогресії 6, а її знаменник - 0.5. Створіть програму для обчислення значень членів прогресії, більших 0.6, і визначте її суму. |
| 13      | Визначити, чи є задане число досконалим, тобто рівним сумі усіх своїх (додатних) дільників, крім самого цього числа.                                    |
| 14      | Знайти всі тризначні числа, середня цифра яких дорівнює сумі першої і другої.   |
| 15      | Знайти суму парних значень функції $y=x^2$ для $x$ від -5 до 5 з кроком 0,5   |
| 16      | Перевірити, чи введене число просте.  |
| 17      | Протабулювати функцію $y = 9x^3 + 8$ на проміжку від А до В з кроком k.   |
| 18      | За даним натуральним $n$ обчисліть суму $1^2+2^2+3^2+\dots+n^2$   |



## Лабораторна робота № 2

### Тема: Лінійні алгоритми. Елементарні оператори

(2 завдання, оцінюється в 2 бали)

#### Теоретичні відомості

*Лінійні алгоритми* – це алгоритми, які виконують послідовні операції над даними. У мові програмування C++, ви можете використовувати різні оператори вводу та виводу для взаємодії з користувачем.

**Оператор виводу** (<<) використовується для виводу даних на екран. Наприклад:

```
#include <iostream>

int main() {
    int number = 10;
    std::cout << "Значення числа: " << number << std::endl;
    return 0;
}
```

У цьому прикладі std::cout є об'єктом виводу, а << – це оператор виводу. Він використовується для виводу тексту та значення змінної на консоль.

**Оператор вводу** (>>) використовується для отримання даних від користувача.

Наприклад:

```
#include <iostream>

int main() {
    int number;
    std::cout << "Введіть число: ";
    std::cin >> number;
    std::cout << "Ви ввели: " << number << std::endl;
    return 0;
}
```

У цьому прикладі std::cin є об'єктом вводу, а >> – це оператор вводу. Він використовується для отримання значення від користувача.

Лінійні алгоритми можуть використовувати ці оператори вводу/виводу для взаємодії з користувачем та виконання послідовних операцій над даними.

В мові програмування C++, бібліотека cmath надає набір функцій для виконання математичних операцій, зокрема для роботи з дійсними числами.

Основні можливості цієї бібліотеки включають стандартні математичні функції, тригонометричні функції, експоненціальні та логарифмічні функції та інші. Нижче представлені деякі приклади функцій, які можна знайти у бібліотеці `cmath` в мові C++:

```
#include <cmath>
double result = std::abs(-5.5); // result = 5.5
```

### Функції бібліотеки `cmath`

| <i>Приклад звернення до функції</i>            | <i>Обчислення функції</i>              |
|--|--|
| <code>fabs(вираз)</code>                       | Модуль числа                           |
| <code>sqrt(вираз)</code>                       | Квадратний корінь                      |
| <code>pow(вираз, ступінь)</code>               | Піднесення до ступеню                  |
| <code>exp(вираз для степеню числа e)</code>    | Експонента                             |
| <code>log(вираз)</code>                        | Натуральний логарифм                   |
| <code>log10(вираз)</code>                      | Логарифм за основою 10                 |
| <code>sin(вираз)</code>                        | Синус                                  |
| <code>cos(вираз)</code>                        | Косинус                                |
| <code>tan(вираз)</code>                        | Тангенс                                |
| <code>asin(вираз)</code>                       | Арксинус                               |
| <code>acos(вираз)</code>                       | Арккосинус                             |
| <code>atan(вираз)</code>                       | Арктангенс                             |
| <code>sinh(вираз)</code>                       | Гіперболічний синус                    |
| <code>cosh(вираз)</code>                       | Гіперболічний косинус                  |
| <code>tanh(вираз)</code>                       | Гіперболічний тангенс                  |
| <code>ceil(вираз)</code>                       | Округлення до найближчого більшого     |
| <code>floor(вираз)</code>                      | Округлення до найближчого меншого      |
| <code>fmod(вираз ділене, вираз дільник)</code> | Залишок від ділення з плаваючою точкою |

#### ***Завдання 1 (0.5 бала)***

Для даного завдання використовуючи розроблену блок-схему алгоритму завдання 1 лабораторної роботи № 1 написати відповідну програму з виведенням результатів на екран.

#### ***Вимоги до програми:***

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної змінної та результати обчислень

### Завдання 2(1.5 бали)

Розробити алгоритм та записати відповідну програму обчислення значення математичного виразу.

#### Вимоги до програми:

- вхідні дані ввести за допомогою стандартних процедур введення;
- на друк вивести значення вхідної змінної та результати обчислень.

|    |  |     |   |
|----|--|-----|---|
| 1. | $\frac{\sqrt{e^x - \cos^4(x^2 a^5)} + \operatorname{ctg}^4(a - x^5)}{\sqrt{ a + xc^4 }}$     | 10. | $\frac{\cos x^3 - a\sqrt{6} - \cos(3ab)}{\sin^2(a \cdot \sin x + \ln y)}$                               |
| 2. | $\frac{\sqrt{(2+y)^2 + \sqrt{\sin(y+5)}}}{\ln(x+1) - y^3}$                                   | 11. | $\frac{\cos^3 y+x  - (x+y)}{\operatorname{tg}^4(x+a)x^5}$   |
| 3. | $\frac{\operatorname{tg}(x^4 - 6) - \cos^{3x}(z + x^3 y)}{\cos^2 x^3 c^2}$                   | 12. | $\frac{a}{x-a} + \frac{b^x + \cos^3 x}{\log^3 a + 4.5}$   |
| 4. | $\frac{a^5 + \cos(a + x^3) - \sin^4(y - c)}{\sin^3(x+y) +  x-y }$                            | 13. | $\frac{(n+1)(m-4) + \sin^2 nm - n^4 + m^3}{\ln(m+2)^2}$   |
| 5. | $\frac{\cos^7 bx^5 - (\sin a^2 + \cos(x^3 + z^5 - a^2))}{\sin a^2 + \cos(x^7 - a^2)}$        | 14. | $\operatorname{arctg} \frac{x+1}{y-2} + \lg k+x $   |
| 6. | $\sin(xy) + \frac{(x+y)^3}{x^2 + \cos y^2} (x - y)$  | 15. | $\frac{\sin^2 a^3 - \arcsin \frac{1}{b}}{\ln a+b  - 1}, z = \sqrt{\left  \frac{a+b}{ab} \right } + \pi$ |
| 7. | $\frac{\sqrt{ x  + \cos^3 x + z^4}}{\ln x - \sin(bx - a)}$                                   | 16. | $\operatorname{arccos} \left( \frac{x^2}{0,13} \right)^{-1} + \ln y^{-1} $                              |
| 8. | $\frac{\cos(x^3 + 6) - \sin(y - a)}{\ln x^4 - 2\sin^5 x}$                                    | 17. | $\frac{\ln m-y  + \cos^3 my}{\sqrt{ m+y ^3 + 17,14}}$   |
| 9. | $\frac{\sqrt{a^{5-x} - \cos^4(x^2) + \operatorname{tg}^4(a - x^5)}}{7x - \sqrt{ a + xc^4 }}$ | 18. | $\frac{\sqrt{3 + \cos(a + x^3)a} + a^5 + \sin^4(y - c)}{\sin^3(y) +  x\sqrt{a - y} }$                   |

## Лабораторна робота № 3

### Тема: Умовні оператори та оператор вибору

(2 завдання, (оцінюється в 2 бали)

#### Теоретичні відомості

Умовні оператори та оператор вибору включають різні конструкції, які дозволяють вам виконувати різні дії залежно від умов. В C++ – це *if (if-else)* та *switch*.

#### **Оператор *if-else*:**

Оператор *if* використовується для виконання блоку коду, якщо вказана умова є істинною.

Оператор *else* використовується для визначення блоку коду, який виконується, якщо жодна з попередніх умов не є істинною.

```
if (умова) {  
    // код, який виконається, якщо умова є істинною  
} else {  
    // код, який виконається, якщо умова не є істинною  
}
```

#### **Оператор *switch*:**

Оператор *switch* дозволяє вам визначити блок коду для виконання в залежності від значення виразу.

```
switch (вираз) {  
    case значення1:  
        // код, який виконається, якщо вираз дорівнює значенню1  
        break;  
    case значення2:  
        // код, який виконається, якщо вираз дорівнює значенню2  
        break;  
    // і так далі...  
    default:  
        // код, який виконається,  
        //якщо жоден з попередніх case не виконається  
}
```

Ці оператори важливі для написання гнучких та структурованих програм, які можуть змінювати свою поведінку в залежності від умов та введених даних.

**Завдання 1 (1 бал) (Умовний оператор if-else)**

Дійсні величини  $x, y, z, a, b, c, d$  вводяться з клавіатури. Розробити програму, яка впорядковує величини  $a, b, c, d$  так, щоб  $a \leq b$  і  $c \leq d$ , та обчислює значення функції та виводить на екран результат:

|    |  |
|----|--|
| 1. | $V = \begin{cases} \max(x, y, z), & \text{якщо } x, y, z \in [c, d], \\ \min(x, y, z), & \text{якщо } x, y, z \notin [c, d], \\ (a + b)/2, & \text{в інших випадках.} \end{cases}$   |
| 2. | $V = \begin{cases} \max( x ,  y ,  z ), & \text{якщо } x, y, z \in [c, d], \\ \sqrt{a^2 + b^2}, & \text{якщо } x, y, z \notin [c, d], \\ \min(x^2, y^2, z^2), & \text{в інших випадках.} \end{cases}$  |
| 3. | $V = \begin{cases} (\max(x, y, z) + \min(x, y, z))/a, & \text{якщо } a, b \in [c, d], \\ (\max(x, y, z) + \min(x, y, z))/b, & \text{якщо } a, b \notin [c, d], \\ ( a  +  b )/2, & \text{в інших випадках.} \end{cases}$   |
| 4. | $V = \begin{cases} x^2 + y^2 + z^2, & \text{якщо } c \leq x \leq y \leq z \leq d, \\ \sqrt{x^2 + y^2 + z^2}, & \text{якщо } x \leq y \leq z \leq c, \\  x  +  y  +  z , & \text{якщо } d \leq x \leq y \leq z, \\ (a + b)/2, & \text{в інших випадках.} \end{cases}$ |
| 5. | $V = \begin{cases} ( x  +  y )/ z , & \text{якщо } x, y \in [c, d] \text{ і } z \neq 0, \\ \min(x, y) + \sqrt{x^2 + y^2}, & \text{якщо } x, y \in [c, d] \text{ і } z = 0, \\ \max( x ,  y ,  z ), & \text{в інших випадках.} \end{cases}$                           |
| 6. | $V = \begin{cases} a / \max(x, y, z), & \text{якщо } c^2 \leq x^2 + y^2 + z^2 \leq d^2, \\ b / \min(x, y, z), & \text{якщо } x^2 + y^2 + z^2 \leq c^2, \\ (a + b)/2, & \text{в інших випадках.} \end{cases}$   |
| 7. | $V = \begin{cases} ( x  +  y )/ z , & \text{якщо } x, y \in [c, d] \text{ і } z \neq 0, \\ \min(x, y) + \sqrt{x^2 + y^2}, & \text{якщо } x, y \in [c, d] \text{ і } z = 0, \\ \max( x ,  y ,  z ), & \text{в інших випадках.} \end{cases}$                           |

|    |  |
|----|--|
| 8. | $V = \begin{cases} a / \max(x, y, z), & \text{якщо } c^2 \leq x^2 + y^2 + z^2 \leq d^2, \\ b / \min(x, y, z), & \text{якщо } x^2 + y^2 + z^2 \leq c^2, \\ (a + b) / 2, & \text{в інших випадках.} \end{cases}$ |
|----|--|

Дійсні величини  $x, y, z$  та логічні  $p, q$  (true/false) вводяться з клавіатури. Розробити програму, яка обчислює значення функції та виводить на екран результат:

|     |   |
|-----|---|
| 9.  | $V = \begin{cases} x^2 + y^2 + z^2, & \text{якщо } p = \text{true} \text{ і } q = \text{true}, \\ \sqrt{x^2 + y^2 + z^2}, & \text{якщо } p = \text{false} \text{ і } q = \text{false}, \\ \max( x ,  y ,  z ), & \text{якщо } p = \text{true} \text{ і } q = \text{false}, \\ ( x  +  y  +  z ) / 3, & \text{якщо } p = \text{false} \text{ і } q = \text{true}. \end{cases}$ |
| 10. | $V = \begin{cases} p, & \text{якщо } \min( x ,  y ,  z ) > \max(x, y, z), \\ q, & \text{якщо } \min( x ,  y ,  z ) \leq \max(x, y, z). \end{cases}$   |
| 11. | $V = \begin{cases} p, & \text{якщо } \min( x ,  y ,  z ) > x + y + z, \\ q, & \text{якщо } \min( x ,  y ,  z ) \leq x + y + z. \end{cases}$   |
| 12. | $V = \begin{cases} (\max(x, y, z) + \min(x, y, z)), & \text{якщо } \begin{cases} p = \text{true}, \\ q = \text{true}, \end{cases} \\ (\max(x, y, z) - \min(x, y, z)), & \text{якщо } \begin{cases} p = \text{false}, \\ q = \text{false}, \end{cases} \\ x + y + z, & \text{в інших випадках.} \end{cases}$   |
| 13. | $V = \begin{cases} p, & \text{якщо } x^2 + 3y > \max(x, y, z), \\ q, & \text{якщо } x^2 + 3y \leq \max(x, y, z). \end{cases}$   |
| 14. | $V = \begin{cases} x^2 - y^2 * z^2, & \text{якщо } c \leq x \leq y \leq z \leq d, \\  x  +  y  +  z , & \text{якщо } d \leq x \leq y \leq z, \\ 3\sqrt{x^2 + y^2 + z^2}, & \text{в інших випадках.} \end{cases}$  |
| 15. | $V = \begin{cases} p, & \text{якщо } \max( x ,  y ,  z ) > 2x^2, \\ q, & \text{якщо } \min( x ,  y ,  z ) \leq 2x^2. \end{cases}$   |

|     |  |
|-----|--|
| 16. | $V = \begin{cases} (\max(x, y, z) + \max(x, z)), & \text{якщо } \begin{cases} p = true, \\ q = true, \end{cases} \\ (\max(x, y, z) - \min(y, z)), & \text{якщо } \begin{cases} p = false, \\ q = false, \end{cases} \\ 3x + 3y + 3z, & \text{в інших випадках.} \end{cases}$ |
| 17. | $V = \begin{cases} x^2 - y^2 - z^2, & \text{якщо } c \leq x \leq y \leq z \leq d, \\  x  +  y  + 3 z , & \text{якщо } d \leq x \leq y \leq z, \\ \sqrt{x^5 + y^5 + z^5}, & \text{в інших випадках.} \end{cases}$   |
| 18. | $V = \begin{cases} p, & \text{якщо } \min( x ,  y ,  z ) > 2x^2 + 7, \\ q, & \text{якщо } \min( x ,  y ,  z ) \leq 2x^2 + 7. \end{cases}$  |

**Завдання 2 (1 бал) (Оператор вибору switch)**

Задані дійсні величини  $x, y, z$  та ціле  $k$ . Розробити програму, яка обчислює значення функції:

|    |  |
|----|--|
| 1. | $V = \begin{cases} \min(x, \max(y, z)), & \text{якщо } k = 1, \\ \min(y, \max(x, z)), & \text{якщо } k = 2, \\ \min(z, \max(y, x)), & \text{якщо } k = 3, \\ x + y + z, & \text{в інших випадках.} \end{cases}$                          |
| 2. | $V = \begin{cases} x + \max(y, z), & \text{якщо } k = 1, \\ y + \max(x, z), & \text{якщо } k = 2, \\ z + \max(x, y, z), & \text{якщо } k = 3, \\ x^2 + y^2 + z^2, & \text{в інших випадках.} \end{cases}$                                |
| 3. | $V = \begin{cases} \max(x, y, z), & \text{якщо } k = 10, \\ \min(x, y, z)/k, & \text{якщо } k = 20, \\ \min(x, y, z)/k, & \text{якщо } k = 30, \\ (x^2 + y^2 + z^2)/k^2, & \text{в інших випадках.} \end{cases}$                         |
| 4. | $V = \begin{cases} k + \max(x, y, z), & \text{якщо } k = 1, \\ k + \max(x^2, y^2, z^2), & \text{якщо } k = 2, \\ k + \min(x, y, z), & \text{якщо } k \in [30, 32], \\ k^2 + (x^2 + y^2 + z^2)^2, & \text{в інших випадках.} \end{cases}$ |

|     |   |
|-----|---|
| 5.  | $V = \begin{cases} \min(x, \max(y, z)), & \text{якщо } k = 1, \\ \min(x^2, \max(y^2, z^2)), & \text{якщо } k \in [2, 5], \\ \min( x , \max(y, z)), & \text{якщо } k = 8, \\ (x^2 + y^2 + z^2)^2, & \text{в інших випадках.} \end{cases}$  |
| 6.  | $V = \begin{cases} k^2 + \max(x, y, z), & \text{якщо } k = 1, \\ k + \min(\sqrt{x^2}, \sqrt{y^2}, \sqrt{z^2}), & \text{якщо } k \in [10, 12], \\ k + \max(\sqrt{x}, \sqrt{y}, \sqrt{z}), & \text{якщо } k \in [20, 23], \\ k^2 + (x^2 + y^2 + z^2)^2, & \text{в інших випадках.} \end{cases}$ |
| 7.  | $V = \begin{cases} \min(\max(x, y), z), & \text{якщо } k = 1, \\ \min(\max(y, z), x), & \text{якщо } k \in [5, 10], \\ \min(\max(z, x), y), & \text{якщо } k \in [15, 20], \\ x + y + z, & \text{в інших випадках.} \end{cases}$  |
| 8.  | $V = \begin{cases} \max(\min(x, y, z)), & \text{якщо } k \in [1, 5], \\ \min(\max(x, y, z))/k, & \text{якщо } k = 16, \\ \min(\max(\sqrt{x^2}, \sqrt{y^2}, \sqrt{z^2}))/k^2, & \text{якщо } k \in [18, 22], \\ (x^2 + y^2 + z^2)/k^2, & \text{в інших випадках.} \end{cases}$                 |
| 9.  | $V = \begin{cases} \min(x, \max(y, z)), & \text{якщо } k = 1 \\ \min(y, \max(z, x)), & \text{якщо } k = 2 \\ \min(z, \max(y, x)), & \text{якщо } k = 5 \\ x + y + z, & \text{в інших випадках.} \end{cases}$  |
| 10. | $V = \begin{cases} (\max(x, y, z) + \min(x, y, z)), & \text{якщо } k = 1 \\ (\max(x, y, z) - \min(x, y, z)), & \text{якщо } k = 2 \\ x + y + z, & \text{в інших випадках.} \end{cases}$   |
| 11. | $V = \begin{cases} k^2 * x + \max(y, z), & \text{якщо } k = 1, \\ y/k^2 + \max(x, z), & \text{якщо } k = 2, \\ z + k/\min(x, y) + \max(x, y, z), & \text{якщо } k = 3, \\ \sqrt{x^2 + y^2 + z^2}, & \text{в інших випадках.} \end{cases}$   |



|     |   |
|-----|---|
| 12. | $V = \begin{cases} x^2 + y^2 + z^2, & \text{якщо } k = 1, \\ \sqrt{x^2 + y^2 + z^2}, & \text{якщо } k = 2 \\ \max( x ,  y ,  z ), & \text{якщо } k = 3 \text{ і} \\ ( x  +  y  +  z )/3, & \text{в інших випадках.} \end{cases}$  |
| 13. | $V = \begin{cases} (\max(x, y, z) / \min(x, z)), & \text{якщо } k = 4 \\ (\max(x, y, z) - \min(x, y, z)), & \text{якщо } k = 5 \\ x + y + z, & \text{в інших випадках.} \end{cases}$  |
| 14. | $V = \begin{cases} \min(x, \max(y, z)), & \text{якщо } k = 1, \\ \min(x^2, \max(y^2, z^2)), & \text{якщо } k \in [2, 7], \\ \min( x , \max(y, z)), & \text{якщо } k = 8, \\ (x^2 + y - z^2)^2, & \text{в інших випадках.} \end{cases}$  |
| 15. | $V = \begin{cases} \min(x, \max(y, z)), & \text{якщо } k = 1, \\ \min(y, \max(\sqrt{y}, z)), & \text{якщо } k = [2, 5] \\ \min(z^3, \max(y, x)), & \text{якщо } k = 6, \\ x + y + z, & \text{в інших випадках.} \end{cases}$  |
| 16. | $V = \begin{cases} x^2 + y^2 + z^2, & \text{якщо } k = 5, \\ \sqrt{x^2 + y^2 + z^2}, & \text{якщо } k = 8 \\ \max( x ,  y ,  z ), & \text{якщо } k = 10 \text{ і} \\ ( x  +  y  +  z )/3, & \text{в інших випадках.} \end{cases}$   |
| 17. | $V = \begin{cases} k^2 + \max(x, y, z), & \text{якщо } k = 1, \\ k + \min(\sqrt{x^2}, \sqrt{y^2}, \sqrt{z^2}), & \text{якщо } k \in [10, 12], \\ k + \max(\sqrt{x}, \sqrt{y}, \sqrt{z}), & \text{якщо } k \in [20, 23], \\ k^2 + (x^2 + y^2 + z^2)^2, & \text{в інших випадках.} \end{cases}$ |
| 18. | $V = \begin{cases} (\max(x, y, z) + \min(x, y, z)), & \text{якщо } k = 1 \\ (\max(x, y, z) - \min(x, y, z)), & \text{якщо } k \in [2, 7] \\ x + y + z, & \text{в інших випадках.} \end{cases}$  |

## Лабораторна робота № 4

### Тема: Циклічні оператори

(2 завдання, оцінюється в 3 бали)

#### Теоретичні відомості

Циклічні оператори дозволяють виконувати блок коду кілька разів під час виконання програми. Це дозволяє автоматизувати ітеративні процеси та обробку повторюваних задач. Основні циклічні оператори в C++ є `for`, `while` та `do-while`.

#### **Цикл `for`:**

Цикл `for` є дуже потужним та компактним. Він включає три частини: ініціалізацію, умову та ітерацію.

```
for (ініціалізація; умова; ітерація) {  
    // блок коду, який виконується у кожній ітерації  
}
```

Приклад:

```
for (int i = 0; i < 5; ++i) {  
    // блок коду, який виконується п'ять разів  
}
```

#### **Цикл `while`:**

Цикл `while` виконується, доки вказана умова є істинною.

```
while (умова) {  
    // блок коду, який виконується, доки умова істинна  
}
```

Приклад:

```
int i = 0;  
while (i < 5) {  
    // блок коду, який виконується п'ять разів  
    ++i;  
}
```

#### **Цикл `do-while`:**

Цикл `do-while` схожий на цикл `while`, але блок коду виконується принаймні один раз, навіть якщо умова не є істинною.

```
do {
// блок коду, який виконується, доки умова істинна
} while (умова);
```

Приклад:

```
int i = 0;
do {
// блок коду, який виконується принаймні один раз
++i;
} while (i < 5);
```

Ці циклічні оператори дозволяють програмісти створювати більш гнучкі та ефективні алгоритми, які можуть автоматизувати повторювані завдання та обробку даних в ітеративний спосіб.

### **Завдання 1 (1.5 бали) (оператор for)**

*Розробити програму, використовуючи оператор циклу **for**, яка дозволяє:*

| Вар. | Завдання  |
|------|---|
| 1    | Знайти добуток всіх двозначних чисел, сума цифр яких кратна 3.  |
| 2    | Визначити, скільки чисел є кратним 3 і скільки одночасно 3 і 5 на проміжку від $x$ до $y$ , де $x=4$ , $y=183$ .  |
| 3    | Знайти всі двозначні числа, сума цифр яких зменшується при множенні числа на 2.   |
| 4    | Знайти всі двозначні числа, сума цифр яких дорівнює даному числу.   |
| 5    | Визначити кількість дільників та самі дільники заданого числа   |
| 6    | Знайти всі тризначні числа, які можна представити як різницю між квадратом числа, утвореного другої і третьої цифрами і квадратом першої.               |
| 7    | Знайти всі двозначні числа, сума квадратів цифр яких поділяється на 7.  |
| 8    | Знайти НСД двох натуральних чисел.  |
| 9    | Знайти двозначне число, що куб суми його цифр дорівнює квадрату самого числа.   |
| 10   | Знайти двозначне число, рівне потроєному добутку його цифр.   |
| 11   | Знайти двозначні числа, подвоєна сума цифр яких дорівнює їхньому добутку?   |
| 12   | Перший член геометричної прогресії 6, а її знаменник - 0.5. Створіть програму для обчислення значень членів прогресії, більших 0.6, і визначте її суму. |
| 13   | Визначити, чи є задане число досконалим, тобто рівним сумі усіх своїх (додатних) дільників, крім самого цього числа.                                    |

|    |   |
|----|---|
| 14 | Знайти всі тризначні числа, перша цифра яких дорівнює сумі другої і третьої.      |
| 15 | Знайти суму парних значень функції $y=x^2$ для $x$ від $-5$ до $5$ з кроком $0,5$ |
| 16 | Перевірити, чи введене число просте.  |
| 17 | Протабулювати функцію $y = 9x^3 + 8$ на проміжку від $A$ до $B$ з кроком $k$ .    |
| 18 | За даним натуральним $n$ обчисліть суму $1^2+2^2+3^2+\dots+n^2$                   |

### Завдання 2 (1,5 бали) (Оператор while)

Розробити програму, яка обчислює значення суми з заданою точністю  $\varepsilon$  і вказує кількість врахованих доданків. Дійсні величини  $x, a, \varepsilon$  вводяться з клавіатури. Виконати перевірку на коректність введених даних:  $x \neq 0, a \neq 0, \varepsilon > 0$ . Організувати циклічний (повторюваний) ввід даних, поки не відбудеться введення коректних даних. В рамках даного завдання використовувати оператор for() та функції для піднесення до степеня та обчислення факторіалу заборонено.

|    |   |     |   |
|----|---|-----|---|
| 1. | $\sum_{k=1}^{\infty} \frac{(-1)^k \cos^k(a)}{a^{-k} + 3(k-1)!}$ | 10. | $\sum_{k=1}^{\infty} \frac{e^{-k}}{a^{2k} + k!}$          |
| 2. | $\sum_{k=0}^{\infty} \frac{\ln(a+x)^{2k}}{2^k + k!}$            | 11. | $\sum_{k=0}^{\infty} \frac{\cos^k x}{a^{4k} + k!}$        |
| 3. | $\sum_{k=1}^{\infty} \frac{(-1)^k x^{-k}}{a^k (2k)!}$           | 12. | $\sum_{k=1}^{\infty} \frac{\sin^{2k}(ax)}{a^k + k!}$      |
| 4. | $\sum_{k=1}^{\infty} \frac{\sin(a^k + x^k)}{k!}$                | 13. | $\sum_{k=0}^{\infty} \frac{(a+x)^{-k}}{a^{2k} + k!}$      |
| 5. | $\sum_{k=1}^{\infty} \frac{\sin x^k}{a^{2k} + (2k)!}$           | 14. | $\sum_{k=1}^{\infty} \frac{\sin x^k}{a^{2k} (2k)!}$       |
| 6. | $\sum_{k=1}^{\infty} \frac{(-1)^k x^{-k}}{ka + a^k}$            | 15. | $\sum_{k=1}^{\infty} \frac{\cos x^k + \sin a^k}{(2k-1)!}$ |
| 7. | $\sum_{k=1}^{\infty} \frac{\cos^k(a) - x^k}{2^k + k!}$          | 16. | $\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k} a^k}{2k!}$       |
| 8. | $\sum_{k=1}^{\infty} \frac{(-1)^k x^{-4k}}{a^4 + k!}$           | 17. | $\sum_{k=1}^{\infty} \frac{(-1)^k \ln x^{2k}}{a^k + k!}$  |
| 9. | $\sum_{k=1}^{\infty} \frac{\cos^k(a+x)}{a^k k!}$                | 18. | $\sum_{k=1}^{\infty} \frac{(-1)^k x^{-5k}}{a^k + (2k)!}$  |

## Лабораторна робота № 5

### Тема: Одновимірні масиви

(1 завдання, оцінюється в 2 бали)

#### Теоретичні відомості

Одновимірні статичні масиви в C++ є типом даних, що дозволяє зберігати послідовність одного типу елементів під одним ім'ям. Статичні масиви характеризуються тим, що їхні розміри фіксовані під час компіляції і не можуть змінюватися під час виконання програми. Основні відомості про одновимірні статичні масиви включають:

Оголошення статичного масиву виглядає наступним чином:

```
тип_даних ім'я_масиву[розмір];
```

Наприклад:

```
int numbers[5]; // Оголошення масиву з 5 елементів типу int
```

Масив може бути також ініціалізований під час оголошення:

```
int numbers[] = {1, 2, 3, 4, 5};
```

Доступ до елементів масиву можна отримати за допомогою індекса, що починається з нуля.

```
тип_даних значення = ім'я_масиву[індекс];
```

Наприклад:

```
// Отримання третього елемента масиву  
int value = numbers[2];
```

Для перебору всіх елементів масиву можна використовувати цикл for:

```
for (int i = 0; i < розмір; ++i) {  
    // Обробка кожного елемента масиву  
    тип_даних елемент = ім'я_масиву[i];  
}
```

Наприклад:

```
for (int i = 0; i < 5; ++i) {  
    // Обробка кожного елемента масиву numbers  
    int element = numbers[i];  
}
```

Статичні масиви використовуються для зберігання та обробки групи однотипних даних, і їх можна використовувати в різних аспектах програмування, таких як обчислення, сортування, та інші завдання.

### Завдання (2 бали)

|    |  |
|----|--|
| 1. | Задано дійсні величини $a, b$ ( $a < b$ ) і масив чисел $X[n]$ , $n < 200$ . Розробити програму, яка обчислює суму всіх $X[i] < a$ , добуток всіх $X[i] > b$ і знаходить $\max X[i]$ та $\min X[i]$ серед $X[i] \in (a, b)$ , $i = 0, 1, 2, \dots, n-1$ . Якщо чисел $X[i] < a$ , або $X[i] > b$ , або $X[i] \in (a, b)$ немає, то виводить повідомлення про це. |
| 2. | Задано масив дійсних чисел $A[n*m]$ , $n \leq 100$ , $m \leq 10$ . Розробити програму, яка обчислює<br>$Z = \max(\min(A[0], \dots, A[m-1]), \min(A[m], \dots, A[2m-1]), \dots, \min(A[m*(n-1)], \dots, A[n * m-1]))$ .<br>У масиві з $n$ груп по $m$ чисел. Максимальне значення визначається серед мінімальних значень елементів кожної групи.                  |
| 3. | Задано масив дійсних чисел $A[n]$ , $n \leq 200$ . Розробити програму, яка записує в масив $B[n]$ по одному разу всі повторювані числа в масиві $A[n]$ . Виводить повідомлення, якщо таких чисел немає.  |
| 4. | Задано масив цілих чисел $A[2n]$ , $n \leq 200$ . Розробити програму, яка добуток тих чисел із $A[n], \dots, A[2n-1]$ , які менші за середнє арифметичне значення чисел $A[0], \dots, A[n-1]$ . Якщо таких чисел немає, то виводить повідомлення про це.   |
| 5. | Дано два масиви цілих чисел $X[n]$ і $Y[n]$ , $n \leq 300$ . Розробити програму, яка обчислює суму елементів об'єднання цих масивів. Об'єднанням двох масивів є взяті по одному разу всі елементи масиву $X[n]$ і $Y[n]$ . Вивести масив об'єднання та їх суму.  |
| 6. | Дано масив цілих чисел $A[n]$ , $n \leq 300$ . Розробити програму, яка обчислює суму елементів масиву, якщо він упорядкований за зростанням або добуток, якщо не упорядкований. Масив вважається упорядкованим за зростанням якщо $A[i] \leq A[i+1]$ для $i = 0, 1, 2, \dots, n-2$ .   |
| 7. | Задано масив цілих чисел $A[n]$ , $n \leq 400$ . Розробити програму, яка знаходить мінімальне серед тих чисел, які не повторюються. Якщо таких чисел немає, то виводить повідомлення про це.   |
| 8. | Задано масив дійсних чисел $A[n]$ , $n \leq 500$ . Розробити програму, яка обчислює добуток всіх чисел, які повторюються, і суму всіх чисел, які не повторюються. Якщо чисел що повторюються або не повторюються немає, то виводить відповідне повідомлення.   |

|     |  |
|-----|--|
| 9.  | <p>Задано масив дійсних чисел <math>A[n * m]</math>, <math>n \leq 300</math> і <math>m \leq 25</math>. Розробити програму, яка обчислює</p> $V = \min(\max(A[0], \dots, A[m-1]), \max(A[m], \dots, A[2m-1]), \dots, \max(A[m * (n-1)], \dots, A[n * m-1])).$ <p>У масиві <math>n</math> груп по <math>m</math> чисел. Мінімальне значення шукається серед максимальних значень елементів у кожній групі.</p> |
| 10. | <p>Задано масив чисел <math>A[2n]</math>, <math>n \leq 200</math>. Написати програму, яка обчислює значення:</p> $Z = \max(\min(A[0] \dots A[2n-1]), \min(A[1] \dots A[2n-2]), \dots, \min(A[n-1] \dots A[n])).$   |
| 11. | <p>Задано дійсних масив чисел <math>A[n]</math>, <math>n \leq 500</math>. Написати програму, яка обчислює добуток всіх чисел, які знаходяться між першим і останнім від'ємними елементами цього масиву і виводить цей діапазон. Якщо від'ємних чисел немає або є тільки одне, або вони стоять поряд, то виводить відповідне повідомлення.</p>  |
| 12. | <p>Задано масив чисел <math>A[n]</math>, <math>n \leq 500</math>. Написати програму, яка заміняє кожен третій елемент на суму двох попередніх.</p>   |
| 13. | <p>Задано два масиви цілих чисел <math>A[n]</math>, <math>n \leq 500</math> і <math>B[m]</math>, <math>m \leq 200</math>. Обчислити суму об'єднання <math>A \cup B</math>. Об'єднання <math>A \cup B</math> – це множина елементів масиву <math>A</math> і <math>B</math>, взятих по одному разу. Вивести елементи об'єднання та їх суму.</p>  |
| 14. | <p>Задано масив дійсних чисел <math>A[n]</math>, <math>n \leq 400</math>. Написати програму, яка визначає кількість додатних елементів, що містяться між його максимальним та мінімальним елементом.</p>   |
| 15. | <p>Задано масив чисел <math>A[n]</math>, <math>n \leq 300</math>. Написати програму, яка заміняє всі від'ємні елементи на середнє арифметичне елементів масиву</p>   |
| 16. | <p>Задано масив дійсних чисел <math>A[n]</math>, <math>n \leq 500</math>. Написати програму, яка обчислює суму всіх чисел, які повторюються, і середнє арифметичне всіх чисел, які не повторюються. Якщо чисел що повторюються або не повторюються немає, то виводить відповідне повідомлення.</p>   |
| 17. | <p>Задано масив цілих чисел <math>A[n]</math>, <math>n \leq 200</math>. Написати програму, яка виводить на екран всі числа, що повторюються, по одному разу. Якщо чисел, що повторюються, немає, то виводить повідомлення про це.</p>  |
| 18. | <p>Задано масив дійсних чисел <math>A[n]</math>, <math>n \leq 100</math>. Написати програму, яка обчислює добуток елементів цього масиву, якщо масив упорядкований за зростанням або суму, якщо масив не упорядкований. Масив буде упорядкованим за зростанням якщо якщо <math>A[i] \leq A[i+1]</math> для <math>i = 0, 1, 2, \dots, n - 2</math>.</p>   |

## Лабораторна робота № 6

### Тема: Багатовимірні масиви.

(2 завдання, оцінюється в 3 бали)

#### Теоретичні відомості

Багатовимірні статичні масиви дозволяють зберігати дані у вигляді багатовимірних таблиць. Це означає, що ми можемо мати масив, який складається з інших масивів.

Оголошення багатовимірного масиву виглядає:

```
тип_даних ім'я_масиву[розмір1][розмір2]...[розмірN];
```

Наприклад:

```
int matrix[3][4]; // Оголошення матриці 3x4
```

Багатовимірний масив можна також ініціалізувати:

```
// Оголошення та ініціалізація матриці 2x3  
int matrix[2][3] = {{1, 2, 3}, {4, 5, 6}};
```

Доступ до елементів багатовимірного масиву здійснюється за допомогою кількох індексів:

```
тип_даних значення = ім'я_масиву[індекс1][індекс2]...[індексN];
```

Наприклад:

```
// Отримання значення з другого рядка та третього стовпця  
int value = matrix[1][2];
```

Для обходження багатовимірного масиву можна використовувати вкладені цикли for:

```
for (int i = 0; i < rows; ++i) {  
    for (int j = 0; j < columns; ++j) {  
        // Обробка кожного елемента масиву  
        тип_даних елемент = ім'я_масиву[i][j];  
    }  
}
```

Наприклад:

```
for (int i = 0; i < 2; ++i) {  
    for (int j = 0; j < 3; ++j) {  
        // Обробка кожного елемента матриці matrix  
        int element = matrix[i][j];  
    }  
}
```



### Завдання (3 бали)

|     |   |
|-----|---|
| 1.  | Дано двовимірний масив дійсних чисел $A[n][n]$ , $n \leq 20$ . Написати програму, що будує масив $X[n]$ за правилом: $X[i]$ ( $i = 0, 2, \dots, n-1$ ) дорівнює половині суми модулів максимального і мінімального елементів $i$ -го рядка.   |
| 2.  | Задано дві матриці цілих чисел $A[n][n]$ і $B(n, n)$ , $n \leq 15$ . Написати програму, що будує матрицю $X[n][n]$ множенням елементів кожного рядка першої матриці на найбільше із значень елементів відповідного рядка другої матриці.  |
| 3.  | Дано двовимірний масив дійсних чисел $A[n][m]$ . Написати програму, що переформує масив $A[n][m]$ таким чином, щоб його стовбці розташувались в порядку зростання їх поелементних сум.  |
| 4.  | Дано квадратна матриця цілих чисел $A[n][n]$ , $n \leq 20$ . Написати програму, що перестановкою рядків і стовпчиків перетворює цю матрицю так, щоб максимальний елемент матриці (вважається, що він єдиний) розмістився в лівому верхньому куті.                                     |
| 5.  | Задано дійсну матрицю $A[n][n]$ , $n \leq 20$ . Написати програму, що будує масив $B[n]$ за правилом: елементи масива $B$ є середніми арифметичними значеннями елементів рядків матриці $A$ .   |
| 6.  | Дано двовимірний масив дійсних чисел $A[n][m]$ , $n \leq 15$ , $m \leq 15$ . Написати програму, яка міняє місцями рядки, в яких розміщені мінімальний та максимальний елементи масиву. Якщо такі елементи розташовані в одному рядку – виводиться про це повідомлення.                |
| 7.  | Дано матриця цілих чисел $A[n][n]$ , $n \leq 15$ . Написати програму, що обчислює суму елементів, розміщених вище головної діагоналі і перевищують за величиною всі елементи, що знаходяться нижче головної діагоналі. Якщо таких елементів немає, то виводиться повідомлення про це. |
| 8.  | Дано матриця цілих чисел $A[n][n]$ , $n \leq 15$ . Написати програму, що знаходить і друкує номери тих рядків, у яких всі елементи однакові і номери тих стовпчиків, всі елементи в яких парні. Якщо таких рядків або стовпчиків немає, то виводить повідомлення про це.              |
| 9.  | Дано матриця цілих чисел $A[n][n]$ , $n \leq 20$ . Написати програму, яка шукає номери рядків, елементи яких утворюють симетричні послідовності (наприклад 1, 2, 3, 3, 2, 1 або 1, 2, 3, 5, 3, 2, 1). Якщо таких рядків немає, то виводиться повідомлення про це.                     |
| 10. | Дано матриця дійсних чисел $A[n][m]$ . Написати програму, що будує масив $B(n)$ , де $B[k]=1$ , якщо $k$ -й рядок матриці $A$ симетричний (тобто, перший елемент дорівнює останньому, другий – передостанньому і т. д.: 1, 2, 3, 3, 2, 1 або 1, 2, 3, 5, 3, 2, 1), і $B[k] = 0$       |

|     |   |
|-----|---|
|     | – в протилежному випадку.   |
| 11. | Дано дійсна матриця $A[n][n]$ , $n \leq 20$ . Написати програму, яка буде масив $X[n]$ за правилом: $X[i]$ дорівнює сумі модулів тих елементів, які розміщені між першим і останнім включно додатними елементами $i$ -го рядка. Якщо додатних елементів у $i$ -му рядку немає або є тільки один, або вони розміщені поряд, то $X[i] = -1$ .           |
| 12. | Дано дійсна матриця $A[n][n]$ , $n \leq 15$ . Написати програму, яка буде масив $Y[n]$ за правилом: $Y[i]$ дорівнює сумі модулів тих елементів, які розміщені за першим включно від'ємним елементом $i$ -го рядка. Якщо від'ємного елемента у $i$ -му рядку немає, то $Y[i] = -1$ .   |
| 13. | Дано цілочисловий двовимірний масив чисел $A[n][m]$ . Сформувати матрицю $B[2][k]$ , де в першому рядку матриці $B$ записані різні елементи матриці $A$ , а в другому рядку – кількість зустрічань цих елементів в $A$ .  |
| 14. | Дано матриця цілих чисел $A[n][n]$ , $n \leq 20$ . Написати програму для пошуку найбільшого значення серед тих елементів стовпчиків матриці, сума елементів у яких найменша. Таких стовбчиків з найменшою сумою може бути декілька.   |
| 15. | Дано двовимірний масив чисел $A[n][n]$ , $n \leq 15$ . Написати програму, що буде масив $Y[n]$ за правилом: $Y[i] = true$ , якщо в $i$ -му рядку масиву $A$ кількість від'ємних елементів більша від кількості додатних, інакше $Y[i] = false$ .  |
| 16. | Задано дві матриці цілих чисел $A[n][n]$ і $B[n][n]$ , $n \leq 20$ . Написати програму, що буде і друкує масив $X[n]$ за правилом: $X[n] = 1$ , якщо кількість від'ємних елементів $i$ -го рядка першої матриці дорівнює кількості від'ємних елементів $i$ -го рядка другої матриці та $X[i] = 0$ в інших випадках.                                   |
| 17. | Дано дійсна двовимірний масив чисел $A[n][m]$ , $n \leq 15$ , $m \leq 15$ . Впорядкувати за спаданням стовбець, номер якого введеного з клавіатури. В модифікованому масиві у рядках, що починаються з елемента, кратного 3, замінити на «0» всі елементи, окрім першого елемента в цьому рядку і елемента, який належить впорядкованому стовпчику.   |
| 18. | Дано матриця дійсних чисел $A[n][n]$ , $n \leq 20$ . Написати програму, яка буде масив $X[n]$ за правилом: $X[i]$ дорівнює максимуму з тих елементів, які розміщені між першим і останнім включно додатними елементами $i$ -го стовбця. Якщо додатних елементів у $i$ -му стовбці немає або є тільки один, або вони розміщені поряд, то $X[i] = -1$ . |

## Лабораторна робота № 7

### Тема: Вказівники. Робота з динамічною пам'яттю.

(1 завдання, оцінюється в 2 бали)

#### Теоретичні відомості

Вказівники та робота з динамічною пам'яттю є важливою частиною мови програмування C++. Вони надають можливість динамічно виділяти та звільняти пам'ять, а також працювати з адресами пам'яті.

**Вказівники** є змінними, які містять адресу в пам'яті. Вони вказують на місце розташування іншого об'єкта або змінної.

```
тип_даних *вказівник;
```

Наприклад:

```
int number = 42;
int *pointer = &number; // вказівник, що вказує на адресу
                        // змінної number
```

\* – розіменування вказівника, тобто отримання значення, на яке вказує вказівник.

& – отримання адреси змінної.

```
int value = *pointer; // Значення, на яке вказує вказівник
int *newPointer = &value; // Вказівник, що вказує на адресу
                        // змінної value
```

В C++, для виділення пам'яті використовується оператор `new`, а для звільнення – `delete`. Динамічна пам'ять дозволяє створювати об'єкти, що існують до тих пір, доки їх не буде видалено.

```
тип_даних *вказівник = new тип_даних;
```

Наприклад:

```
int *dynamicNumber = new int; // Виділення пам'яті для
//цілочисельного значення
*dynamicNumber = 123; // Запис значення в динамічно виділену
//пам'ять
```

Пам'ять слід звільнити оператором `delete`:

```
delete dynamicNumber; // Звільнення динамічно виділеної пам'яті
```

Динамічні масиви можна створити за допомогою оператора `new[]`:

```
тип_даних *вказівник_на_масив = new тип_даних[розмір];
```

Наприклад:

```
// Створення динамічного масиву з 5 цілих чисел  
int *dynamicArray = new int[5];
```

Пам'ять для динамічного масиву слід звільнити оператором delete[]:

```
// Звільнення динамічно виділеної пам'яті для масиву  
delete[] dynamicArray;
```

Динамічна пам'ять дозволяє більш гнучко управляти ресурсами та створювати структури даних за часом виконання програми. Однак її використання вимагає обережності, оскільки невірне використання може призвести до витоку пам'яті та інших проблем з роботою програми.

### Завдання (2 бали)

Виконати завдання варіанту використовуючи динамічну пам'ять та вказівники на неї. До елементів масиву звертатись за допомогою вказівників (Звернення до елементів через індекс (напр.  $A[i]$ ) не дозволяється).

|    |   |
|----|---|
| 1. | Задано два масиви цілих чисел $A[n]$ , $n \leq 500$ і $B[m]$ , $m \leq 200$ . Обчислити суму об'єднання $A \cup B$ . Об'єднання $A \cup B$ – це множина елементів масиву $A$ і $B$ , взятих по одному разу. Вивести елементи об'єднання та їх суму.   |
| 2. | Задано дійсних масив чисел $A[n]$ , $n \leq 500$ . Написати програму, яка обчислює добуток всіх чисел, які знаходяться між першим і останнім від'ємними елементами цього масиву і виводить цей діапазон. Якщо від'ємних чисел немає або є тільки одне, або вони стоять поряд, то виводить відповідне повідомлення.                              |
| 3. | Задано масив чисел $A[2n]$ , $n \leq 200$ . Написати програму, яка обчислює значення:<br>$Z = \max(\min(A[0] \dots A[2n-1]), \min(A[1] \dots A[2n-2]), \dots, \min(A[n-1] \dots A[n])).$  |
| 4. | Задано масив дійсних чисел $A[n * m]$ , $n \leq 300$ і $m \leq 25$ . Розробити програму, яка обчислює<br>$V = \min(\max(A[0], \dots, A[m-1]), \max(A[m], \dots, A[2m-1]), \dots, \max(A[m * (n-1)], \dots, A[n * m-1])).$<br>У масиві $n$ груп по $m$ чисел. Мінімальне значення шукається серед максимальних значень елементів у кожній групі. |

|     |   |
|-----|---|
| 5.  | Дано масив цілих чисел $A[n]$ , $n \leq 300$ . Розробити програму, яка обчислює суму елементів масиву, якщо він упорядкований за зростанням або добуток, якщо не упорядкований. Масив вважається упорядкованим за зростанням якщо $A[i] \leq A[i+1]$ для $i = 0, 1, 2, \dots, n-2$ .            |
| 6.  | Задано масив дійсних чисел $A[n]$ , $n \leq 500$ . Розробити програму, яка обчислює добуток всіх чисел, які повторюються, і суму всіх чисел, які не повторюються. Якщо чисел що повторюються або не повторюються немає, то виводить відповідне повідомлення.                                    |
| 7.  | Задано масив цілих чисел $A[n]$ , $n \leq 200$ . Написати програму, яка виводить на екран всі числа, що повторюються, по одному разу. Якщо чисел, що повторюються, немає, то виводить повідомлення про це.  |
| 8.  | Задано масив дійсних чисел $A[n]$ , $n \leq 100$ . Написати програму, яка обчислює добуток елементів цього масиву, якщо масив упорядкований за зростанням або суму, якщо масив не упорядкований. Масив буде упорядкованим за зростанням якщо $A[i] \leq A[i+1]$ для $i = 0, 1, 2, \dots, n-2$ . |
| 9.  | Задано масив дійсних чисел $A[n]$ , $n \leq 400$ . Написати програму, яка визначає кількість додатних елементів, що містяться між його максимальним та мінімальним елементом.   |
| 10. | Задано масив чисел $A[n]$ , $n \leq 300$ . Написати програму, яка замінює всі від'ємні елементи на середнє арифметичне елементів масиву   |
| 11. | Дано два масиви цілих чисел $X[n]$ і $Y[n]$ , $n \leq 300$ . Розробити програму, яка обчислює суму елементів об'єднання цих масивів. Об'єднанням двох масивів є взяті по одному разу всі елементи масиву $X[n]$ і $Y[n]$ . Вивести масив об'єднання та їх суму.                                 |
| 12. | Задано масив цілих чисел $A[n]$ , $n \leq 400$ . Написати програму, яка знаходить максимальне серед тих чисел, які не повторюються. Якщо таких чисел немає, то виводить повідомлення про це.  |
| 13. | Задано два масиви цілих чисел $A[n]$ , $n \leq 500$ і $B[m]$ , $m \leq 200$ . Обчислити суму об'єднання $A \cup B$ . Об'єднання $A \cup B$ – це множина елементів масиву $A$ і $B$ , взятих по одному разу. Вивести елементи об'єднання та їх суму.   |
| 14. | Задано масив цілих чисел $A[2n]$ , $n \leq 200$ . Розробити програму, яка добуток тих чисел із $A[n], \dots, A[2n-1]$ , які менші за середнє арифметичне значення чисел $A[0], \dots, A[n-1]$ . Якщо таких чисел  |

|     |   |
|-----|---|
|     | немає, то виводить повідомлення про це.   |
| 15. | Задано масив дійсних чисел $A[n]$ , $n \leq 500$ . Розробити програму, яка обчислює добуток всіх чисел, які повторюються, і суму всіх чисел, які не повторюються. Якщо чисел що повторюються або не повторюються немає, то виводить відповідне повідомлення.  |
| 16. | Задано масив дійсних чисел $A[n]$ , $n \leq 200$ . Розробити програму, яка записує в масив $B[n]$ по одному разу всі повторювані числа в масиві $A[n]$ . Виводить повідомлення, якщо таких чисел немає.   |
| 17. | Задано масив дійсних чисел $A[n*m]$ , $n \leq 100$ , $m \leq 10$ . Розробити програму, яка обчислює<br>$Z = \max(\min(A[0], \dots, A[m-1]), \min(A[m], \dots, A[2m-1]), \dots, \min(A[m*(n-1)], \dots, A[n*m-1]))$ .<br>У масиві з $n$ груп по $m$ чисел. Максимальне значення визначається серед мінімальних значень елементів кожної групи.                       |
| 18. | Задано дійсні величини $a$ , $b$ ( $a < b$ ) і масив чисел $X[n]$ , $n < 200$ . Розробити програму, яка обчислює суму всіх $X[i] < a$ , добуток всіх $X[i] > b$ і знаходить $\max X[i]$ та $\min X[i]$ серед $X[i] \in (a, b)$ , $i = 0, 1, 2, \dots, n-1$ . Якщо чисел $X[i] < a$ , або $X[i] > b$ , або $X[i] \in (a, b)$ немає, то виводить повідомлення про це. |

## Лабораторна робота № 8

### Тема: Рядки

(2 завдання, оцінюється в 3 бали)

#### Теоретичні відомості

В C ++ для зберігання рядків використовують символні масиви. Це такі ж масиви, як ми з вами вже розглядали в темі про масиви в C ++, але зберігають вони не числові дані, а символні.

Символи такого масиву розташовані послідовно в сусідніх комірках пам'яті – в кожній комірці зберігається один символ і займає один байт. Останнім символом кожного такого рядка є символ \0 (нульовий символ).  
наприклад:

```
char str[12] = {'H', 'e', 'l', 'l', 'o', ' ', 'w', 'o', 'r', 'l', 'd', '\0'};  
або  
char str[] = "Hello World";
```

Тепер розглянемо введення рядків з клавіатури детальніше. У C ++ є функція `getline()`, яку ми можемо використовувати разом з `cin`. `cin.getline()` зчитує в масив весь рядок з пробілами і табуляцією, поки не відбудеться натиснення Enter або поки не буде перевищено розмір масиву. Символ нового рядка в масиві не збережеться, а заміниться на нульовий символ.

```
//передаємо у функцію ім'я масиву та його розмір  
cin.getline(str, 128);
```

#### Функції для роботи з рядками в C++

| Функція                            | Пояснення  |
|------------------------------------|--|
| <code>strlen (str)</code>          | визначає довжину зазначеного рядка, без урахування нуль-символу  |
| <b><i>Копіювання строк</i></b>     |  |
| <code>strcpy(str, str2)</code>     | виконує побайтове копіювання символів з рядка <code>str2</code> у рядок <code>str</code>   |
| <code>strncpy(str, str2, n)</code> | виконує побайтове копіювання <code>n</code> символів з рядка <code>str2</code> у рядок <code>str</code> . повертає значення <code>str</code> |
| <b><i>Конкатенація строк</i></b>   |  |
| <code>strcat(str, str2)</code>     | об'єднує рядок <code>str2</code> з рядком <code>str</code> . Результат зберігається в <code>str</code>                                       |
| <code>strncat(str, str2,n)</code>  | об'єднує <code>n</code> символів рядка <code>str2</code> з рядком <code>str</code> . Результат   |

|                         |   |
|-------------------------|---|
|                         | зберігається в str  |
| <b>Порівняння строк</b> |   |
| strcmp(str, str2)       | порівнює рядок str з рядком str2 і повертає результат типу int: 0 -якщо рядки еквівалентні, > 0 - якщо str <str2, <0 - якщо str > str2 З урахуванням регістру                 |
| strncmp(str, str2,n)    | порівнює n символів рядка str з рядком str2 і повертає результат типу int: 0 – якщо рядки еквівалентні, > 0 – якщо str < str2, < 0 - якщо str > str2 З урахуванням регістру   |
| stricmp(str, str2)      | порівнює рядок str з рядком str2 і повертає результат типу int: 0 – якщо рядки еквівалентні, > 0 – якщо str < str2, < 0 – якщо str > str2. Без урахування регістру            |
| strnicmp(str, str2,n)   | порівнює n символів рядка str з рядком str2 і повертає результат типу int: 0 – якщо рядки еквівалентні, > 0 – якщо str < str2, < 0 – якщо str > str2. Без урахування регістру |
| <b>Обробка символів</b> |   |
| isalnum(c)              | повертає значення true, якщо c є буквою або цифрою і false у інших випадках   |
| isalpha(c)              | повертає значення true, якщо c є буквою і false у інших випадках  |
| isdigit(c)              | повертає значення true, якщо c є цифрою і false у інших випадках  |
| islower(c)              | повертає значення true, якщо c є буквою нижнього регістру і false у інших випадках  |
| isupper(c)              | повертає значення true, якщо c є буквою верхнього регістру і false у інших випадках   |
| isspace(c)              | повертає значення true, якщо c є пробілом і false у інших випадках  |
| toupper(c)              | якщо символ c, є символом нижнього регістра, то функція повертає перетворений символ з верхнього регістру, інакше символ повертається без змін.                               |
| <b>Функції пошуку</b>   |   |
| strchr(s,c)             | пошук першого входження символу c в рядку s. У разі вдалого пошуку повертає вказівник на місце першого входження символу c. Якщо символ не знайдено, то повертається нуль.    |
| strcspn(str, str2)      | визначає довжину початкового сегмента рядка str, що   |



|  |  |
|--|--|
|  | містить ті символи, які не входять в рядок str2  |
| strspn(str, str2)  | повертає довжину початкового сегмента рядка str, що містить тільки ті символи, які входять в рядок str2  |
| strprbk(str, str2)                                       | Повертає вказівник першого входження будь-якого символу рядка str2 в рядку str   |
| <b>Функції перетворення</b>                              |  |
| atof(str)  | перетворює рядок str в тип double  |
| atoi(str)  | перетворює рядок str в тип int   |
| atol(str)  | перетворює рядок str в тип long int  |
| <b>Функції стандартної бібліотеки вводу/виводу stdio</b> |  |
| getchar(c)   | зчитує символ c зі стандартного потоку введення, повертає символ в форматі int   |
| gets(s)  | зчитує потік символів зі стандартного пристрою вводу в рядок s до тих пір, поки Ви не натиснете ENTER  |
| <b>Функції стандартної бібліотеки string.h</b>           |  |
| char *strtok (char *st, const char *sep);                | розбиття рядка на лексеми (сегменти), обмежені символами, що входять до складу рядка sep. Цей параметр може містити будь-яку кількість різних обмежників – ознак границь лексем, після виділення лексеми в рядок st поміщається символ «\0». |

Приклад:

Скласти програму, яка вводить речення, здійснює розбиття його на слова, підраховує кількість символів у кожному сло-ві та виводить відповідну інформацію

```
#include <string.h>
#include <iostream>
using namespace std;
int main ()
{ char *tk, *spt=", .!";
  char st[] = "Hi! It is my favorite subject";
  cout<<st<<endl;
  int i=1;
  tk=strtok(st, spt);
  while (tk != NULL)
  {
  cout<<i<< " word - " << tk << " - has " << strlen(tk) << "
symbols" << endl;
  tk=strtok(NULL,spt );
  i++;
}
```

```

    }
    return 0;
}

```

### Завдання 1 (1 бал)

Виконати завдання варіанту використовуючи функції для роботи з рядками

|     |   |
|-----|---|
| 1.  | Задано рядок символів. Визначити, чи є заданий символ <i>c</i> в цьому рядку символів   |
| 2.  | Задано деякий рядок символів. У заданому тексті замінити послідовність символів '...' на '.'.   |
| 3.  | Задано деякий рядок символів. У заданому тексті визначити позицію першої крапки '.'.  |
| 4.  | Задано деякий рядок символів. Створити новий рядок, який утворено із даного читанням з кінця до початку.  |
| 5.  | Задано рядок символів. Підрахувати кількість символів '+' у цьому рядку.  |
| 6.  | Задано деякий текст. Посортувати слова в алфавітному порядку.   |
| 7.  | Задано деякий рядок символів. Визначити позиції останньої крапки '.' ' в тексті.  |
| 8.  | Нехай задано деякий текст. Обчислити, скільки разів повторюється наперед заданий символ .   |
| 9.  | Задано деякий текст. З заданого тексту вибрати і надрукувати ті символи, які зустрічаються в ньому рівно два рази (в тому порядку, як вони зустрічаються в тексті). |
| 10. | Задано деякий текст. Знайти найдовше симетричне слово заданого рядка.   |
| 11. | Задано деякий текст. Вивести слова, в яких немає повторюваних букв.   |
| 12. | Задано деякий рядок символів. Вивести слова, які зустрічається в рядку 2 рази.  |
| 13. | Задано деякий рядок символів. Вивести послідовність слів, в яких літери впорядковані за алфавітом.  |
| 14. | Задано деякий рядок символів. Вивести слова рядку, що починаються і закінчуються однією і тією ж буквою.  |
| 15. | Задано деякий текст. Перерахувати всі слова заданого рядка, які починаються з тієї ж букви, що і перше слово рядка.   |
| 16. | Задано деякий рядок символів. Вивести слова рядку, що містять 2 рази введений з клавіатури символ (наприклад, слово «програмування» містить 2 літери «н»).          |

|     |  |
|-----|--|
| 17. | Задано деякий рядок символів. Визначити останню позицію в тексті введеного символу. Якщо символу немає, вивести відповідне повідомлення. |
| 18. | Задано деякий рядок символів. У заданому тексті знайти символ, що зустрічається найчастіше.  |

**Завдання 2 (2 бали)**

|    |   |
|----|---|
| 1. | Ввести з клавіатури текст, в якому слова розділені пробілами і різними розділовими знаками. Написати програму, яка видаляє з введеного тексту всі слова з подвоєнням літер і записує їх в окремий рядок, розділяючи пробілами. Друкує окремо вилучені слова і текст, що залишився після вилучення слів. |
| 2. | Ввести з клавіатури текст, в якому слова розділені пробілами і різними розділовими знаками. Написати програму, яка видаляє з введеного тексту всі слова найбільшої довжини. (Слів найбільшої довжини може бути декілька). Друкує текст, що залишився після вилучення слів.                              |
| 3. | Ввести з клавіатури текст, в якому слова розділені пробілами і різними розділовими знаками. Написати програму, яка знаходить і друкує всі симетричні слова (наприклад, слова абооба і абоба є симетричними).  |
| 4. | Ввести з клавіатури текст, в якому слова розділені пробілами і різними розділовими знаками. Написати програму, яка видаляє в кожному слові початкового тексту всі наступні входження першої літери.   |
| 5. | Ввести з клавіатури текст, в якому слова розділені пробілами і різними розділовими знаками. Написати програму, яка видаляє в кожному слові початкового тексту всі попередні входження останньої літери.   |
| 6. | Ввести з клавіатури текст, в якому слова розділені пробілами і різними розділовими знаками. Написати програму, яка перетворює рядок так, щоб букви кожного слова в ньому були відсортовані по спаданню.   |
| 7. | Ввести з клавіатури текст, в якому слова розділені пробілами і різними розділовими знаками. Написати програму, яка знаходить і видаляє всі слова, що входять в цей текст по одному разу.  |

|     |   |
|-----|---|
| 8.  | Ввести з клавіатури текст, в якому слова розділені пробілами і різними розділовими знаками. Написати програму, яка в словах непарної довжини початкового тексту видаляє середню літеру.                                     |
| 9.  | Ввести з клавіатури текст, в якому слова розділені пробілами і різними розділовими знаками. Написати програму, яка знаходить і друкує всі слова, що входять у заданий текст по одному разу.                                 |
| 10. | Ввести з клавіатури два тексти, слова в яких розділені пробілами і різними розділовими знаками. Написати програму, яка видаляє із другого тексту всі входження слів першого тексту.   |
| 11. | Ввести з клавіатури два тексти, слова в яких розділені пробілами і різними розділовими знаками. Написати програму, яка створює третій текст із слів першого тексту, які не входять у другий текст, розділяючи їх пробілами. |
| 12. | Ввести з клавіатури символ і текст, в якому слова розділені пробілами і різними розділовими знаками. Написати програму, яка знаходить і друкує всі слова, що містять заданий символ найбільшу кількість разів.              |
| 13. | Ввести з клавіатури масив слів. Скласти речення зі слів, які починаються на заданий символ, попередньо видаливши з слова всі такі символи крім першого.   |
| 14. | Ввести з клавіатури текст, в якому слова розділені пробілами і різними розділовими знаками. знайти і видалити слова з парною кількістю символів.  |
| 15. | Ввести з клавіатури текст, в якому слова розділені пробілами і різними розділовими знаками. Написати програму, яка видаляє з введеного тексту всі повторні входження слів.  |
| 16. | Ввести з клавіатури текст, в якому слова розділені пробілами і різними розділовими знаками. Написати програму, яка видаляє з введеного тексту всі слова заданої довжини.  |
| 17. | Ввести з клавіатури два тексти, слова в яких розділені пробілами і різними розділовими знаками. Написати програму, яка створює третій текст із слів першого тексту, які входять у другий текст, розділяючи їх пробілами.    |
| 18. | Ввести з клавіатури два тексти, слова в яких розділені пробілами і різними розділовими знаками. Скласти текст зі слів, які не містять заданий символ.   |

## МОДУЛЬ 2. СТРУКТУРИ ДАНИХ МОВОЮ C++

### Лабораторна робота № 9

#### Тема: Функції користувача

(2 завдання, оцінюється в 3 бали)

#### Теоретичні відомості

У C++, функції користувача дозволяють групувати певний блок коду в окрему одиницю, яку можна викликати з інших частин програми. Функції полегшують кодування, роблять програми більш структурованими та зменшують дублювання коду.

#### *Оголошення функції*

Функцію потрібно оголосити перед її використанням. Оголошення включає ім'я функції, її тип, та, при необхідності, параметри.

```
тип_даних ім'я_функції(пар1_тип пар1_ім'я, пар2_тип пар2_ім'я, ...);
```

Наприклад:

```
// Оголошення функції "add" для додавання двох чисел
int add(int a, int b);
```

#### *Визначення функції*

Визначення функції містить реалізацію коду функції, що вказана між фігурними дужками {}.

```
тип_даних ім'я_функції(пар1_тип пар1_ім'я, пар2_тип пар2_ім'я, ...) {
// тіло функції
// повернення_значення;
}
```

Наприклад:

```
int add(int a, int b) {
return a + b;
}
```

Функцію можна викликати в іншій частині програми, передавши їй аргументи.

```
тип_даних результат = ім'я_функції(аргумент1, аргумент2, ...);
```

Наприклад:

```
int sum = add(5, 3); // Виклик функції "add" з аргументами 5 та 3
```

### ***Параметри та повернення значення***

Функції можуть мати параметри, які передаються для обробки, та повертати значення за допомогою оператора return.

```
тип_даних ім'я_функції(пар1_тип пар1_ім'я, пар2_тип пар2_ім'я, ...)  
{  
    // тіло функції  
    return результат;  
}
```

Наприклад:

```
int square(int x) {  
    return x * x;  
}
```

### ***Прототипи функцій***

Прототипи функцій (функцій без тіла) можуть бути використані для оголошення функцій перед їх визначенням.

```
тип_даних ім'я_функції(пар1_тип пар1_ім'я, пар2_тип пар2_ім'я, ...);
```

Прототипи дозволяють компілятору знати, що така функція буде визначена пізніше у коді.

```
int add(int a, int b); // Прототип функції
```

### ***Рекурсія***

Рекурсія – це процес, коли функція викликає саму себе.

Рекурсивна функція повинна мати правило досяжності – досягну умову виходу з рекурсії, щоб уникнути безкінечного виклику.

```
int factorial(int n) {  
    // правило досяжності  
    if (n == 0 || n == 1) {  
        return 1;  
    }  
}
```

```
// Рекурсивний виклик
return n * factorial(n - 1);
}
```

Кожен виклик рекурсивної функції додає новий запис у стек пам'яті. У випадку безкінечної рекурсії це може призвести до переповнення стека.

Рекурсія використовується там, де алгоритм може бути виражений в термінах самого себе.

```
// Приклад: обчислення чисел Фібоначчі рекурсивно
int fibonacci(int n) {
    if (n <= 1) {
        return n;
    }
    return fibonacci(n - 1) + fibonacci(n - 2);
}
```

Рекурсія може робити код більш зрозумілим та компактним, але слід бути обережним при використанні її для великих значень аргументів, оскільки це може призвести до переповнення стека.

Отже, функції користувача в С++ є важливим елементом структурованого програмування, дозволяючи розділяти код на логічні блоки та полегшуючи його обслуговування та розширення.

### **Завдання 1 (1.5 бали)**

|    |  |
|----|--|
| 1. | Дано текст, в якому слова розділені пробілами і розділовими знаками. Створити програму, яка знаходить і друкує всі слова, що входять у Даний текст по одному разу. Створити і викликати власну функцію для перевірки входження слова в текст і функцію для виводу слів.  |
| 2. | Задано два масиви цілих чисел $A[n]$ , $n \leq 500$ і $B[m]$ , $m \leq 200$ . Обчислити суму об'єднання $A \cup B$ . Об'єднання $A \cup B$ – це множина елементів масиву $A$ і $B$ , взятих по одному разу. Створити і викликати власну функцію для побудови об'єднання і функцію для обчислення суми. Вивести на екран елементи об'єднання та їх суму.                            |
| 3. | Дана квадратна матриця $X[n][n]$ , $n \leq 15$ . Створити програму перетворення даної матриці так, щоб добутки елементів рядків утворювали не спадну послідовність. Створити і викликати власну функцію для перетворення матриці та функцію для обчислення добутку елементів вектора і використати її для обчислення добутків елементів рядків. Вивести на екран отриману матрицю. |

|     |  |
|-----|--|
| 4.  | Дана квадратна матриця $X[n][n]$ , $n \leq 15$ . Створити програму побудови матриці $Y[n][n]$ за правилом: $Y[i][j]$ дорівнює скалярному добутку $i$ -го рядка на $j$ -ий стовпчик матриці $X$ . Створити функцію для побудови матриці та функцію для обчислення скалярного добутку векторів і використати її для обчислення скалярних добутків рядків і стовпчиків. Вивести на екран отриману матрицю.                            |
| 5.  | Дана квадратна матриця $X[n][n]$ , $n \leq 15$ . Створити програму перетворення Даної матриці так, щоб суми елементів стовпців утворювали не зростаючу послідовність. Створити функцію для перетворення матриці та функцію для обчислення суми елементів вектора і використати її для обчислення сум елементів стовпців. Вивести на екран отриману матрицю.  |
| 6.  | Дана квадратна матриця $X[n][n]$ , $n \leq 15$ . Створити програму, яка будує вектори: $A[i]$ – сума елементів $i$ -го рядка, $B[j]$ – сума елементів $j$ -го стовпчика Даної матриці, $i, j = 1, 2, \dots, n$ . Створити функцію для побудови вектора та функцію для обчислення суми елементів вектора (одновимірного масива) і використати її для обчислення сум елементів рядків і стовпців. Вивести на екран отримані вектори. |
| 7.  | Дано два масиви чисел $A[n]$ , ( $n \leq 300$ ) і $B[m]$ , ( $m \leq 100$ ). Створити програму обчислення суми $\sum_{x \in A \cap B} \sin x$ . ( $A \cap B$ – множина елементів $A$ , що входять у $B$ і взятих по одному разу). Створити і викликати власну функцію для побудови перерізу і функцію для обчислення суми. Вивести на екран елементи перерізу та їх суму.  |
| 8.  | Дана квадратна матриця $X[n][n]$ , $n \leq 15$ . Створити програму, яка будує вектор $B[k]$ , ( $k \leq n$ ) з номерів тих рядків матриці, елементи яких утворюють спадну послідовність. Якщо таких рядків немає, то друкує повідомлення про це. Створити і викликати власну функцію для побудови вектора і логічну функцію для перевірки умови того, що послідовність спадна.   |
| 9.  | Дано текст, в якому слова розділені пробілами і різними розділовими знаками. Створити програму, яка вилучає всі повторні входження кожного слова в цей текст. Створити і викликати власну функцію для вилучення слів і логічну функцію для перевірки входження слова в текст. Вивести на екран текст, що залишився після вилучення слів.   |
| 10. | Дана матриця $A[n][n]$ , $n \leq 15$ . Створити програму побудови вектора  |



|     |  |
|-----|--|
|     | <p><math>B[m]</math> (<math>m \leq n</math>) із номерів тих стовпців даної матриці, які упорядковані за спаданням. Створити функцію для побудови вектора та логічну функцію для перевірки умови упорядкованості послідовності і використати її для перевірки стовпців. Якщо таких рядків немає, то Вивести на екран повідомлення про це.</p>   |
| 11. | <p>Дано масиви цілих чисел <math>A[n]</math>, <math>n \leq 300</math> і <math>B[m]</math>, <math>m \leq 400</math>. Створити програму побудови симетричної різниці масивів <math>A \setminus B \cup B \setminus A</math> і пошуку її мінімального елемента. (<math>A \setminus B \cup B \setminus A</math> – множина елементів <math>A</math>, що не входять у <math>B</math> і множина елементів <math>B</math>, що не входять в <math>A</math> і взятих по одному разу). Створити і викликати власну функцію для побудови симетричної різниці і функцію для пошуку мінімального елемента. Вивести на екран елементи симетричної різниці та її мінімальний елемент.</p> |
| 12. | <p>Дано два тексти, слова в яких розділені пробілами і розділовими знаками. Створити програму побудови нового тексту, в який входять слова першого і другого текстів по одному разу, розділені пробілами. Створити і викликати власну функцію для побудови нового тексту і функцію перевірки входження слова в текст. Вивести на екран отриманий текст.</p>  |
| 13. | <p>Дано два тексти, слова в яких розділені пробілами і розділовими знаками. Створити програму побудови нового тексту, в який входять всі слова першого тексту, що не входять у другий текст, і всі слова другого тексту, що не входять в перший і розділені пробілами. Створити і викликати власну функцію для побудови нового тексту і функцію перевірки входження слова в текст. Вивести на екран отриманий текст.</p>   |
| 14. | <p>Дано текст, в якому слова розділені пробілами і різними розділовими знаками. Створити програму, яка в словах непарної довжини цього тексту вилучає середню літеру. Створити і викликати власну функцію для перевірки довжини слова в тексті і функцію для видалення середньої літери.</p>   |
| 15. | <p>Дано символ і текст, в якому слова розділені пробілами і різними розділовими знаками. Створити програму, яка знаходить і друкує всі слова, що містять даний символ найбільшу кількість разів. Створити і викликати власну функцію для підрахунку кількості входжень символу в слово і функцію для виводу слів.</p>  |
| 16. | <p>Дано масив дійсних чисел <math>A[n]</math>, <math>n \leq 200</math>. Створити програму, яка обчислює суму всіх чисел, які повторюються, і середнє арифметичне всіх чисел, які не повторюються. Якщо чисел що повторюються або</p>   |

|     |   |
|-----|---|
|     | не повторюються немає, то Вивести на екран відповідне повідомлення. Створити і викликати власну функцію функції для обчислення суми елементів масиву і функцію для перевірки елемента масиву на повтор.   |
| 17. | Дано текст, в якому слова розділені пробілами і різними розділовими знаками. Створити програму, яка вилучає з цього тексту всі слова даної довжини. Друкує текст, що залишився після вилучення слів. Створити і викликати власну функцію для перевірки довжини слова в тексті і функцію для виводу тексту.  |
| 18. | Дано масив цілих чисел $A[n]$ , $n \leq 200$ . Створити програму, яка переписує всі числа, що повторюються, з масиву $A[n]$ в масив $B[n]$ по одному разу. Якщо чисел що повторюються немає, то Вивести на екран повідомлення про це. Створити і викликати власну функцію для перевірки наявності числа в масиві і функцію для виводу масивів на екран. |

**Завдання 2** (оцінюється в 1.5 бали)

Задано натуральне  $n$ . Розробити програму для обчислення заданих сум. При обчисленні сум використати рекурсивні процедури або функції.

|    |  |
|----|--|
| 1. | $\sum_{k=1}^n \frac{a_k b_k}{k!}$ , $a_1 = 1$ , $a_k = 0.5(\sqrt{b_{k-1}} + 0.5\sqrt{a_{k-1}})$ ,<br>$b_1 = 1$ , $b_k = 2a_{k-1}^2 + b_{k-1}$ .                                  |
| 2. | $\sum_{i=1}^n \frac{x_i}{1+ y_i }$ , $x_1 = 1$ , $x_i = 0.3x_{i-1}$ ,<br>$y_1 = 1$ , $y_i = x_{i-1}^2 + y_{i-1}$ .   |
| 3. | $\sum_{k=1}^n \frac{2^k}{(1+a_k+b_k)}$ , $a_1 = 1$ , $a_k = 3b_{k-1} + 2a_{k-1}$ ,<br>$b_1 = 1$ , $b_k = a_{k-1}^2 + b_{k-1}$ .  |
| 4. | $\sum_{k=1}^n \frac{a_k b_k}{(k+1)!}$ , $a_1 = 1$ , $a_k = 0.3b_{k-1} + 0.2a_{k-1}$ ,<br>$b_1 = 1$ , $b_k = a_{k-1}^2 + b_{k-1}^2$ .   |
| 5. | $\sum_{i=1}^n \frac{x_i}{k!!}$ , $x_1 = 1$ , $x_2 = 2$ , $x_3 = 3$ ,<br>$x_i = x_{i-1} + x_{i-2} + x_{i-3}$ .  |
| 6. | $\sum_{k=1}^n \frac{a_k}{(k+1)!}$ , $a_k = \left(1 - \frac{1}{2!}\right)\left(1 + \frac{1}{3!}\right) \cdots \left(1 + \frac{(-1)^k}{(k+1)!}\right)$ .                           |
| 7. | $\sum_{k=1}^n \frac{a_k}{3^k}$ , $a_k = \frac{1}{2}\sqrt{\frac{1}{1} + \frac{1}{2}\sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{3} + \cdots + \frac{1}{2}\sqrt{\frac{1}{k}}}}$ . |

|     |  |
|-----|--|
| 8.  | $\sum_{k=1}^n \frac{(-1)^{k+1}}{(2k+1)!} a_k, a_0 = 1, a_k = 5 + \frac{1}{3^k} \sin^2(a_{k-1} - 1).$ |
| 9.  | $\sum_{k=1}^n \frac{(-1)^{k+1}}{k!!} a_k, a_0 = 2, a_k = \frac{\sin^2(a_{k-1} - 1)}{k^2}.$           |
| 10. | $\sum_{k=m}^n \frac{(-1)^k}{k!} \left(\frac{a_k}{2}\right)^k, a_0 = 1, a_k = \sqrt{ 4a_{k-1} + 2 }.$ |

### Завдання 2 (1,5 бали)

Задано натуральні  $n$  і  $m$ . Розробити програму для обчислення значень заданих виразів. При обчисленні виразів використати рекурсивні процедури або функції.

|     |   |
|-----|---|
| 11. | $\frac{3^m}{\sqrt{1! + \sqrt{2! + \dots + \sqrt{n!}}}}$   |
| 12. | $\frac{\sqrt{n + \sqrt{(n-1) + \dots + \sqrt{1}}}}{m!!}$  |
| 13. | $\frac{u_m}{\sqrt{n! + \sqrt{(n-1)! + \dots + \sqrt{1!}}}}, u_0 = 0, u_1 = 1,$<br>$u_m = u_{m-1} + u_{m-2},$  |
| 14. | $\frac{2^m}{\frac{1}{3} \sqrt{\frac{1}{1!}} + \frac{1}{3} \sqrt{\frac{1}{2!}} + \frac{1}{3} \sqrt{\frac{1}{3!}} + \dots + \frac{1}{3} \sqrt{\frac{1}{n!}}}$ |
| 15. | $\frac{m!}{\sqrt{2^n + \sqrt{2^{n-1} + \dots + \sqrt{2^1}}}}$   |
| 16. | $\frac{n^m}{\frac{1}{n} \sqrt{\frac{1}{1!}} + \frac{1}{n} \sqrt{\frac{1}{2!}} + \frac{1}{n} \sqrt{\frac{1}{3!}} + \dots + \frac{1}{n} \sqrt{\frac{1}{n!}}}$ |
| 17. | $\frac{m!}{\sqrt{m^n + \sqrt{m^{n-1} + \dots + \sqrt{m^1}}}}$   |
| 18. | $\frac{u_m}{\sqrt{1! + \sqrt{2! + \dots + \sqrt{n!}}}}, u_0 = 0, u_1 = 1,$<br>$u_m = u_{m-1} + u_{m-2},$  |

## Лабораторна робота № 10

### Тема: Шаблони функцій

(1 завдання, оцінюється в 2 бали)

#### Теоретичні відомості

Шаблони функцій дозволяють писати універсальні функції, які можуть працювати з різними типами даних. Вони дозволяють створювати код, який є гнучким та здатним працювати з різними типами без повторення коду.

Оголошення шаблонної функції виглядає наступним чином:

```
template <typename T>
T add(T a, T b) {
return a + b;
}
```

Тут typename T вказує, що шаблон приймає тип T, який може бути будь-яким типом даних.

Використання шаблонів включає вказівку типу даних під час виклику функції.

```
// Виклик шаблонної функції для цілих чисел
int sumInt = add(3, 5);
// Виклик шаблонної функції для дробових чисел
double sumDouble = add(2.5, 3.7);
```

Шаблони можуть бути багатопараметричними, тобто приймати більше одного типу даних.

```
template <typename T, typename U>
T maxValue(T a, U b) {
return (a > b) ? a : b;
}
```

Наприклад:

```
// Шаблон з двома типами даних
int resultInt = maxValue(10, 15.5);
// Шаблон з двома типами даних
double resultDouble = maxValue(3.14, 2.71);
```

Шаблони функцій дозволяють створювати гнучкий та зручний код, який може працювати з різними типами даних без повторення структури функції для кожного типу. Вони є важливою частиною сучасного C++ та використовуються в багатьох бібліотеках та фреймворках.

Завдання 1 (2 бали)

|     |  |
|-----|--|
| 1.  | Задана матриця $A[n][n]$ , $n \leq 15$ . Розробити шаблонну функцію, яка міняє місцями діагональний елемент з мінімальним елементом у цьому рядку.   |
| 2.  | Задана матриця $A[n][n]$ , $n \leq 10$ . Назвемо сусідами елемента матриці $A[i][j]$ елементи $A[k][l]$ з індексами $i-1 \leq k \leq i+1$ , $j-1 \leq l \leq j+1$ , $(i, j) \neq (k, l)$ . Розробити шаблонну функцію, яка буде матрицю $B[n][n]$ за правилом: $B[i][j]$ є середнім арифметичним значенням усіх сусідів елемента $A[i][j]$ , $i, j = 1, 2, \dots, n$ . |
| 3.  | Задано масиви чисел $A[n]$ , ( $n \leq 400$ ) і $B[m]$ , ( $m \leq 200$ ). Розробити шаблонну функцію обчислення добутку елементів симетричної різниці масивів $A \setminus B \cup B \setminus A$ . ( $A \setminus B \cup B \setminus A$ – набір елементів $A$ , що не входять у $B$ , і набір елементів $B$ , що не входять в $A$ , взятих по одному разу).           |
| 4.  | Задано матрицю $A[n][n]$ , $n \leq 15$ . Розробити шаблонну функцію для побудови масиву $X[i]$ , $i = 1, 2, \dots, n$ , за правилом: $X[i]$ дорівнює середньому арифметичному значенню елементів $i$ -го рядка матриці.  |
| 5.  | Задана дійсна матриця $A[n][n]$ , $n \leq 20$ . Розробити шаблонну функцію, яка буде масив $X[n]$ за правилом: $X[i]$ дорівнює максимуму з тих елементів, які розміщені між першим і останнім включно додатними елементами $i$ -го стовбця. Якщо додатних елементів у $i$ -му стовбці немає або є тільки один, або вони розміщені поряд, то $X[i] = -1$ .              |
| 6.  | Задана дійсна матриця $A[n][m]$ . Розробити шаблонну функцію, яка буде масив $B[n]$ , де $B[k] = 1$ , якщо $k$ -й рядок матриці $A$ симетричний (тобто, перший елемент дорівнює останньому, другий – передостанньому і т. д.: 1, 2, 3, 3, 2, 1 або 1, 2, 3, 5, 3, 2, 1), і $B[k] = 0$ – в протилежному випадку.  |
| 7.  | Задана матриця $A[n][n]$ , $n \leq 20$ . Розробити шаблонну функцію для побудови масиву $B[i]$ , $i = 1, 2, \dots, n$ , за правилом: $B[i]$ дорівнює кількості різних елементів $i$ -го рядка матриці $A$ .  |
| 8.  | Задана матриця $X[n][n]$ , $n \leq 20$ . Назвемо слідом матриці суму елементів головної діагоналі. Розробити шаблонну функцію обчислення слідів матриць $X, X^2$ .   |
| 9.  | Задано масиви чисел $A[n]$ , $n \leq 300$ і $B[m]$ , $m \leq 400$ . Розробити шаблонну функцію обчислення суми модулів елементів перерізу масивів $A \cap B$ . ( $A \cap B$ – набір елементів $A$ , які містяться в $B$ і взятих по одному разу).  |
| 10. | Задано матрицю $X[n][n]$ , $n \leq 20$ . Розробити шаблонну функцію для  |

|     |  |
|-----|--|
|     | побудови масиву $Y[i]$ , $i=1,2,\dots,n$ , за правилом: $Y[i]$ дорівнює добутку квадратів тих елементів $i$ -го рядка матриці, модулі яких належать проміжку $[a,b]$ , якщо таких елементів немає, то $Y(i)=0$ .   |
| 11. | Задана матриця $X[n][n]$ , $n \leq 10$ . Розробити шаблонну функцію, яка упорядковує за зростанням ті рядки, в яких міститься максимальний елемент матриці (максимальних елементів може бути декілька).  |
| 12. | Задано масиви чисел $A[n]$ , $n \leq 300$ і $B[m]$ , $m \leq 400$ . Розробити шаблонну функцію для побудови симетричної різниці масивів $A \setminus B \cup B \setminus A$ і пошуку її мінімального елемента. ( $A \setminus B \cup B \setminus A$ – множина елементів $A$ , що не входять у $B$ і множина елементів $B$ , що не входять в $A$ і взятих по одному разу). |
| 13. | Задана матриця $X[n][n]$ , $n \leq 15$ . Розробити шаблонну функцію, яка будує масиви: $A[i]$ – сума елементів $i$ -го рядка, $B[j]$ – сума елементів $j$ -го стовпчика заданої матриці, $i, j = 1, 2, \dots, n$ .   |
| 14. | Задано масиви чисел $A[n]$ , ( $n \leq 400$ ) і $B[m]$ , ( $m \leq 200$ ). Розробити шаблонну функцію для обчислення добутку елементів симетричної різниці масивів $A \setminus B \cup B \setminus A$ . ( $A \setminus B \cup B \setminus A$ – набір елементів $A$ , що не входять у $B$ , і набір елементів $B$ , що не входять в $A$ , взятих по одному разу).         |
| 15. | Задано матрицю $A[n][n]$ , $n \leq 15$ . Розробити шаблонну функцію для побудови масиву $X(i)$ , $i=1,2,\dots,n$ , за правилом: $X[n]$ дорівнює середньому арифметичному значенню елементів $i$ -го рядка матриці.   |
| 16. | Задана матриця $A[n][n]$ , $n \leq 15$ . Розробити шаблонну функцію побудови масиву $B[k]$ , ( $k \leq n$ ) із номерів рядків матриці, елементи яких є симетричними послідовностями виду 1,2,3,3,2,1 або 1,2,3,5,3,2,1.  |
| 17. | Задано дві цілочислові матриці $A[n][n]$ і $B[n][n]$ , $n \leq 20$ . Розробити шаблонну функцію для побудови масиву $X[n]$ за правилом: $X[n]=1$ , якщо кількість від'ємних елементів $i$ -го рядка першої матриці дорівнює кількості від'ємних елементів $i$ -го рядка другої матриці та $X[i]=0$ в інших випадках.   |
| 18. | Задана дійсна матриця $A[n][n]$ , $n \leq 20$ . Розробити шаблонну функцію для побудови масиву $X[n]$ за правилом: $X[i]$ дорівнює сумі модулів тих елементів, які розміщені між першим і останнім включно додатними елементами $i$ -го рядка. Якщо додатних елементів у $i$ -му рядку немає або є тільки один, або вони розміщені поряд, то $X[i]=-1$ .                 |

## Лабораторна робота № 11

### Тема: Структури

(1 завдання, оцінюється в 4 бали)

#### Теоретичні відомості

Структури дозволяють об'єднувати різні типи даних під одним ім'ям, створюючи таким чином складні дані. Структури в C++ можуть містити поля (члени), які можуть бути різних типів, і вони є основною будівельною одиницею для створення складних даних в програмах.

Структура оголошується за допомогою ключового слова `struct`:

```
struct Person {  
    // Члени структури  
    char name[10];  
    int age;  
    double height;  
};
```

Після оголошення можна створити змінну цієї структури:

```
Person person1; // Створення змінної типу Person
```

До членів структури можна отримати за допомогою оператора крапки (`.`):

```
person1.name = "John";  
person1.age = 25;  
person1.height = 1.75;
```

Екземпляри структури можна ініціалізувати при створенні:

```
Person person2 = {"Alice", 30, 1.65};
```

Структури можуть бути вкладеними в інші структури:

```
struct Address {  
    char city[15];  
    char street[15];  
    int zipCode;  
};  
  
struct Person {  
    char name[10];  
    int age;  
    Address address; // Вкладена структура  
};
```

Структури можуть містити різні типи даних, включаючи вкладені структури, вказівники, та масиви.

```

struct Student {
char name[10];
int age;
double grades[5];
Address homeAddress;
Student* nextStudent;//Вказівник на інший об'єкт структури
};

```

Структури в C++ використовуються для створення складних типів даних, які можуть об'єднувати різні типи та об'єднувати їх у зручний для роботи спосіб. Вони є основною складовою для організації та моделювання даних в програмах.

**Завдання 1 (4 бали)**

|           |   |
|-----------|---|
| <p>1.</p> | <p>1. Створити структуру з ім'ям STUDENT, яка містить наступні поля:</p> <ul style="list-style-type: none"> <li>• ПІБ;</li> <li>• номер групи;</li> <li>• успішність (масив з п'яти елементів).</li> </ul> <p>2. Розробити програму, яка виконує такі дії:</p> <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу STUDENT; записи мають бути впорядковані за спаданням номера групи;</li> <li>• вивести на екран номери груп та прізвища для всіх студентів, включених в масив, якщо середній бал студента більше 4.0;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul> |
| <p>2.</p> | <p>1. Створити структуру з ім'ям STUDENT, яка містить наступні поля:</p> <ul style="list-style-type: none"> <li>• ПІБ;</li> <li>• номер групи;</li> <li>• успішність (масив з п'яти елементів).</li> </ul> <p>2. Розробити програму, яка виконує такі дії:</p> <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу STUDENT; записи мають бути впорядковані за зростанням середнього бала;</li> <li>• вивести на екран номерів груп та прізвища для всіх студентів, які мають оцінки 4 і 5;</li> <li>• вивести на екран відповідне повідомлення, якщо таких</li> </ul>   |



|    |   |
|----|---|
|    | записів не знайдено.  |
| 3. | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям STUDENT, яка містить наступні поля: <ul style="list-style-type: none"> <li>• ПІБ;</li> <li>• помер групи;</li> <li>• успішність (масив з п'яти елементів).</li> </ul> </li> <li>2. Розробити програму, яка виконує такі дії: <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу STUDENT; записи мають бути впорядковані за алфавітом;</li> <li>• вивести на екран прізвищ та номерів груп для всіх студентів, які мають хоча б одну оцінку 2;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul> </li> </ol>   |
| 4. | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям AIRPORT, яка містить наступні поля: <ul style="list-style-type: none"> <li>• назва пункту призначення рейсу;</li> <li>• номер рейсу;</li> <li>• тип літака.</li> </ul> </li> <li>2. Розробити програму, яка виконує такі дії: <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу AIRPORT; записи мають бути впорядковані по зростанню номера рейсу;</li> <li>• вивести на екран номерів рейсів і типів літаків, що вилітають в пункт призначення, назва якого співпала з назвою, введеною з клавіатури;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul> </li> </ol> |
| 5. | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям AIRPORT, яка містить такі нуля: <ul style="list-style-type: none"> <li>• назва пункту призначення рейсу;</li> <li>• номер рейсу;</li> <li>• тип літака.</li> </ul> </li> <li>2. Розробити програму, яка виконує такі дії: <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу AIRPORT; записи мають бути розміщені в алфавітному порядку за назвами пунктів призначення;</li> </ul> </li> </ol>  |

|    |  |
|----|--|
|    | <ul style="list-style-type: none"> <li>• вивести на екран пунктів призначення і номерів рейсів, що обслуговуються літаком, тип якого введено з клавіатури;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul>   |
| 6. | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям EMPLOYEE, яка містить наступні поля: <ul style="list-style-type: none"> <li>• ПІБ працівника;</li> <li>• назва займаної посади;</li> <li>• рік вступу на роботу.</li> </ul> </li> <li>2. Розробити програму, яка виконує такі дії: <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу WORKER; записи мають бути розміщені за алфавітом. <ul style="list-style-type: none"> <li>• вивести на екран прізвищ працівників, чий стаж роботи в організації перевищує значення, введене з клавіатури;</li> <li>• якщо таких записів немає, вивести на екран відповідне повідомлення.</li> </ul> </li> </ul> </li> </ol>                       |
| 7. | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям TRAIN, яка містить наступні поля: <ul style="list-style-type: none"> <li>• назва пункту призначення;</li> <li>• номер поїзда;</li> <li>• час відправлення.</li> </ul> </li> <li>2. Розробити програму, яка виконує такі дії: <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу TRAIN; записи мають бути розміщені в алфавітному порядку за назвами пунктів призначення; <ul style="list-style-type: none"> <li>• вивести на екран інформації про поїзди, що відправляються після введеного з клавіатури часу;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul> </li> </ul> </li> </ol> |

|     |   |
|-----|---|
| 8.  | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям TRAIN, яка містить наступні поля: <ul style="list-style-type: none"> <li>• назва пункту призначення;</li> <li>• номер поїзда;</li> <li>• час відправлення.</li> </ul> </li> <li>2. Розробити програму, яка виконує такі дії: <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу TRAIN; записи мають бути впорядковані за часом відправлення поїзда; <ul style="list-style-type: none"> <li>• вивести на екран інформації про поїзди, що прямують до пункту, назва якого введено з клавіатури;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul> </li> </ul> </li> </ol> |
| 9.  | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям TRAIN, яка містить наступні поля: <ul style="list-style-type: none"> <li>• назва пункту призначення;</li> <li>• номер поїзда;</li> <li>• час відправлення.</li> </ul> </li> <li>2. Розробити програму, яка виконує такі дії: <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу TRAIN; записи мають бути впорядковані за номерами поїздів; <ul style="list-style-type: none"> <li>• вивести на екран інформації про потяг, номер якого введено з клавіатури;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul> </li> </ul> </li> </ol>                                  |
| 10. | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям MARSHRUTRUT, яка містить наступні поля: <ul style="list-style-type: none"> <li>• назва початкового пункту маршруту;</li> <li>• назва кінцевого пункту маршруту;</li> <li>• номер маршруту.</li> </ul> </li> <li>2. Розробити програму, яка виконує такі дії: <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу MARSHRUT; записи мають бути впорядковані за назвою кінцевого пункту маршруту; <ul style="list-style-type: none"> <li>• вивести на екран інформації про маршрут, номер якого введено з клавіатури;</li> </ul> </li> </ul> </li> </ol>  |

|     |  |
|-----|--|
|     | <ul style="list-style-type: none"> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul>  |
| 11. | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям MARSHRUT, яка містить наступні поля: <ul style="list-style-type: none"> <li>• назва початкового пункту маршруту;</li> <li>• назва кінцевого пункту маршруту;</li> <li>• номер маршруту.</li> </ul> </li> <li>2. Розробити програму, яка виконує такі дії: <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу MARSHRUT; записи мають бути впорядковані за номерами маршрутів; <ul style="list-style-type: none"> <li>• вивести на екран інформації про маршрути, які починаються або закінчуються в пункті, назва якого введено з клавіатури;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul> </li> </ul> </li> </ol> |
| 12. | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям NOTE, яка містить наступні поля: <ul style="list-style-type: none"> <li>• прізвище, ім'я;</li> <li>• номер телефону;</li> <li>• день народження (масив з трьох чисел).</li> </ul> </li> <li>2. Розробити програму, яка виконує такі дії: <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу NOTE; записи мають бути впорядковані по перших трьох цифрах номера телефону ; <ul style="list-style-type: none"> <li>• вивести на екран інформації про людей, чиї дні народження припадають на місяць, значення якого введено з клавіатури;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul> </li> </ul> </li> </ol>      |
| 13. | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям NOTE, яка містить наступні поля: <ul style="list-style-type: none"> <li>• прізвище, ім'я;</li> <li>• номер телефону;</li> <li>• день народження (масив з трьох чисел).</li> </ul> </li> <li>2. Розробити програму, яка виконує такі дії: <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу NOTE; записи мають бути впорядковані по</li> </ul> </li> </ol>   |

|     |  |
|-----|--|
|     | <p>прізвищу;</p> <ul style="list-style-type: none"> <li>• вивести на екран інформації про людей, чиї номери телефонів починаються з цифри, значення якої введено з клавіатури;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul>   |
| 14. | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям NOTE, яка містить наступні поля: <ul style="list-style-type: none"> <li>• прізвище, ім'я;</li> <li>• номер телефону;</li> <li>• день народження (масив з трьох чисел).</li> </ul> </li> <li>2. Розробити програму, яка виконує такі дії: <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу NOTE; записи мають бути впорядковані по року народження ; <ul style="list-style-type: none"> <li>• вивести на екран інформації про людей, чиї прізвища починаються з літери, значення якої введено з клавіатури;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul> </li> </ul> </li> </ol>   |
| 15. | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям MARSHRUT, яка містить наступні поля: <ul style="list-style-type: none"> <li>• назва початкового пункту маршруту;</li> <li>• назва кінцевого пункту маршруту;</li> <li>• номер маршруту.</li> </ul> </li> <li>2. Розробити програму, яка виконує такі дії: <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу MARSHRUT; записи мають бути впорядковані за назвою початкового пункту; <ul style="list-style-type: none"> <li>• вивести на екран інформації про номер маршруту, номер якого введено з клавіатури;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul> </li> </ul> </li> </ol> |
| 16. | <ol style="list-style-type: none"> <li>1. Створити структуру з ім'ям PIDPR, яка містить наступні поля: <ul style="list-style-type: none"> <li>• назва підприємства;</li> <li>• кількість співробітників;</li> <li>• дата заснування(масив з трьох чисел).</li> </ul> </li> </ol>   |

|     |  |
|-----|--|
|     | <p>2. Розробити програму, яка виконує такі дії:</p> <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу PIDPR; записи мають бути впорядковані за роком заснування;</li> <li>• вивести на екран інформації про підприємства, у яких кількість співробітників більша за кількість, введено з клавіатури;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul>   |
| 17. | <p>1. Створити структуру з ім'ям PIDPR, яка містить наступні поля:</p> <ul style="list-style-type: none"> <li>• назва підприємства;</li> <li>• кількість співробітників;</li> <li>• дата заснування(масив з трьох чисел).</li> </ul> <p>2. Розробити програму, яка виконує такі дії:</p> <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу NOTE; записи мають бути впорядковані по назві підприємства;</li> <li>• вивести на екран інформації про підприємства, чії дати заснування припадають на місяць, значення якого введено з клавіатури;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul> |
| 18. | <p>1. Створити структуру з ім'ям PIDPR, яка містить наступні поля:</p> <ul style="list-style-type: none"> <li>• назва підприємства;</li> <li>• кількість співробітників;</li> <li>• дата заснування(масив з трьох чисел).</li> </ul> <p>2. Розробити програму, яка виконує такі дії:</p> <ul style="list-style-type: none"> <li>• ввести з клавіатури дані в масив, що містить 4 елементи (структури) типу PIDPR; записи мають бути впорядковані за кількістю співробітників;</li> <li>• вивести на екран інформацію про підприємства, назва яких міститься введено з клавіатури слово;</li> <li>• вивести на екран відповідне повідомлення, якщо таких записів не знайдено.</li> </ul>                        |

## Лабораторна робота № 12

### Тема: Робота з текстовими файлами

(2 завдання, оцінюється в 5 балів)

#### Теоретичні відомості

У C++, робота з текстовими файлами зазвичай включає в себе використання об'єктів класу `fstream` або `ifstream` та `ofstream` для читання та запису даних.

#### **Відкриття та закриття файлів:**

Для відкриття файлу для читання використовується об'єкт класу `ifstream`, а для відкриття файлу для запису – `ofstream`.

```
#include <fstream>

std::ifstream inputFile("input.txt"); // Відкриття для читання
std::ofstream outputFile("output.txt"); // Відкриття для запису
```

Файли слід закривати після завершення роботи з ними:

```
inputFile.close(); // Закриття файла для читання
outputFile.close(); // Закриття файла для запису
```

#### **Читання та запис рядків**

Для читання та запису рядків використовуються методи `getline` та `<<`:

```
char line[20];
std::getline(inputFile, line); // Читання рядка
outputFile << line; // Запис рядка
```

Метод `eof` можна використовувати для перевірки наявності кінця файлу:

```
while (!inputFile.eof()) {
    // Операції читання
}
```

Але зазвичай краще використовувати цю умову в самому циклі читання.

При роботі з файлами важливо перевіряти, чи вони успішно відкрилися, та перевіряти стан після операцій читання/запису.

```
if (inputFile.is_open()) {
    // Робота з файлом
} else {
    std::cout << "Unable to open the file." << std::endl;
}
```

Робота з текстовими файлами в С++ дозволяє читати дані з файлів, записувати дані у файли, та виконувати інші операції з файлами. Користуючись різними методами та функціями, можна зручно опрацьовувати інформацію у текстових файлах.

**Завдання 1 (2 бали)**

|    |   |
|----|---|
| 1. | Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини, слова в яких розділені пробілами і різними розділовими знаками; б) зчитує файл text_1 і записує кожне слово в окремий рядок файлу text_2 (розділові знаки опускаються); в) зчитує файл text_2 і виводить його вміст по рядках.   |
| 2. | Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини; б) зчитує файл text_1, знаходить всі послідовності цифр і записує кожну із них в окремий рядок файлу text_2; в) зчитує файл text_2 і виводить його по рядках.  |
| 3. | Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини, слова в яких розділені пробілами і різними розділовими знаками (слова не перевищують довжину 16 символів); б) зчитує файл text_1, визначає скільки в ньому є слів із одного, двох, трьох, і т. д. символів, результат записує у файл text_2: у перший рядок слова і кількість слів із одного символу, у другий рядок слова і кількість слів із двох символів і т. д.; в) зчитує файл text_2 і виводить його по рядках. |
| 4. | Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини, слова в яких розділені пробілами і різними розділовими знаками; б) зчитує файл text_1, знаходить слова з подвоєнням букв і записує їх у файл text_2 по одному в рядок (якщо таких слів немає, то повідомлення про це записується у файл text_2); в) зчитує файл text_2 і виводить його вміст по рядках.  |
| 5. | Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини; б) зчитує файл text_1, формує рядки за правилом: якщо рядок файлу text_1 коротший ніж 20 символів, то він доповнюється знаками '&', а якщо довший, то лишні символи ігноруються, і записує їх у файл text_2; в) зчитує файл text_2 і виводить його по рядках.  |



|     |  |
|-----|--|
| 6.  | Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини, слова в яких розділені пробілами і різними розділовими знаками; б) зчитує файл text_1, знаходить слова, які розпочинаються голосною літерою і записує кожне в окремий рядок файлу text_2; в) зчитує файл text_2 і виводить його вміст по рядках.  |
| 7.  | Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини, слова в яких розділені пробілами і різними розділовими знаками; б) зчитує файл text_1, знаходить всі слова в яких є символ “а” і записує кожне з них в окремий рядок файлу text_2 (якщо таких слів немає, то повідомлення про це записується у файл text_2); в) зчитує файл text_2 і виводить його по рядках. |
| 8.  | Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини; б) зчитує файл text_1, замінює символ ”1” на “0” та навпаки і записує у файл text_2 по 15 символів у рядок (останній рядок може бути неповним); в) зчитує файл text_2 і виводить його по рядках.  |
| 9.  | Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини; б) зчитує файл text_1, пропускає цифри і формує рядки по 10 символів (останній рядок може бути неповним), вставляє перед кожним рядком порядковий номер (номер займає п’ять позицій і відділяється від тексту пробілом) і записує їх у файл text_2; в) зчитує файл text_2 і виводить його по рядках.          |
| 10. | Задано слово. Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини, слова в яких розділені пробілами і різними розділовими знаками; б) зчитує файл text_1, вилучає в кожному рядку всі входження заданого слова і записує їх у файл text_2; в) зчитує файл text_2 і виводить його по рядках.  |
| 11. | Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини, слова в яких розділені пробілами і різними розділовими знаками; б) зчитує файл text_1 і записує кожне слово в окремий рядок файлу text_2 (розділові знаки опускаються); в) зчитує файл text_2 і виводить його вміст по рядках.  |
| 12. | Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини; б) зчитує файл text_1, знаходить всі  |

|     |  |
|-----|--|
|     | <p>послідовності цифр і записує кожен із них в окремий рядок файлу text_2; в) зчитує файл text_2 і виводить його по рядках.</p>  |
| 13. | <p>Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини, слова в яких розділені пробілами і різними розділовими знаками (слова не перевищують довжину 16 символів); б) зчитує файл text_1, визначає скільки в ньому є слів із одного, двох, трьох, і т. д. символів, результат записує у файл text_2: у перший рядок слова і кількість слів із одного символу, у другий рядок слова і кількість слів із двох символів і т. д.; в) зчитує файл text_2 і виводить його по рядках.</p> |
| 14. | <p>Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини, слова в яких розділені пробілами і різними розділовими знаками; б) зчитує файл text_1, знаходить слова з подвоєнням букв і записує їх у файл text_2 по одному в рядок (якщо таких слів немає, то повідомлення про це записується у файл text_2); в) зчитує файл text_2 і виводить його вміст по рядках.</p>  |
| 15. | <p>Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини, слова в яких розділені пробілами і різними розділовими знаками; б) зчитує файл text_1 і записує кожне слово в окремий рядок файлу text_2 (розділові знаки опускаються); в) зчитує файл text_2 і виводить його вміст по рядках.</p>   |
| 16. | <p>Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини; б) зчитує файл text_1, знаходить всі послідовності цифр і записує кожен із них в окремий рядок файлу text_2; в) зчитує файл text_2 і виводить його по рядках.</p>  |
| 17. | <p>Написати програму, яка: а) створює .txt файл TF8_1 із символічних рядків довільної довжини; б) зчитує файл text_1, пропускає цифри і формує рядки по 10 символів (останній рядок може бути неповним), вставляє перед кожним рядком порядковий номер (номер займає п'ять позицій і відділяється від тексту пробілом) і записує їх у файл text_2; в) зчитує файл text_2 і виводить його по рядках.</p>  |
| 18. | <p>Написати програму, яка: а) створює .txt файл text_1 із символічних рядків довільної довжини, слова в яких розділені пробілами і різними розділовими знаками; б) зчитує файл text_1, знаходить слова, які</p>  |

|  |
|--|
| розпочинаються голосною літерою і записує кожне в окремий рядок файлу text_2; в) зчитує файл text_2 і виводить його вміст по рядках. |
|--|

### **Завдання 2 (3 бали)**

*Записати в файл інформацію про співробітників підприємства такої структури: табельний номер, прізвище та ініціали робітника, номер цеху, посада, адреса, заробітна плата, кількість дітей.*

*Розробити програму, яка дозволяє: а) створювати список записів співробітників підприємства, б) додавати в файл записи про нових співробітників у кінець списку, в) переглядати список співробітників, г) вносити зміни у запис про співробітника, д) обробляти певним чином інформацію про співробітників і виводити результат.*

*Всі зміни зберегти у файлі.*

|    |   |
|----|---|
| 1. | Для заданого номера цеху вивести за алфавітним порядком список співробітників і їх середню заробітну плату.               |
| 2. | Для заданого номера цеху вивести за алфавітним порядком список співробітників і фонд заробітної плати співробітників цеху |
| 3. | Для заданого прізвища співробітника знайти і вивести його посаду, адресу і заробітну плату.                               |
| 4. | Для заданого номера цеху вивести за алфавітним порядком список співробітників, які мають дітей.                           |
| 5. | За заданим табельним номером співробітника знайти і вивести: номер цеху, посаду і заробітну плату.                        |
| 6. | Для заданого номера цеху вивести за алфавітним порядком список співробітників, які отримують зарплату, більшу за середню. |
| 7. | Для заданого прізвища співробітника знайти і вивести номер цеху, адресу і заробітну плату.                                |
| 8. | Для заданої посади знайти і вивести прізвище співробітника і номер цеху,  |
| 9. | Для заданої посади знайти і вивести прізвище співробітника, адресу і заробітну плату.                                     |

*Записати в файл інформацію про власників автомобілів такої структури: марка автомобіля, номер, колір, рік випуску, прізвище та ініціали власника, адреса власника (вулиця, номер будинку).*

*Розробити програму, яка дозволяє: а) створити список записів власників автомобілів, б) додавати в файл записи про нових власників автомобілів у*

кінець списку, в) переглядати список власників автомобілів, г) вносити зміни у запис про власника автомобіля, д) обробляти певним чином інформацію про власників автомобілів і виводити результат.

*Всі зміни зберегти у файлі.*

|     |  |
|-----|--|
| 10. | Для заданої марки і кольору автомобіля, знайти і вивести за алфавітним порядком список власників.                                      |
| 11. | Знайти і вивести за алфавітним порядком список власників. автомобілів заданої марки, номер яких розпочинається цифрами '5', '5'.       |
| 12. | Знайти і вивести за алфавітним порядком список власників. автомобілів, які мешкають на заданій вулиці і мають задану марку автомобіля. |
| 13. | Для заданого кольору автомобіля, номер яких розпочинається цифрами '77', знайти і вивести список власників та їх адреси.               |
| 14. | Для заданого номера автомобіля, знайти і вивести прізвище власника та його адресу.   |
| 15. | Знайти і вивести за алфавітним порядком список власників автомобілів, заданого кольору.  |
| 16. | Для заданого кольору автомобіля, знайти і вивести список власників та їх адреси і рік випуску автомобілів.                             |
| 17. | Для заданого року випуску автомобіля, знайти і вивести прізвища власників та їх адреси.  |
| 18. | Для заданого кольору автомобіля, знайти і вивести прізвища власників та їх адреси.   |

## Лабораторна робота № 13

### Тема: Робота з STL

(2 завдання, оцінюється в 5 балів)

#### Теоретичні відомості

Стандартна бібліотека шаблонів (STL) є потужним інструментарієм в мові програмування C++. Вона надає набір шаблонів та класів для розв'язання різних задач, таких як робота з контейнерами, алгоритмами та ітераторами. Основні компоненти STL включають:

#### *Контейнери*

Контейнери в STL служать для зберігання та управління групами об'єктів. Деякі з основних контейнерів:

- `std::vector`: Динамічний масив.
- `std::list`: Направлений список.
- `std::deque`: Двонаправлений черговий контейнер.
- `std::queue`: Черга.
- `std::string`: Рядок.
- `std::stack`: Стек.
- `std::set`: Впорядкована множина унікальних елементів.
- `std::map`: Впорядкована колекція пар ключ-значення.
- `std::unordered_set`: Невпорядкована множина унікальних елементів.
- `std::unordered_map`: Невпорядкована колекція пар ключ-значення.

```
#include <vector>
#include <list>
#include <set>
#include <map>

std::vector<int> vec = {1, 2, 3, 4, 5};
std::list<char> myList = {'a', 'b', 'c'};
std::set<std::string> mySet = {"apple", "banana", "orange"};
std::map<int, std::string> myMap = {{1, "one"}, {2, "two"}, {3,
"three"}}};
```

Використання контейнерів дозволяє зручно та ефективно працювати з даними в програмі.

## *Алгоритми*

STL надає широкий набір алгоритмів для обробки даних, що знаходяться у контейнерах. Деякі з них:

- `std::sort`: Сортування діапазону.
- `std::find`: Пошук елемента у діапазоні.
- `std::for_each`: Застосування функції до кожного елемента у діапазоні.
- `std::transform`: Застосування операції до кожного елемента та збереження результату у новому діапазоні.

Алгоритми дозволяють виконувати різні операції над даними без прямого втручання в структуру контейнера.

## *Ітератори*

Ітератори є важливою частиною STL і дозволяють працювати з елементами контейнера. Деякі типи ітераторів:

Використання ітераторів дозволяє забезпечити універсальний доступ до елементів у контейнерах.

## *Клас string*

В C++, `std::string` є стандартним класом для роботи з рядками. Він реалізований в бібліотеці `<string>`. `std::string` забезпечує ряд методів та операцій для роботи з рядками, що робить його зручним для використання.

Створення рядка та його ініціалізація може відбуватися за допомогою різних конструкторів та операцій:

```
#include <string>

std::string str1; // Порожній рядок
std::string str2 = "Hello"; // Рядок із текстом "Hello"
std::string str3(5, 'a'); // Рядок із п'ятьма символами 'a'
```

`std::string` може бути введений і виведений за допомогою потокових операцій:

```
#include <iostream>

std::string inputString;
std::cin >> inputString; // Введення рядка
std::cout << inputString; // Виведення рядка
```

Рядки можуть бути конкатеновані за допомогою оператора +:

```
std::string firstName = "John";
std::string lastName = "Doe";
std::string fullName = firstName + " " + lastName; // "John Doe"
```

Метод length() або size() повертає кількість символів у рядку:

```
std::string myString = "Hello";
int length = myString.length(); // або myString.size()
```

Символ за певним індексом може бути отриманий за допомогою оператора []:

```
char firstChar = myString[0]; // Отримання першого символу
```

Рядки можуть бути порівнювані за допомогою операторів порівняння (^==, ^!=, ^<, ^>, ^<=, ^>=):

```
std::string str1 = "apple";
std::string str2 = "orange";

if (str1 == str2) {
// Рядки ідентичні
} else {
// Рядки різні
}
```

### ***Клас vector***

У C++, std::vector є динамічним масивом, який забезпечує автоматичне збільшення розміру при додаванні нових елементів.

std::vector можна створити та ініціалізувати за допомогою різних конструкторів:

```
#include <vector>

// Порожній вектор
std::vector<int> myVector;
// Вектор із 5 елементами, кожен із яких 3.14
std::vector<double> anotherVector(5, 3.14);
```

Елементи можна додавати в кінець вектора за допомогою методу `push_back()`:

```
myVector.push_back(42); // Додати 42 в кінець вектора
```

Елементи можна видаляти з кінця вектора за допомогою методу `pop_back()`:

```
myVector.pop_back(); // Видалити елемент з кінця вектора
```

Елементи можна отримувати за допомогою оператора `[]` або методу `at()`:

```
int value = myVector[2]; // Отримати третій елемент вектора  
int anotherValue = myVector.at(2); // Отримати третій елемент вектора
```

Метод `size()` повертає кількість елементів у векторі:

```
size_t size = myVector.size();
```

Вектор автоматично збільшує свій розмір при додаванні нових елементів, і його поточна ємність може бути отримана за допомогою методу `capacity()`:

```
size_t capacity = myVector.capacity();
```

`std::vector` може використовувати ітератори для ітерації через свої елементи:

```
for (std::vector<int>::iterator it = myVector.begin(); it !=  
myVector.end(); ++it) {  
    // Доступ до елемента: *it  
}
```

Або використовуючи цикл `for` за допомогою діапазонного ітератора:

```
for (const auto& element : myVector) {  
    // Доступ до елемента: element  
}
```

`std::vector` підтримує операції копіювання та об'єднання з іншими векторами:

```
std::vector<int> anotherVector = {1, 2, 3};  
myVector = anotherVector; // Копіювання вектора  
  
myVector.insert(myVector.end(), anotherVector.begin(),  
anotherVector.end()); // Об'єднання векторів
```



std::vector дуже корисний для роботи з динамічними колекціями даних, оскільки він автоматично розширюється при необхідності і надає зручний доступ до елементів.

### Завдання 1 (2.5 бали)

Використовуючи контейнерний клас <vector> та ітератори для забезпечення доступу до елементів колекції. За необхідності використовуйте алгоритми, які надає бібліотека STL.

|    |   |
|----|---|
| 1. | Задано два масиви цілих чисел $A[n]$ , $n \leq 500$ і $B[m]$ , $m \leq 200$ . Обчислити суму об'єднання $A \cup B$ . Об'єднання $A \cup B$ – це множина елементів масиву $A$ і $B$ , взятих по одному разу. Вивести елементи об'єднання та їх суму.   |
| 2. | Задано масив дійсних чисел $A[n]$ , $n \leq 400$ . Написати програму, яка визначає кількість додатних елементів, що містяться між його максимальним та мінімальним елементом.   |
| 3. | Задано масив дійсних чисел $A[n*m]$ , $n \leq 100$ , $m \leq 10$ . Розробити програму, яка обчислює<br>$Z = \max(\min(A[0], \dots, A[m-1]), \min(A[m], \dots, A[2m-1]), \dots, \min(A[m*(n-1)], \dots, A[n*m-1])).$ <p>У масиві <math>n</math> груп по <math>m</math> чисел. Максимальне значення шукається серед мінімальних значень елементів у кожній групі.</p> |
| 4. | Задано дійсні величини $a$ , $b$ ( $a < b$ ) і масив чисел $X[n]$ , $n < 200$ . Розробити програму, яка обчислює суму всіх $X[i] < a$ , добуток всіх $X[i] > b$ і знаходить $\max X[i]$ та $\min X[i]$ серед $X[i] \in (a, b)$ , $i = 0, 1, 2, \dots, n-1$ . Якщо чисел $X[i] < a$ , або $X[i] > b$ , або $X[i] \in (a, b)$ немає, то виводить повідомлення про це. |
| 5. | Задано масив чисел $A[n]$ , $n \leq 300$ . Написати програму, яка заміняє всі від'ємні елементи на середнє арифметичне елементів масиву   |
| 6. | Задано масив дійсних чисел $A[n]$ , $n \leq 200$ . Розробити програму, яка записує в масив $B[n]$ по одному разу всі повторювані числа в масиві $A[n]$ . Виводить повідомлення, якщо таких чисел немає.   |
| 7. | Задано масив цілих чисел $A[2n]$ , $n \leq 200$ . Розробити програму, яка добуток тих чисел із $A[n], \dots, A[2n-1]$ , які менші за середнє арифметичне значення чисел $A[0], \dots, A[n-1]$ . Якщо таких чисел немає, то виводить повідомлення про це.  |
| 8. | Дано два масиви цілих чисел $X[n]$ і $Y[n]$ , $n \leq 300$ . Розробити програму, яка обчислює суму елементів об'єднання цих масивів. Об'єднанням двох масивів є взяті по одному разу всі елементи масиву $X[n]$ і $Y[n]$ . Вивести масив об'єднання та їх суму.   |

|     |  |
|-----|--|
| 9.  | Дано масив цілих чисел $A[n]$ , $n \leq 300$ . Розробити програму, яка обчислює суму елементів масиву, якщо він упорядкований за зростанням або добуток, якщо не упорядкований. Масив вважається упорядкованим за зростанням якщо $A[i] \leq A[i+1]$ для $i = 0, 1, 2, \dots, n - 2$ .   |
| 10. | Задано масив цілих чисел $A[n]$ , $n \leq 400$ . Розробити програму, яка знаходить мінімальне серед тих чисел, які не повторюються. Якщо таких чисел немає, то виводить повідомлення про це.   |
| 11. | Задано масив дійсних чисел $A[n]$ , $n \leq 500$ . Розробити програму, яка обчислює добуток всіх чисел, які повторюються, і суму всіх чисел, які не повторюються. Якщо чисел що повторюються або не повторюються немає, то виводить відповідне повідомлення.   |
| 12. | Задано масив дійсних чисел $A[n * m]$ , $n \leq 300$ і $m \leq 25$ . Розробити програму, яка обчислює<br>$V = \min(\max(A[0], \dots, A[m-1]), \max(A[m], \dots, A[2m-1]), \dots, \max(A[m*(n-1)], \dots, A[n*m-1]))$ <p>У масиві <math>n</math> груп по <math>m</math> чисел. Мінімальне значення шукається серед максимальних значень елементів у кожній групі.</p> |
| 13. | Задано масив чисел $A[2n]$ , $n \leq 200$ . Написати програму, яка обчислює значення:<br>$Z = \max(\min(A[0] \dots A[2n-1]), \min(A[1] \dots A[2n-2]), \dots, \min(A[n-1] \dots A[n]))$  |
| 14. | Задано дійсних масив чисел $A[n]$ , $n \leq 500$ . Написати програму, яка обчислює добуток всіх чисел, які знаходяться між першим і останнім від'ємними елементами цього масиву і виводить цей діапазон. Якщо від'ємних чисел немає або є тільки одне, або вони стоять поряд, то виводить відповідне повідомлення.   |
| 15. | Задано два масиви цілих чисел $A[n]$ , $n \leq 500$ і $B[m]$ , $m \leq 200$ . Обчислити суму об'єднання $A \cup B$ . Об'єднання $A \cup B$ – це множина елементів масиву $A$ і $B$ , взятих по одному разу. Вивести елементи об'єднання та їх суму.  |
| 16. | Дано два масиви цілих чисел $X[n]$ і $Y[n]$ , $n \leq 300$ . Розробити програму, яка обчислює суму елементів об'єднання цих масивів. Об'єднанням двох масивів є взяті по одному разу всі елементи масиву $X[n]$ і $Y[n]$ . Вивести масив об'єднання та їх суму.  |
| 17. | Задано масив дійсних чисел $A[n]$ , $n \leq 100$ . Написати програму, яка обчислює добуток елементів цього масиву, якщо масив упорядкований за зростанням або суму, якщо масив не упорядкований. Масив буде упорядкованим за зростанням якщо якщо $A[i] \leq A[i+1]$ для $i = 0, 1, 2, \dots, n - 2$ .   |
| 18. | Задано масив цілих чисел $A[n]$ , $n \leq 400$ . Розробити програму, яка   |

|  |   |
|--|---|
|  | знаходить мінімальне серед тих чисел, які не повторюються. Якщо таких чисел немає, то виводить повідомлення про це. |
|--|---|

### Завдання 2 (2.5 бали)

*Використовуючи контейнерний клас `<string>` та ітератори для забезпечення доступу до елементів колекції. За необхідності використайте алгоритми, які надає бібліотека `STL`.*

|     |   |
|-----|---|
| 1.  | Ввести з клавіатури текст. Вивести текст зі слів, що починаються з заданого символу, попередньо видалити зі слів цей символ крім першого.   |
| 2.  | У рядку, що складається з груп нулів і одиниць, знайти і вивести на екран групи з парною кількістю символів.  |
| 3.  | Ввести з клавіатури текст, слова в якому розділені пробілами і різними розділовими знаками. Створити програму, яка вилучає з початкового тексту всі повторні входження слів.  |
| 4.  | Ввести з клавіатури текст, слова в якому розділені пробілами і різними розділовими знаками. Створити програму, яка вилучає з початкового тексту всі слова з подвоєнням літер і записує їх в окремий рядок, розділяючи пробілами. Друкує окремо вилучені слова і текст, що залишився після вилучення слів. |
| 5.  | Ввести з клавіатури текст, слова в якому розділені пробілами і різними розділовими знаками. Створити програму, яка вилучає з початкового тексту всі слова найбільшої довжини. (Слів найбільшої довжини може бути декілька). Друкує текст, що залишився після вилучення слів.                              |
| 6.  | Ввести з клавіатури текст, слова в якому розділені пробілами і різними розділовими знаками. Створити програму, яка знаходить і друкує всі симетричні слова (наприклад, слово абввба є симетричним).   |
| 7.  | Ввести з клавіатури текст, слова в якому розділені пробілами і різними розділовими знаками. Створити програму, яка вилучає в кожному слові початкового тексту всі наступні входження першої літери.   |
| 8.  | Ввести з клавіатури текст, слова в якому розділені пробілами і різними розділовими знаками. Створити програму, яка вилучає в кожному слові початкового тексту всі попередні входження останньої літери.   |
| 9.  | Ввести з клавіатури текст, слова в якому розділені пробілами і різними розділовими знаками. Створити програму, яка вилучає з початкового тексту всі повторні входження слів.  |
| 10. | Ввести з клавіатури текст, слова в якому розділені пробілами і різними  |

|     |   |
|-----|---|
|     | розділовими знаками. Створити програму, яка знаходить і вилучає всі слова, що входять в цей текст по одному разу.   |
| 11. | Ввести з клавіатури текст, слова в якому розділені пробілами і різними розділовими знаками. Створити програму, яка в словах непарної довжини початкового тексту вилучає середню літеру.                 |
| 12. | Ввести з клавіатури текст, слова в якому розділені пробілами і різними розділовими знаками. Створити програму, яка знаходить і друкує всі слова, що входять у заданий текст по одному разу.             |
| 13. | Ввести з клавіатури два тексти, слова в яких розділені пробілами і різними розділовими знаками. Створити програму, яка вилучає із другого тексту всі входження слів першого тексту.                     |
| 14. | Ввести з клавіатури два тексти, слова в яких розділені пробілами і різними розділовими знаками. Створити програму, яка створює третій текст із слів першого тексту, що не входять у другий.             |
| 15. | Ввести з клавіатури два тексти, слова в яких розділені пробілами і різними розділовими знаками. Створити програму, яка вилучає із другого тексту всі входження слів першого тексту.                     |
| 16. | Ввести з клавіатури текст, слова в якому розділені пробілами і різними розділовими знаками. Створити програму, яка знаходить і друкує всі симетричні слова (наприклад, слово абввба є симетричним).     |
| 17. | Ввести з клавіатури текст, слова в якому розділені пробілами і різними розділовими знаками. Створити програму, яка вилучає в кожному слові початкового тексту всі наступні входження першої літери.     |
| 18. | Ввести з клавіатури текст, слова в якому розділені пробілами і різними розділовими знаками. Створити програму, яка вилучає в кожному слові початкового тексту всі попередні входження останньої літери. |