

**ПРИВАТНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«КОМП'ЮТЕРНА АКАДЕМІЯ ШАГ»**



Методичні вказівки

ЛАБОРАТОРНІ РОБОТИ

з дисципліни «Вступ до програмування ч.1 (C/C++)»

галузь знань	<u>12 Інформаційні технології</u>
спеціальність	<u>122 Комп'ютерні науки</u>

Львів - 2017

Методичні вказівки з лабораторних робіт з навчальної дисципліни «**Вступ до програмування ч.1 (C/C++)**» для студентів за спеціальністю 122 Комп'ютерні науки

Розробник:

професор, д.т.н., професор кафедри _____

ЗМІСТ

Лабораторна робота 1. Програмування розгалужених та циклічних процесів. .	5
1.1. Мета роботи	5
1.2. Теоретичний матеріал.....	5
1.3. Порядок виконання роботи	6
1.3.1. Блок завдань 1	6
1.3.2. Блок завдань 2	7
1.4. Зміст звіту	8
Лабораторна робота 2. Масиви даних та використання вказівників для роботи з масивами.	9
2.1. Мета роботи	9
2.2. Теоретичний матеріал.....	9
2.3. Порядок виконання роботи	10
2.3.1. Блок завдань 1	10
2.3.2. Блок завдань 2	13
2.4. Зміст звіту	16
Лабораторна робота 3. Технології програмування структур та розміщення структур у пам'яті.	17
3.1. Мета роботи	17
3.2. Теоретичний матеріал.....	17
3.3. Порядок виконання роботи	17
3.4. Зміст звіту	23
Лабораторна робота 4. Технології програмування функцій та їх використання.	24
4.1. Мета роботи	24
4.2. Теоретичний матеріал.....	24
4.3. Порядок виконання роботи	25
4.4. Зміст звіту	33
Лабораторна робота 5. Перевантаження та шаблони функцій	34
5.1. Мета роботи	34
5.2. Теоретичний матеріал.....	34
5.3. Порядок виконання роботи	34
5.4. Зміст звіту	34

Лабораторна робота 6. Засоби мови C для опрацювання стрінгів та файлів. ..	36
6.1. Мета роботи	36
6.2. Теоретичний матеріал.....	36
6.3. Порядок виконання роботи	36
6.3.1. Блок завдань 1	36
6.3.2. Блок завдань 2	38
6.4. Зміст звіту	43
Лабораторна робота 7. Динамічні структури даних.....	44
7.1. Мета роботи	44
7.2. Теоретичний матеріал.....	44
7.3. Порядок виконання роботи	44
7.4. Зміст звіту	51
Лабораторна робота 8. Використання чисельних бібліотек мови C/C++ для вирішення практичних задач.....	52
8.1. Мета роботи	52
8.2. Теоретичний матеріал.....	52
8.3. Порядок виконання роботи	52
8.3.1. Блок завдань 1	52
8.3.2. Блок завдань 2	54
8.4. Зміст звіту	55
Література.....	56

Лабораторна робота 1. Програмування розгалужених та циклічних процесів.

1.1. Мета роботи

Освоєння технології програмування розгалужених та циклічних процесів мови C/C++ для вирішення практичних задач. Надбання навичок розробки консольних додатків та оптимізації програмного коду.

1.2. Теоретичний матеріал

Необхідно повторити матеріали лекцій 1-3 [13].

Виділемо такі ключові моменти, що стосуються технологій програмування розгалужених та циклічних процесів, які є важливими при вирішенні практичних задач:

1. Щоб отримати максимальну читабельність і простоту структури програми, треба правильно вибрати спосіб реалізації розгалужень (за допомогою, if, switch або умовних операцій), а також найбільш підходящий оператор цикл.
2. Перевірка дійсних величин на рівність небезпечна.
3. Вираз, що стоїть в дужках після ключового слова switch, і сталі вирази в case повинні бути цілочисельного типу. Після кожної гілки для передачі управління на кінець оператора switch використовується оператор break.
4. Рекомендується завжди описувати в операторі switch гілку default.
5. Щоб уникнути помилок при програмуванні циклів, рекомендується:
 - поміщати в блок (фігурні дужки) тіло циклів, якщо в них потрібно виконати більше одного оператора;
 - перевіряти, чи всім змінним, що зустрічаються в правій частині операторів присвоювання в тілі циклу, присвоєні до цього початкові значення, а також чи можливо виконання інших операторів;
 - перевіряти, чи змінюються в циклі хоча б одна змінна, що входить в умову виходу з циклу;
 - якщо кількість повторень циклу заздалегідь не відомо, передбачати аварійний вихід з циклу після досягнення деякого досить великої кількості ітерацій.
6. Оператори циклу в мові C++ взаємозамінні, але можна навести деякі рекомендації по вибору найкращого в кожному конкретному випадку:
 - Оператор do while використовують, коли цикл потрібно обов'язково виконати хоча б один раз - наприклад, при введенні даних.
 - Оператор while зручніше в тих випадках, коли або число ітерацій заздалегідь невідомо, яких очевидних параметрів циклу немає, або модифікацію параметрів зручніше записувати десь в довільному місці тіла циклу.

- Оператор for краще в більшості інших випадків. Однозначно - для організації циклів з лічильниками, тобто з цілочисельними змінними, які змінюють своє значення при кожному проході циклу регулярним чином (наприклад, збільшуються на 1).

1.3. Порядок виконання роботи

Необхідно розв'язати по одному завданню з кожного блоку, які будуть наведені нижче. Варіант завдання обрати за номером журналу. Результати у вигляді програмного коду та результатів виконання програмного коду занести в звіт.

1.3.1. Блок завдань 1

Обчислити і вивести на екран у вигляді таблиці значення функції, заданої за допомогою ряду Тейлора, на інтервалі від $x_{\text{нач}}$ до $x_{\text{кін}}$ з кроком dx з точністю ε . Таблицю забезпечити заголовком і шапкою. Кожен рядок повинен містити значення аргументу, значення функції та кількість підсумувати членів ряду.

$$1. \quad x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} - \dots$$

$$2. \quad x - \frac{x^2}{\sqrt{2}} + \frac{x^3}{\sqrt{3}} - \frac{x^4}{2} + \frac{x^5}{\sqrt{5}} - \dots$$

$$3. \quad 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \frac{x^5}{5!} + \dots$$

$$4. \quad 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

$$5. \quad 1 - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

$$6. \quad x - \frac{1}{2} \cdot \frac{x^3}{3} + \frac{1 \cdot 3}{2 \cdot 4} \cdot \frac{x^5}{5} - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot \frac{x^7}{7} + \dots$$

$$7. \quad 1 - 2x^2 + 3x^3 - 4x^4 + 5x^5 - 6x^6 + \dots$$

$$8. \quad x - \frac{x^2}{2^2} + \frac{x^3}{3^2} - \frac{x^4}{4^2} + \frac{x^5}{5^2} - \frac{x^6}{6^2} + \dots$$

$$9. \quad 1 - 2x + \frac{3x^2}{2!} - \frac{4x^3}{3!} + \frac{5x^4}{4!} - \frac{6x^5}{5!} + \dots$$

$$10. \quad \frac{x}{1 \cdot 2^2} - \frac{x^2}{2 \cdot 3^2} + \frac{x^3}{3 \cdot 4^2} - \frac{x^4}{4 \cdot 5^2} + \frac{x^5}{5 \cdot 6^2} - \dots$$

$$11. \quad 1 - \frac{1}{2}x^2 + \frac{1 \cdot 3}{2 \cdot 4}x^4 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}x^6 + \dots$$

$$12. \quad x - \frac{2^2 x^2}{5} + \frac{2^3 x^3}{10} - \frac{2^4 x^4}{17} + \frac{2^5 x^5}{26} - \dots$$

$$13. \quad \frac{x}{2 \cdot 1} - \frac{x^2}{2^2 \cdot 2} + \frac{x^3}{2^3 \cdot 3} - \frac{x^4}{2^4 \cdot 4} + \frac{x^5}{2^5 \cdot 5} - \dots$$

$$14. \quad \frac{1}{3} - \frac{2x}{2 \cdot 3^2} + \frac{3x^2}{2^2 \cdot 3^3} - \frac{4x^3}{2^3 \cdot 3^4} + \frac{5x^4}{2^4 \cdot 3^5} - \dots$$

$$15. \frac{x}{2 \cdot 1} - \frac{2x^2}{2 \cdot 2 \cdot 3} + \frac{x^3}{2^2 \cdot 3 \cdot 4} - \frac{4x^4}{2^3 \cdot 4 \cdot 5} + \dots$$

$$16. 1 - 4x + 9x^2 - 16x^3 + 25x^4 - 36x^5 + \dots$$

$$17. \frac{2x^2}{3} - \frac{4x^3}{8} + \frac{8x^4}{15} - \frac{16x^5}{24} + \dots$$

$$18. \frac{1}{3^{\frac{1}{2}}} - \frac{x}{3^{\frac{1}{3}}} + \frac{x^2}{3^{\frac{1}{4}}} - \frac{x^3}{3^{\frac{1}{5}}} + \dots$$

$$19. x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \dots$$

$$20. 1 - \frac{x^2}{2} + \frac{x^4}{4} - \frac{x^6}{6} + \frac{x^8}{8} - \dots$$

1.3.2. Блок завдань 2

1. Написати програму, яка введе сім значень та порахує кількість додатних чисел, що діляться на 3 без залишку.

2. Написати програму, яка введе вісім значень та порахує кількість від'ємних чисел, або чисел, рівних нулю.

3. Написати програму, яка введе 5 значень та порахує суму чисел, що діляться без залишку на 7 або 3.

4. Написати програму, яка введе 5 значень змінної n та виведе ряд чисел $n, n+3, n+6, n+9, \dots, n+5 \cdot 3$.

5. Написати програму, яка введе 6 значень та порахує суму чисел, що більші за 4 та менші, або дорівнюють 7.

6. Написати програму, яка введе 7 значень та порахує кількість чисел, що не рівні 5 і менші 7.

7. Написати програму, яка введе 5 значень та порахує добуток чисел, які діляться без залишку на 3 і не діляться без залишку на 5.

8. Написати програму, яка введе 6 значень та порахує суму чисел, які є квадратами цілих чисел (квадратний корінь з яких – ціле число).

9. Написати програму, яка введе 7 значень та порахує кількість чисел, які при діленні на 7 дають залишок 1 або 2.

10. Написати програму, яка введе 5 значень та порахує суму квадратів тих чисел, які менші 1 та більші 0.

11. Скласти програму зведення натурального числа n у квадрат, враховуючи таку закономірність:

$$1^2 = 1,$$

$$2^2 = 1 + 3,$$

$$3^2 = 1 + 3 + 5,$$

$$4^2 = 1 + 3 + 5 + 7,$$

$$n^2 = 1 + 3 + 5 + 7 + 9 + \dots + 2n-1.$$

Не використовувати операцію множення!

12. Знайти суму $1^2 + 2^2 + 3^2 + \dots + 10^2$. Операцію зведення в ступінь не використовувати, врахувати особливості отримання квадрата натурального числа, що відмічені у попередній задачі.

13. Дано два натуральних числа a й n , обчислити $\sqrt{a + \sqrt{a + \sqrt{a + \dots + \sqrt{a}}}}$ (n доданків).

14. Почавши тренування, лижник у перший день пробіг 10 км. Кожний наступний день він збільшував пробіг на 10% від пробігу попереднього дня. Визначити:

- а) пробіг лижника за другий, третій, ..., десятий день тренувань;
- б) який сумарний шлях він пробіг за перші 7 днів тренувань.

15. Дано натуральне число n . Обчислити $1^1 - 2^2 + 3^3 - \dots \pm n^n$.

16. Підприємець 1 березня відкрив рахунок у банці, вклав 1000 руб. Через кожний місяць вклад збільшується на 2% від попередньої суми. Визначити:

- а) приріст суми вкладу за перший, другий, ..., десятий місяць;
- б) суму вкладу через три, чотири, ..., дванадцять місяців.

17. Дано натуральне число n , обчислити $\sqrt{3 + \sqrt{6 + \sqrt{3 + \dots + \sqrt{3}}}}$.

18. Однокліткова амеба кожні 3 години ділиться на 2 клітини. Визначити, скільки клітин буде через 3, 6, 9, ..., 24 години, якщо спочатку була одна амеба.

19. Скласти програму для графічного зображення подільності чисел від 1 до n (значення n вводиться з клавіатури). У кожному рядку необхідно надрукувати чергове число й стільки символів «+», скільки дільників у цього числа.

20. Натуральне число називається досконалим, якщо воно дорівнює сумі своїх дільників, включаючи 1 і, звичайно, виключаючи саме це число. Наприклад, число 6 – досконале ($6 = 1 + 2 + 3$). Дано натуральне число. З'ясувати, чи є воно досконалим.

1.4. Зміст звіту

Звіт повинен містити:

1. Титульний аркуш.
2. Лістинги програм, що вирішують завдання.
3. При необхідності навести екрані форми, або консольний виведення результатів роботи програми.
4. Розширені висновки з лабораторної роботи.

Лабораторна робота 2. Масиви даних та використання вказівників для роботи з масивами.

2.1. Мета роботи

Вивчити технології обробки масивів даних, операції над ними, алгоритми сортування. Навчитися застосовуватися вказівники для роботи з масивами.

2.2. Теоретичний матеріал

Необхідно вивчити матеріали лекцій 10 [13].

Виділемо такі ключові моменти, що стосуються технологій програмування одновимірних масивів та вказівників, які є ключовими при вирішенні практичних задач:

1. Розмірність нединамічного масиву може бути тільки константою або константним виразом. Рекомендується задавати розмірність за допомогою іменованої константи.
2. Елементи масивів нумеруються з нуля, тому максимальний номер елемента завжди на одиницю менше розмірності.
3. Автоматичний контроль виходу індексу за межі масиву не проводиться, тому програміст повинен стежити за цим самостійно.
4. Вказівник - це змінна, в якій зберігається адреса області пам'яті. Ім'я масиву є показником на його нульовий елемент.
5. Обнулення динамічної пам'яті при її виділенні не відбувається. Ініціалізувати динамічний масив не можна.
6. Звільнення пам'яті, виділеної за допомогою `new[]`, виконується за допомогою операції `delete[]`.
7. Перед виходом локального показника з області його дії необхідно звільняти пов'язану з ним динамічну пам'ять.
8. Якщо кількість елементів, які повинні бути введені в програму, відомо до її виконання, визначайте масив в операторі опису змінних (причому краще як локальну змінну, ніж як глобальну); якщо кількість можна задати під час виконання програми, але до введення елементів, створюйте динамічний масив; якщо немає, використовуйте лінійний список або іншу динамічну структуру.
9. Алгоритми сортування масивів розрізняються по швидкодії, займаній пам'яті і області застосування.

Двовимірні масиви:

1. У масивах, визначених за допомогою операторів опису, обидві розмірності повинні бути константами або константними виразами. Масив зберігається по рядках в безперервній області пам'яті. При описі масиву можна задати початкові значення його елементів.

2. Перший індекс завжди є номер рядка, другий - номер стовпця. Кожен індекс може змінюватися від 0 до значення відповідної розмірності, зменшеної на одиницю.

3. При виділенні динамічної пам'яті під масив цілком найлівішу розмірність можна задати за допомогою змінної, всі інші розмірності повинні бути константами. Для двовимірного масиву це означає, що ліва розмірність може бути змінної, а друга - тільки константою.

4. Для виділення динамічної пам'яті під масив, в якому все розмірності змінні, використовуються цикли.

5. Звільнення пам'яті з-під масиву з будь-якою кількістю вимірювань виконується за допомогою операції delete [].

2.3. Порядок виконання роботи

Обрати по одному варіанту завдань з кожного блоку і провести реалізацію на мові C++, результати занести в звіт. Варіант обрати відповідно до порядкового номера по журналу.

2.3.1. Блок завдань 1

Початкові дані для всіх варіантів - n дійсних величин. При написанні програм можна використовувати як динамічні, так і нединамічні масиви. Розмірність останніх задавати іменованою константою.

Варіант 1

1. Знайти суму від'ємних елементів масива.
2. Знайти добуток елементів масиву, що розташовані між максимальним та мінімальним елементами.
3. Упорядкувати елементи масиву по зростанню.

Варіант 2

1. Знайти суму додатних елементів масива.
2. Знайти добуток елементів масива, що розташовані між максимальним по модулю і мінімальним по модулю елементами.
3. Впорядкувати елементи масиву по спаданню.

Варіант 3

1. Знайти добуток елементів масиву з парними номерами.
2. Знайти суму елементів масиву, що розташовані між першим і останнім нульовими елементами.
3. Перетворити масив таким чином, щоб спочатку розташовувалися всі додатні елементи, а потім - все від'ємні (елементи, рівні 0, вважати додатними).

Варіант 4

1. Знайти суму елементів масиву з непарними номерами.

2. Знайти суму елементів масиву, що розташовані між першим і останнім від'ємними елементами.

3. Стиснути масив, видаливши з нього всі елементи, модуль яких не перевищує 1. Вивільнені в кінці масиву елементи заповнити нулями.

Варіант 5

1. Знайти максимальний елемент масиву.

2. Знайти суму елементів масиву, що розташовані до останнього додатного елемента.

3. Стиснути масив, видаливши з нього всі елементи, модуль яких знаходиться в інтервалі $[a, b]$. Вивільнені в кінці масиву елементи заповнити нулями.

Варіант 6

1. Знайти мінімальний елемент масиву.

2. Знайти суму елементів масиву, що розташовані між першим і останнім додатним елементами.

3. Перетворити масив таким чином, щоб спочатку розташовувалися всі елементи, рівні нулю, а потім - всі інші.

Варіант 7

1. Знайти номер максимального елемента масиву.

2. Знайти добуток елементів масиву, що розташовані між першим і другим нульовими елементами.

3. Перетворити масив таким чином, щоб в першій його половині розташовувалися елементи, що стояли в непарних позиціях, а в другій половині - елементи, що стояли в парних позиціях.

Варіант 8

1. Знайти номер мінімального елемента масиву.

2. Знайти суму елементів масиву, розташованих між першим і другим від'ємними елементами.

3. Перетворити масив таким чином, щоб спочатку розташовувалися всі елементи, модуль яких не перевищує 1, а потім - всі інші.

Варіант 9

1. Знайти максимальний по модулю елемент масиву.

2. Знайти суму елементів масиву, розташованих між першим і другим додатними елементами.

3. Перетворити масив таким чином, щоб елементи, рівні нулю, розташовувалися після всіх інших.

Варіант 10

1. Знайти мінімальний по модулю елемент масиву.

2. Знайти суму модулів елементів масиву, розташованих після першого елемента, рівного нулю.

3. Перетворити масив таким чином, щоб в першій його половині розташовувалися елементи, що стояли в парних позиціях, а в другій половині - елементи, що стояли в непарних позиціях.

Варіант 11

1. Знайти номер мінімального за модулем елемента масиву.

2. Знайти суму модулів елементів масиву, розташованих після першого від'ємного елемента.

3. Стиснути масив, видаливши з нього всі елементи, величина яких знаходиться в інтервалі $[a, b]$. Вивільнені в кінці масиву елементи заповнити нулями.

Варіант 12

1. Знайти номер максимального по модулю елемента масиву.

2. Знайти суму елементів масиву, розташованих після першого додатного елемента.

3. Перетворити масив таким чином, щоб спочатку розташовувалися всі елементи, ціла частина яких лежить в інтервалі $[a, b]$, а потім - всі інші.

Варіант 13

1. Знайти кількість елементів масиву, що лежать в діапазоні від A до B .

2. Знайти суму елементів масиву, розташованих після максимального елемента.

3. Впорядкувати елементи масиву по спадючій модулів елементів.

Варіант 14

1. Знайти кількість елементів масиву, рівних 0.

2. Знайти суму елементів масиву, розташованих після мінімального елемента.

3. Впорядкувати елементи масиву по зростанню модулів елементів.

Варіант 15

1. Знайти кількість елементів масиву, що є більше за значення C .

2. Знайти добуток елементів масиву, розташованих після максимального за модулем елемента.

3. Перетворити масив таким чином, щоб спочатку розташовувалися всі від'ємні елементи, а потім - все додатні (елементи, рівні 0, вважати додатними).

Варіант 16

1. Знайти кількість від'ємних елементів масиву.

2. Знайти суму модулів елементів масиву, розташованих після мінімального за модулем елемента.

3. Замінити всі від'ємні елементи масиву їх квадратами і впорядкувати елементи масиву по зростанню.

Варіант 17

1. Знайти кількість додатних елементів масиву.
2. Знайти суму елементів масиву, розташованих після останнього елемента, рівного нулю.
3. Перетворити масив таким чином, щоб спочатку розташовувалися всі елементи, ціла частина яких не перевищує 1, а потім - всі інші.

Варіант 18

1. Знайти кількість елементів масиву, що є менше за значення C.
2. Знайти суму цілих частин елементів масиву, розташованих після останнього від'ємного елемента.
3. Перетворити масив таким чином, щоб спочатку розташовувалися всі елементи, що відрізняються від максимального не більше ніж на 20%, а потім - всі інші.

Варіант 19

1. Знайти добуток від'ємних елементів масиву.
2. Знайти суму додатних елементів масиву, розташованих до максимального елемента.
3. Змінити порядок проходження елементів в масиві на зворотний.

Варіант 20

1. Знайти добуток додатних елементів масиву.
2. Знайти суму елементів масиву, розташованих до мінімального елемента.
3. Відсортувати по зростанню окремо елементи, що стоять на парних місцях, і елементи, що стоять на непарних місцях.

2.3.2. Блок завдань 2

Рекомендується виконувати кожне завдання в двох варіантах: використовуючи локальні і динамічні масиви. Розмірності локальних масивів задавати іменованими константами, значення елементів масиву - в списку ініціалізації. Введення даних в динамічний масив виконувати з файлу.

Варіант 1

Дана цілочислена прямокутна матриця. Визначити:

- 1) кількість рядків, які містять жодного нульового елемента;
- 2) максимальне з чисел, що зустрічається в заданій матриці більше одного разу.

Варіант 2

Дана цілочисельна прямокутна матриця. Визначити кількість стовпців, що не містять жодного нульового елемента.

Характеристикою рядка цілочисленної матриці назвемо суму її додатних парних елементів. Переставляючи рядки заданої матриці, розташовувати їх відповідно до зростання характеристик.

Варіант 3

Дана цілочислена прямокутна матриця. Визначити:

- 1) кількість стовпців, що містять хоча б один нульовий елемент;
- 2) номер рядка, в якій знаходиться найдовша серія однакових елементів.

Варіант 4

Дана цілочислена прямокутна матриця. Визначити:

- 1) добуток елементів в тих рядках, які не містять від'ємних елементів;
- 2) максимум серед сум елементів діагоналей, паралельних головній діагоналі матриці.

Варіант 5

Дана цілочислена прямокутна матриця. Визначити:

- 1) суму елементів в тих стовбцях, які не містять від'ємних елементів;
- 2) мінімум серед сум модулів елементів діагоналей, паралельних побічної діагоналі матриці.

Варіант 6

Дана цілочислена прямокутна матриця. Визначити:

- 1) суму елементів в тих рядках, які містять хоча б один від'ємний елемент;
- 2) номери рядків і стовпців всіх сідлових точок матриці.

Примітка. Матриця A має сідлову точку A_{ij} , якщо A_{ij} є мінімальним елементом в i -му рядку і максимальним в j -му стовпці.

Варіант 7

Для заданої матриці розміром 8×8 знайти такі k , що k -й рядок матриці збігається з k -м стовпцем.

Знайти суму елементів в тих рядках, які містять хоча б один від'ємний елемент.

Варіант 8

Характеристикою стовпця цілочисленої матриці назвемо суму модулів його від'ємних непарних елементів. Переставляючи стовпці заданої матриці, розташовувати їх відповідно до зростання характеристик.

Знайти суму елементів в тих рядках, які містять хоча б один від'ємний елемент.

Варіант 9

Сусідами елемента A_{ij} в матриці назвемо елементи A_{kl} з $i-1 \leq k \leq i+1, j-1 \leq l \leq j+1, (k, l) \neq (i, j)$. Операція згладжування матриці дає нову матрицю того ж розміру, кожен елемент якої виходить як середнє арифметичне наявних сусідів відповідного елемента вихідної матриці. Побудувати результат згладжування заданої дійсної матриці розміром 10×10 .

У згладженій матриці знайти суму модулів елементів, розташованих нижче головної діагоналі.

Варіант 10

Елемент матриці називається локальним мінімумом, якщо він строго менше всіх наявних у нього сусідів. Підрахувати кількість локальних мінімумів заданої матриці розміром 10×10 .

Знайти суму модулів елементів, розташованих вище головної діагоналі.

Варіант 11

Коефіцієнти системи лінійних рівнянь задані у вигляді прямокутної матриці. За допомогою допустимих перетворень привести систему до трикутного вигляду.

Знайти кількість рядків, середнє арифметичне елементів яких менше заданої величини.

Варіант 12

Ущільнити задану матрицю, видаляючи з неї рядки і стовпці, заповнені нулями.

Знайти номер першого з рядків, що містить хоча б один додатний елемент.

Варіант 13

Здійснити циклічний зсув елементів прямокутної матриці на n елементів вправо або вниз (в залежності від введеного режиму). n може бути більше кількості елементів в рядку або стовпці.

Варіант 14

Здійснити циклічний зсув елементів квадратної матриці розмірності $M \times N$ вправо на k елементів таким чином: елементи 1-го рядка зсуваються в останній стовпець зверху вниз, з нього - в останній рядок справа наліво, з неї - в перший стовпець знизу вгору, з нього - в перший рядок; для інших елементів - аналогічно.

Варіант 15

Дана целочисленная прямокутна матриця. Визначити номер першого з стовпців, що містять хоча б один нульовий елемент. Характеристикою рядка целочисленной матриці назвемо суму її негативних парних елементів. Переставляючи рядки заданої матриці, розташовувати їх відповідно до убутання характеристик.

Варіант 16

Впорядкувати рядки цілочисленної прямокутної матриці по зростанню кількості однакових елементів в кожному рядку.

Знайти номер першого з стовпців, що не містить жодного від'ємного елементу.

Варіант 17

Шляхом перестановки елементів квадратної речовій матриці домогтися того, щоб її максимальний елемент перебував в лівому верхньому кутку, наступний за величиною - в позиції (2,2), наступний за величиною - в позиції (3,3), тощо, заповнивши таким чином всю головну діагональ. Знайти номер першого з рядків, що не містять жодного додатного елементу.

Варіант 18

Дана цілочислена прямокутна матриця. Визначити:

- 1) кількість рядків, що містять хоча б один нульовий елемент;
- 2) номер стовпчика, в якому знаходиться найдовша серія однакових елементів.

Варіант 19

Дана цілочислена квадратна матриця. Визначити:

- 1) суму елементів в тих рядках, які не містять від'ємних елементів;
- 2) мінімум серед сум елементів діагоналей, паралельних головній діагоналі.

Варіант 20

Дана цілочислена прямокутна матриця. Визначити:

- 1) кількість від'ємних елементів в тих рядках, які містять хоча б один нульовий елемент;
- 2) номери рядків і стовпців всіх сідлових точок матриці.

Примітка. Матриця A має сідлову точку A_{ij} , якщо A_{ij} є мінімальним елементом в i -му рядку і максимальним в j -му стовпці.

2.4. Зміст звіту

Звіт повинен містити:

1. Титульний аркуш.
2. Лістинги програм, що вирішують завдання.
3. При необхідності навести екрані форми, або консольний виведення результатів роботи програми.
4. Розширені висновки з лабораторної роботи.

Лабораторна робота 3. Технології програмування структур та розміщення структур у пам'яті.

3.1. Мета роботи

Навчитися використовувати базові технології програмування структур та методи розміщення структур в пам'яті. Вміти застосовувати структури даних для вирішення практичних задач.

3.2. Теоретичний матеріал

Необхідно вивчити матеріали лекцій 7 [13].

Виділемо такі ключові моменти, що стосуються технологій програмування структур даних, які є ключовими при вирішенні практичних задач:

1. Структури застосовуються для логічного об'єднання пов'язаних між собою даних різних типів. Елементи структури називаються полями. Поля можуть бути любого основного типу, масивом, вказівником, об'єднанням чи структурою.

2. Для звертання до поля використовується операція вибору: «точка» при звертанні через ім'я структури і «->» при звертанні через вказівник. Після опису структурного типу ставиться крапка з комою.

3. Структури одного типу можна присвоювати друг другу. Введення-виведення структур виконується поелементно.

4. Структури, пам'ять під які виділяє компілятор, можна ініціалізувати перерахуванням значень їх елементів.

3.3. Порядок виконання роботи

Виконати завдання відповідно до варіанту, програмну реалізацію та необхідні результати виконання програми занести в звіт.

Задання

Варіант 1

Описати структуру з ім'ям STUDENT, що містить наступні поля: прізвище та ініціали; номер групи; успішність (масив з п'яти елементів).

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з десяти структур типу STUDENT; записи повинні бути впорядковані за зростанням номера групи;
- виведення на дисплей прізвищ і номерів груп для всіх студентів, включених в масив, якщо середній бал студента більше 4.0;
- якщо таких студентів немає, вивести відповідне повідомлення.

Варіант 2

Описати структуру з ім'ям STUDENT, що містить наступні поля: прізвище та ініціали; номер групи; успішність (масив з п'яти елементів). Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з десяти структур типу STUDENT; записи повинні бути впорядковані за зростанням середнього бала;
- виведення на дисплей прізвищ і номерів груп для всіх студентів, які мають оцінки 4 і 5;
- якщо таких студентів немає, вивести відповідне повідомлення.

Варіант 3

Описати структуру з ім'ям STUDENT, що містить наступні поля: прізвище та ініціали; номер групи; успішність (масив з п'яти елементів).

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з десяти структур типу STUDENT; записи повинні бути впорядковані за алфавітом;
- виведення на дисплей прізвищ і номерів груп для всіх студентів, які мають хоча б одну оцінку 2;
- якщо таких студентів немає, вивести відповідне повідомлення.

Варіант 4

Описати структуру з ім'ям KYIVAVIA, що містить наступні поля: назва пункту призначення рейсу; номер рейсу; тип літака.

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з семи елементів типу KYIVAVIA; записи повинні бути впорядковані за зростанням номера рейсу;
- виведення на екран номерів рейсів і типів літаків, що вилітають в пункт призначення, назва якого співпало з назвою, введеним з клавіатури;
- якщо таких рейсів немає, видати на дисплей відповідне повідомлення.

Варіант 5

Описати структуру з ім'ям KYIVAVIA, що містить наступні поля: назва пункту призначення рейсу; номер рейсу; тип літака.

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з семи елементів типу KYIVAVIA; записи повинні бути розміщені в алфавітному порядку за назвами пунктів призначення;
- виведення на екран пунктів призначення і номерів рейсів, що обслуговуються літаком, тип якого введено з клавіатури;
- якщо таких рейсів немає, видати на дисплей відповідне повідомлення.

Варіант 6

Описати структуру з ім'ям `WORKER`, що містить наступні поля: прізвище та ініціали працівника; назва займаної посади; рік надходження на роботу. Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з десяти структур типу `WORKER`; записи повинні бути розміщені за алфавітом;
- виведення на дисплей прізвищ працівників, чий стаж роботи в організації перевищує значення, введене з клавіатури;
- якщо таких працівників немає, вивести на дисплей відповідне повідомлення.

Варіант 7

Описати структуру з ім'ям `TRAIN`, що містить наступні поля: назва пункту призначення; номер поїзда; час відправлення.

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з восьми елементів типу `TRAIN`; записи впорядкувати за алфавітом за назвами пунктів призначення;
- виведення на екран інформації про поїзди, що відправляються після введеного з клавіатури часу;
- якщо таких поїздів немає, видати на дисплей відповідне повідомлення.

Варіант 8

Описати структуру з ім'ям `TRAIN`, що містить наступні поля: назва пункту призначення; номер поїзда; час відправлення. Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з шести елементів типу `TRAIN`; записи повинні бути впорядковані за часом відправлення поїзда;
- виведення на екран інформації про поїзди, що прямують в пункт, назва якого введено з клавіатури;
- якщо таких поїздів немає, видати на дисплей відповідне повідомлення.

Варіант 9

Описати структуру з ім'ям `TRAIN`, що містить наступні поля: назва пункту призначення; номер поїзда; час відправлення. Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з восьми елементів типу `TRAIN`; записи повинні бути впорядковані за номерами поїздів;

- виведення на екран інформації про потяг, номер якого введено з клавіатури;
- якщо таких поїздів немає, видати на дисплей відповідне повідомлення.

Варіант 10

Описати структуру з ім'ям MARSH, що містить наступні поля: назва початкового пункту маршруту; назва кінцевого пункту маршруту; номер маршруту. Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з восьми елементів типу MARSH; записи повинні бути впорядковані за номерами маршрутів;
- виведення на екран інформації про маршрут, номер якого введено з клавіатури;
- якщо таких маршрутів немає, видати на дисплей відповідне повідомлення.

Варіант 11

Описати структуру з ім'ям MARSH, що містить наступні поля: назва початкового пункту маршруту; назва кінцевого пункту маршруту; номер маршруту. Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з восьми елементів типу MARSH; записи повинні бути впорядковані за номерами маршрутів;
- виведення на екран інформації про маршрути, які починаються або закінчуються в пункті, назва якого введено з клавіатури;
- якщо таких маршрутів немає, видати на дисплей відповідне повідомлення.

Варіант 12

Описати структуру з ім'ям NOTE, що містить наступні поля: прізвище, ім'я; номер телефону; дата народження (масив з трьох чисел). Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з восьми елементів типу NOTE; записи повинні бути впорядковані по датах народження;
- виведення на екран інформації про людину, номер телефону якого введено з клавіатури;
- якщо такого немає, видати на дисплей відповідне повідомлення.

Варіант 13

Описати структуру з ім'ям NOTE, що містить наступні поля: прізвище, ім'я; номер телефону; дата народження (масив з трьох чисел). Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з восьми елементів типу NOTE; записи повинні бути розміщені за алфавітом;
- виведення на екран інформації про людей, чиї дні народження припадають на місяць, значення якого введено з клавіатури;
- якщо таких немає, видати на дисплей відповідне повідомлення.

Варіант 14

Описати структуру з ім'ям NOTE, що містить наступні поля: прізвище, ім'я; номер телефону; дата народження (масив з трьох чисел).

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з восьми елементів типу NOTE; записи повинні бути впорядковані за трьома першими цифрами номера телефону;
- виведення на екран інформації про людину, чиє прізвище введена з клавіатури;
- якщо такого немає, видати на дисплей відповідне повідомлення.

Варіант 15

Описати структуру з ім'ям ZNAK, що містить наступні поля: прізвище, ім'я; знак зодіаку; дата народження (масив з трьох чисел).

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з восьми елементів типу ZNAK; записи повинні бути впорядковані по датах народження;
- виведення на екран інформації про людину, чиє прізвище введена з клавіатури;
- якщо такого немає, видати на дисплей відповідне повідомлення.

Варіант 16

Описати структуру з ім'ям ZNAK, що містить наступні поля: прізвище, ім'я; знак зодіаку; дата народження (масив з трьох чисел).

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з восьми елементів типу ZNAK; записи повинні бути впорядковані по датах народження;
- виведення на екран інформації про людей, які народилися під знаком, назва якого введено з клавіатури;
- якщо таких немає, видати на дисплей відповідне повідомлення.

Варіант 17

Описати структуру з ім'ям ZNAK, що містить наступні поля: прізвище, ім'я; знак зодіаку; дата народження (масив з трьох чисел). Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з восьми елементів типу ZNAK; записи повинні бути впорядковані по знакам Зодіаку;
- виведення на екран інформації про людей, які народилися в місяць, значення якого введено з клавіатури;
- якщо таких немає, видати на дисплей відповідне повідомлення.

Варіант 18

Описати структуру з ім'ям PRICE, що містить наступні поля: назва товару; назва магазину, в якому продається товар; вартість товару в грн. Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з восьми елементів типу PRICE; записи розмістити в алфавітному порядку за назвами товарів;
- виведення на екран інформації про товар, назва якого введено з клавіатури;
- якщо таких товарів немає, видати на дисплей відповідне повідомлення.

Варіант 19

Описати структуру з ім'ям PRICE, що містить наступні поля: назва товару; назва магазину, в якому продається товар; вартість товару в грн.

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з восьми елементів типу PRICE; записи розмістити в алфавітному порядку за назвами магазинів;
- виведення на екран інформації про товари, що продаються в магазині, назва якого введено з клавіатури;
- якщо такого магазину немає, видати на дисплей відповідне повідомлення.

Варіант 20

Описати структуру з ім'ям ORDER, що містить наступні поля: розрахунковий рахунок платника; розрахунковий рахунок одержувача; перераховується сума в руб.

Написати програму, що виконує наступні дії:

- введення з клавіатури даних в масив, що складається з восьми елементів типу ORDER; записи повинні бути розміщені в алфавітному порядку за розрахунковими рахунками платників;
- виведення на екран інформації про суму, знятої з розрахункового рахунку платника, введеного з клавіатури; якщо такого розрахункового рахунку немає, видати на дисплей відповідне повідомлення.

3.4. Зміст звіту

Звіт повинен містити:

1. Титульний аркуш.
2. Лістинги програм, що вирішують завдання.
3. При необхідності навести екрані форми, або консольний виведення результатів роботи програми.
4. Розширені висновки з лабораторної роботи.

Лабораторна робота 4. Технології програмування функцій та їх використання.

4.1. Мета роботи

Навчитися будувати та використовувати різного типу функції (функції з аргументами по замовчуванню, функції із змінною кількістю параметрів, лямбда-функції та інші) для вирішення практичних задач.

4.2. Теоретичний матеріал

Необхідно вивчити матеріали лекцій 11 [13].

Виділемо такі ключові моменти, що стосуються технологій програмування та використання функцій, які є ключовими при вирішенні практичних задач:

1. Функція - це іменована послідовність операторів, що виконує закінчену дію. Функції потрібні для спрощення структури програми. Інтерфейс грамотно написаної функції визначається її заголовком.

2. Для виклику функції треба вказати її ім'я і набір аргументів. У визначенні, в оголошенні і при виконанні функції типи і порядок проходження аргументів і параметрів повинні збігатися.

3. Передача параметрів в функцію може виконуватися за значенням або за адресою. Вхідні дані функції треба передавати за значенням або за константною посиланням, результати її роботи - через яке значення, а при необхідності передати більше однієї величини - через параметри по посиланню або вказівником.

4. При написанні функції потрібно передбачити всі можливі помилки і забезпечити користувача функції засобами їх діагностики. Виведення діагностичних повідомлень всередині функції небажана.

5. Функція може мати кілька параметрів зі значеннями за замовчуванням. Вони повинні знаходитися в кінці списку параметрів.

6. Масиви завжди передаються в функцію за адресою. Кількість елементів в масиві має передаватися окремим параметром.

7. Рекурсивна функція повинна містити хоча б одну нерекурсивну гілку. При використанні рекурсії слід враховувати виникаючі при цьому проблеми і накладні витрати.

8. У багатофайлових проектах важливо грамотно розбити задачу на підзадачі і розподілити функції по файлах. Для запобігання помилок повторного включення заголовних файлів слід використовувати варті включення.

8. В багатофайлових проектах важливо грамотно розбити задачу на підзадачі і розподілити функції по файлам.

4.3. Порядок виконання роботи

Оберати варіант індивідуального завдання відповідно до порядкового номеру по журналу. Реалізувати завдання у вигляді програмного коду та результати занести до звіту.

Варіант 1

Для зберігання даних про планшетні сканери описати структуру виду:

```
struct scan_info {
    char model [25]; // найменування моделі
    int price; // ціна
    double x_size; // горизонтальний розмір області сканування
    double y_size; // вертикальний розмір області сканування
    int optr; // оптична роздільна здатність
    int grey; // число градацій сірого
```

Написати функцію, яка записує в бінарний файл дані про сканер з наведеної структури. Структура файлу: в перших двох байтах розміщається значення типу int, що визначає кількість зроблених в файл записів; далі без пропусків розміщуються записи про сканерах.

Написати функцію, яка витягує з цього файлу дані про сканер в структуру типу scaninfo. Обов'язковий параметр - номер необхідної записи. Функція повинна повертати нульове значення, якщо читання пройшло успішно, і -1 в іншому випадку.

Привести приклад програми, що створює файл з даними про сканерах (дані вводяться з клавіатури) - 6-8 записів і виводить на дисплей дані про запрошений запис.

Всі необхідні дані для функцій повинні передаватися їм в якості параметрів. Використання глобальних змінних у функціях не допускається.

Варіант 2

Для зберігання даних про планшетні сканери описати структуру вигляду, що описано в варіанті 1.

Написати функцію, яка записує в бінарний файл дані про сканер з наведеної структури. Структура файлу: в перших двох байтах розміщається значення типу int, що визначає кількість зроблених в файл записів; далі без пропусків розміщуються записи про сканерах.

Написати функцію, яка сортує записи в описаному вище бінарному файлі по одній з наступних характеристик: ціна або число градацій сірого. Обов'язковий параметр - ознака, що задає критерій сортування.

Привести приклад програми, що створює файл з даними про сканери (дані вводяться з клавіатури) з не менше восьми записами і здійснює його сортування.

Всі необхідні дані для функцій повинні передаватися їм в якості параметрів. Використання глобальних змінних у функціях не допускається.

Варіант 3

Для зберігання даних про планшетні сканери описати структуру вигляду, що описано в варіанті 1.

Написати функцію, яка записує в бінарний файл дані про сканер з наведеної структури. Структура файлу: в перших чотирьох байтах розміщується значення типу long, що визначає кількість зроблених в файл записів; далі без пропусків розміщуються записи про сканери.

Написати функцію, яка сортує записи в описаному вище бінарному файлі по найменуванню моделі сканера.

Привести приклад програми, що створює файл з даними про сканери (дані вводяться з клавіатури) з не менше восьми записами і здійснює його сортування.

Всі необхідні дані для функцій повинні передаватися їм в якості параметрів. Використання глобальних змінних у функціях не допускається.

Варіант 4

Для зберігання даних про планшетні сканери описати структуру вигляду, що описано в варіанті 1.

Написати функцію, яка динамічно виділяє пам'ять під масив структур (не менш шести елементів), заповнює його даними в режимі діалогу і записує масив в бінарний файл. Структура файлу: в перших двох байтах розміщує значення типу int, що визначає кількість зроблених в файл записів; далі без пропусків розміщуються записи про сканерах.

Написати функцію, яка отримує дані про сканер з описаного вище бінарного файлу в структуру типу scanjinfo. Обов'язковий параметр - номер необхідного запису. Функція повинна повертати нульове значення, якщо читання пройшло успішно, і -1 в іншому випадку.

Привести приклад програми, що створює файл з даними про сканери (дані вводяться з клавіатури) з не менше восьми записами і здійснює виведення на дисплей даних про необхідної записи.

Всі необхідні дані для функцій повинні передаватися їм в якості параметрів. Використання глобальних змінних у функціях не допускається.

Варіант 5

Для зберігання даних про планшетні сканери описати структуру вигляду, що описано в варіанті 1.

Написати функцію, яка записує дані про сканер з наведеної структури в необхідну позицію в бінарному файлі. Структура файлу: в перших двох байтах розміщується значення типу int, що визначає кількість зроблених в файл записів; далі без пропусків розміщуються записи про сканерах. Запис може здійснюватися в будь-яку позицію, причому якщо між записом, що вводиться і останнім (або початком файлу) є пропуски, вони заповнюються нулями.

Написати функцію, яка «ущільнює» описаний вище бінарний файл шляхом видалення з нього записів, що містять всі нулі.

Привести приклад програми, що створює файл з даними про сканери (дані вводяться з клавіатури) з не менше шести записів і здійснює його ущільнення.

Всі необхідні дані для функцій повинні передаватися їм в якості параметрів. Використання глобальних змінних у функціях не допускається.

Варіант 6

Для зберігання даних про планшетні сканери описати структуру вигляду, що описано в варіанті 1.

Написати функцію, яка динамічно виділяє пам'ять під масив структур (не менш шести елементів), заповнює його даними в режимі діалогу і записує масив в бінарний файл. Структура файлу: в перших двох байтах розміщується значення типу `int`, що визначає кількість зроблених в файл записів; далі без пропусків розміщуються записи про сканерах.

Написати функцію, яка запитує дані про сканер в режимі діалогу і заміщає записи в бінарному файлі але заданому номеру. Обов'язковий параметр - номер замещаемой записи. Функція повинна повертати нульове значення, якщо запис пройшла успішно, і -1 в іншому випадку.

Привести приклад програми, що створює файл з даними про сканерах (дані вводяться з клавіатури) з не менше восьми записів і здійснює вставку нових даних про сканер.

Всі необхідні дані для функцій повинні передаватися їм в якості параметрів. Використання глобальних змінних у функціях не допускається.

Варіант 7

Для зберігання даних про планшетні сканери описати структуру вигляду, що описано в варіанті 1.

Написати функцію, яка записує в бінарний файл дані про сканер з наведеної структури. Структура файлу: в перших двох байтах розміщується значення типу `int`, що визначає кількість зроблених в файл записів; далі без пропусків розміщуються записи про сканерах.

Написати функцію, яка вводить дані про сканер з клавіатури в структуру `scan_info`, і якщо дані про це сканері відсутні у файлі, поміщає вміст структури в кінець файлу; в іншому випадку видає відповідне повідомлення.

Привести приклад програми, що створює файл з даними про сканерах (дані вводяться з текстового файлу) - 6-8 записів і доповнює файл записами про 2-3 сканерах, що вводяться з клавіатури.

Всі необхідні дані для функцій повинні передаватися їм в якості параметрів. Використання глобальних змінних у функціях не допускається.

Варіант 8

Для зберігання даних про планшетні сканери описати структуру вигляду, що описано в варіанті 1.

Написати функцію, яка записує в бінарний файл дані про сканер з наведеної структури. Структура файлу: в перших двох байтах розміщується

значення типу `int`, що визначає кількість зроблених в файл записів; далі без пропусків розміщуються записи про сканерах.

Написати функцію, яка вводить дані про сканер з клавіатури в структуру типу `scan_info` і поміщає її вміст на місце першого запису в файлі. Файл повинен існувати. При цьому, запис, що раніше займав першу позицію, поміщається на другу, другий запис на третю, тощо.

Привести приклад програми, що створює файл з даними про сканерах (дані вводяться з текстового файлу) - 6-8 записів і доповнює цей файл 1-2 новими записами, що вводяться з клавіатури.

Всі необхідні дані для функцій повинні передаватися їм в якості параметрів. Використання глобальних змінних у функціях не допускається.

Варіант 9

Для зберігання даних про планшетні сканери описати структуру вигляду, що описано в варіанті 1.

Написати функцію, яка запитує кількість сканерів, інформація про яких буде вводиться, динамічно виділяє пам'ять під масив структур відповідного розміру і заповнює його даними в режимі діалогу (з клавіатури). При цьому ім'я сканера може містити пробіли.

Написати функцію, яка записує даний масив в створюваний бінарний файл. Якщо ціна сканера менше 200, то дані про ці сканери в файл не записуються. Інформація про інші сканери поміщається в бінарний файл, причому спочатку пишуться дані про всі сканери, ім'я яких починається з маленької літери, а потім - з великої.

Структура файлу: в перших чотирьох байтах розміщається значення типу `long`, що визначає кількість зроблених в файл записів; далі без пропусків розміщуються записи про сканерах.

Привести приклад програми, що створює файл з даними про сканери і здійснює висновок на дисплей даних про необхідний запис (або всіх, або за номером).

Всі необхідні дані для функцій повинні передаватися їм в якості параметрів. Використання глобальних змінних у функціях не допускається.

Варіант 10

Для зберігання даних про ноутбуки описати структуру вигляду (при необхідності доповнивши її):

```
struct NOTEBOOK{
    char model[21];    // назва
    struct size{       // габаритні розміри
        float x;
        float y;
        float z;
    };
};
```

```

float w;           // вага
int price;        // ціна
}

```

Написати функцію, яка читає дані про ноутбуках з файлу note.txt в структуру наведеного виду.

Приклад структури файла note.txt:

Acer Note Light	2699	5.6	02.0x11.8x08.3	100	40	10.4	1	1024x0768	60	0.774
ASW ND5123T	3489	7.2	02.3x11.8x10.1	133	32	12.1	2	1024x0768	70	1.300
ARMNote TS80CD	3699	7.2	02.0x11.5x08.8	133	64	11.3	1	1024x0768	75	1.300
AST Ascentia P50	4499	7.5	02.3x11.3x09.0	133	40	11.3	1	0800x0600	70	0.774
BSI NP8657D	2605	8.0	02.3x11.8x09.3	133	40	11.3	1	1024x0768	60	0.810
BSI NP5265A	3765	8.2	02.5x12.0x09.0	150	32	12.1	2	1024x0768	70	1.300
Dell Xpi P100SD	3459	6.0	02.3x11.0x08.8	100	40	10.3	1	1024x0768	60	0.773
Digital HiNote	4799	4.0	01.3x11.0x08.8	120	40	10.4	1	0800x0600	56	1.000
Gateway Solo S5	4499	5.6	02.0x11.9x08.8	133	40	11.3	2	1024x0768	60	0.686
Hertz Z-Optima NB	3995	8.0	02.3x11.9x09.0	150	40	11.2	2	1024x0768	75	1.000
HP Omni Book 5500	6120	7.1	02.0x11.5x09.0	133	64	11.4	1	1024x0768	75	1.300
IBM ThinkPad 560	3749	4.1	01.3x11.8x08.8	120	40	12.1	2	1024x0768	85	0.774
NEC Versa 4080H	4780	6.6	02.3x11.8x09.5	120	48	10.4	1	0800x0600	70	0.776
Polywell Poly 500	3300	7.9	02.3x11.9x09.0	120	40	10.4	1	1024x0768	72	1.000
Samsung SENS 810	3667	8.7	02.3x11.5x09.5	100	32	11.4	2	1024x0768	75	0.773
Twinhead Slimnote	2965	7.4	02.0x11.5x08.0	075	64	10.4	1	1024x0768	70	0.772

У файлі note.txt знаходиться текстова інформація про ноутбуки. Кожен рядок містить дані про одну модель. Дані в рядку розміщуються в наступних полях:

- 1: 20 - найменування моделі;
- 21: 24 - ціна в доларах (ціле число);
- 26: 28 - маса ноутбука в кг (число з десятковою крапкою з чотирьох символів);
- 30: 43 - габаритні розміри ноутбука в дюймах (ВИСОТАхДЛІНАхШІРІНА - три числа з десятковою точкою (4 символу, включаючи точку, розділені V));
- 44: 47 - частота процесора в МГц (ціле число з трьох символів);
- 49: 50 - максимальний обсяг ОЗУ в мегабайтах (ціле число з двох символів);
- 52: 55 - розмір діагоналі дисплея в дюймах (число з десятковою крапкою з чотирьох символів, включаючи точку);
- 57 - розмір відеопам'яті в мегабайтах - ціле число з одного символу;
- 59: 67 - роздільна здатність дисплея в пікселях (два цілих числа, разделение V);
- 69: 70 - частота регенерації дисплея в Гц (ціле число з двох символів);
- 72: 76 - обсяг HDD в гигабайтах (число з десятковою крапкою з п'яти символів).

Все неописані позиції заповнені пробілами.

Написати функцію, яка записує вміст структури в кінець бінарного файлу. Структура бінарного файлу: перші два байта (ціле) - число записів в файлі; далі записи в форматі структури NOTEBOOK.

Написати програму, в якій на основі розроблених функцій здійснюється читання даних тільки для тих ноутбуків, частота процесора яких більше 120 МГц, і запис в бінарний файл зменшенням ціни.

Варіант 11

Для зберігання даних про ноутбуки описати структуру вигляду, що описана в варіанті 10.

Написати функцію, яка читає дані про ноутбуках з файлу note.txt в структуру наведеного виду. Написати функцію, яка записує вміст структури в кінець бінарного файлу. Структура бінарного файлу: перші два байта (ціле) - число записів в файлі; далі записи в форматі структури NOTEBOOK.

Написати програму, в якій на основі розроблених функцій здійснюється читання даних тільки для тих ноутбуків, обсяг HDD яких менше 1 Гбайт, і запис зчитаних даних в бінарний файл в алфавітному порядку за найменуванням.

Варіант 12

Для зберігання даних про ноутбуки описати структуру вигляду, що описана в варіанті 10.

Написати функцію, яка читає дані про ноутбуках з файлу note.txt в структуру наведеного виду. Написати функцію, яка записує вміст структури в кінець бінарного файлу. Структура бінарного файлу: перші два байта (ціле) - число записів в файлі; далі записи в форматі структури NOTEBOOK.

Написати програму, в якій на основі розроблених функцій здійснюється запис в двійковий файл даних тільки про тих ноутбуках, цілу кількість яких в одному кубічному метрі не перевищує 285 штук.

Варіант 13

Для зберігання даних про ноутбуки описати структуру вигляду, що описана в варіанті 10.

Написати функцію, яка читає дані про ноутбуках з файлу note.txt в структуру наведеного виду. Написати функцію, яка записує вміст структури в кінець бінарного файлу. Структура бінарного файлу: перші два байта (ціле) - число записів в файлі; далі записи в форматі структури NOTEBOOK.

Написати програму, в якій на основі розроблених функцій здійснюється запис в двійковий файл даних тільки про тих ноутбуках, максимальний обсяг ОЗУ яких не менше 40 Мбайт, відсортованих за обсягом.

Варіант 14

Для зберігання даних про ноутбуки описати структуру вигляду, що описана в варіанті 10.

Написати функцію, яка читає дані про ноутбуках з файлу note.txt в структуру наведеного виду. Написати функцію, яка записує вміст структури в кінець бінарного файлу. Структура бінарного файлу: перші два байта - ціле число записів в файлі; далі записи в форматі структури NOTEBOOK.

Написати програму, в якій на основі розроблених функцій відбувається запис в двійковий файл даних тільки про тих ноутбуках, діагональ дисплея яких більше одинадцяти дюймів.

Варіант 15

Для зберігання даних про ноутбуках описати структуру виду (при необхідності доповнивши її):

```
struct NOTEBOOK{
    struct disp res{           // роздільна здатність дисплея
        int x;                 // по горизонталі
        int y;                 // по вертикалі
    };
    int f;                     // частота регенерації
    float d;                   // розмір діагоналі матриці
    int price;                 // ціна
    char model[21];           // назва
}
```

Написати функцію, яка читає дані про ноутбуках з файлу note.txt в структуру наведеного виду. Написати функцію, яка записує вміст структури в кінець бінарного файлу. Структура бінарного файлу: перші два байта - ціле число записів в файлі; далі записи в форматі структури NOTEBOOK.

Написати програму, в якій на основі розроблених функцій здійснюється запис в двійковий файл даних тільки про тих ноутбуках, вага яких менше 7 кг, відсортованих в порядку зростання ціни.

Варіант 16

Для зберігання даних про ноутбуки описати структуру виду, описаного в варіанті 15.

Написати функцію, яка читає дані про ноутбуки з файлу note.txt в структуру наведеного виду. Написати функцію, яка записує вміст структури в кінець бінарного файлу. Структура бінарного файлу: перші два байта ціле число записів в файлі; далі записи в форматі структури NOTEBOOK.

Написати програму, в якій на основі розроблених функцій здійснюється запис в двійковий файл даних тільки про тих ноутбуках, обсяг відеопам'яті яких 2 Мбайт, відсортованих в порядку зменшення тактової частоти процесора.

Варіант 17

Для зберігання даних про ноутбуки описати структуру виду, описаного в варіанті 15.

Написати функцію, яка читає дані про ноутбуки з файлу note.txt в структуру наведеного виду. Написати функцію, яка записує вміст структури в кінець бінарного файлу. Структура бінарного файлу: перші два байта - ціле число записів в файлі; далі записи в форматі структури NOTEBOOK.

Написати програму, в якій на основі розроблених функцій здійснюється запис в двійковий файл даних тільки про тих ноутбуках, обсяг HDD яких більше 1 Гбайт, відсортованих в порядку зростання розміру діагоналі дисплея.

Варіант 18

Для зберігання даних про ноутбуках описати структуру виду, описаного в варіанті 15.

Написати функцію, яка читає дані про ноутбуки з файлу note.txt в структуру наведеного виду. Написати функцію, яка записує вміст структури в кінець бінарного файлу. Структура бінарного файлу: перші два байта - ціле число записів в файлі; далі записи в форматі структури NOTEBOOK.

Написати програму, в якій на основі розроблених функцій здійснюється запис в двійковий файл даних тільки про тих ноутбуках, тактова частота процесора яких більше 120МГц, відсортованих в порядку зменшення ваги.

Варіант 19

Для зберігання даних про ноутбуках описати структуру виду (при необхідності доповнивши її):

```
struct NOTEBOOK{
    struct disp_res{           // роздільна здатність дисплея
        int x;                 // по горизонталі
        int y;                 // по вертикалі
    };
    int f;                     // частота регенерації
    float d;                   // розмір діагоналі дисплея
    float hdd;                 // обсяг диску
    char model[21];            // назва
}
```

Написати функцію, яка читає дані про ноутбуках з файлу note.txt в структуру наведеного виду. Написати функцію, яка записує вміст структури в кінець бінарного файлу. Структура бінарного файлу: перші два байта ціле число записів в файлі; далі записи в форматі структури NOTEBOOK. Написати програму, в якій на основі розроблених функцій здійснюється запис в двійковий файл даних тільки про тих ноутбуках, тактова частота процесора яких більше 120МГц, відсортовані в порядку зростання ціни.

Варіант 20

Для зберігання даних про ноутбуках описати структуру виду, описаного в варіанті 19.

Написати функцію, яка читає дані про ноутбуках з файлу note.txt в структуру наведеного виду. Написати функцію, яка записує вміст структури в кінець бінарного файлу. Структура бінарного файлу: перші два байта - ціле число записів в файлі; далі записи в форматі структури NOTEBOOK. Написати програму, в якій на основі розроблених функцій здійснюється запис в двійковий файл даних тільки про тих ноутбуках, ціна яких більше \$3500, відсортовані в порядку зростання тактової частоти процесора.

4.4. Зміст звіту

Звіт повинен містити:

1. Титульний аркуш.
2. Лістинги програм, що вирішують завдання.
3. При необхідності навести екрані форми, або консольний виведення результатів роботи програми.
4. Розширені висновки з лабораторної роботи.

Лабораторна робота 5. Перевантаження та шаблони функцій

5.1. Мета роботи

Вивчити особливості перевантаження функцій та побудови шаблонів функцій. Навчитися застосовувати шаблони функцій при вирішенні практичних задач.

5.2. Теоретичний матеріал

Необхідно вивчити матеріали лекції 11 [13].

Виділемо такі ключові моменти, що стосуються технологій програмування та використання шаблонів функцій та особливості перевантажень функцій, які є ключовими при вирішенні практичних задач:

1. Перевантаженням функцій називається використання декількох функцій з одним ім'ям і різними типами параметрів. Перевантаження застосовується, коли одна і та же дія реалізується по-різному для різних типів або структур даних.

2. При написанні перевантажених функцій необхідно, щоб в процесі пошуку потрібного варіанту функції за її викликом не виникало неоднозначності. Неоднозначність може виникнути через перетворення типів, параметрів за замовчуванням і посилань.

3. Функції не можуть бути перевантажені, якщо опису їх параметрів розрізняються тільки модифікатором `const` або використанням посилання.

4. Шаблони функцій застосовуються для запису ідентичних дій над даними різних типів. Інстанціювання шаблону функції - це створення компілятором конкретного варіанту функції.

5. Шаблони можна перевантажувати як шаблонами, так і звичайними функціями.

5.3. Порядок виконання роботи

Оберіть одне з варіантів завдань, що наведені нижче. Реалізувати у вигляді програмного коду та результати занести до звіту.

Завдання:

Виконати варіант завдання з лабораторної роботи 2, п. 2.3.1. Блок завдань 1 (стор. 10), оформивши кожен пункт варіанту завдання у вигляді шаблону функції. Всі необхідні дані для функцій повинні передаватися їм в якості параметрів. Використання глобальних змінних у функціях не допускається. Привести приклади програм, що використовують ці шаблони для типів `int`, `float` і `double`.

5.4. Зміст звіту

Звіт повинен містити:

1. Титульний аркуш.
2. Лістинги програм, що вирішують завдання.
3. При необхідності навести екрані форми, або консольний виведення результатів роботи програми.
4. Розширені висновки з лабораторної роботи.

Лабораторна робота 6. Засоби мови C для опрацювання стрінгів та файлів.

6.1. Мета роботи

Навчитися технологіям програмування та обробки стрінгів мови C. Вивчити особливості обробки текстової інформації, що занесена в файл.

6.2. Теоретичний матеріал

Необхідно вивчити матеріали лекції 13, 14 [13].

Виділемо такі ключові моменти, що стосуються технологій програмування та обробки стрінгів мови C, які є ключовими при вирішенні практичних задач:

1. При завданні довжини стрінгу необхідно враховувати завершальний нуль-символ. Довжина нединамічного стрінгу повинна бути сталим виразом, довжина динамічного стрінгу може бути змінною. Динамічні стрінги не можна ініціалізувати при створенні.

2. Присвоєння стрінгів виконується за допомогою функцій бібліотеки.

3. Для консольного введення-виведення стрінгів використовуються або об'єкти `cin` і `cout`, або функції бібліотеки `gets`, `scanf` і `puts`, `printf`. Введення-виведення з файлу може виконуватися за допомогою або об'єктів класів `ifstream` і `ofstream`, або функцій бібліотеки `fgets`, `fscanf` і `fputs`, `fprintf`.

4. Введення стрінгу за допомогою операції `>>` виконується до першого пробілу символу. Для введення стрінгу, що містить пробіли, можна використовувати або методи `getline`, або `get` класу `istream`, або функції `gets` і `scanf`.

5. Змішувати в одній програмі введення-виведення за допомогою потокових класів і за допомогою функцій бібліотеки не рекомендується. Програма, написана з використанням функцій, а не класів введення-виведення, може вийти більш швидкодіючою, але менш безпечною.

6. Недоліком C-стрінгів в порівнянні з класом `string` є відсутність перевірки виходу стрінгів за межі відведеної їй пам'яті. Вихід за межі стрінгу і відсутність нуль-символу є поширеними причинами помилок в програмах.

7. Засоби, надані мовою і бібліотеками, впливають на алгоритм вирішення задачі, тому перед розробкою алгоритму необхідно їх вивчити.

8. Розбивайте написання і налагодження програми на послідовність кроків.

6.3. Порядок виконання роботи

Оберіть по одному варіанту завдань в кожному блоці відповідно до порядкового номеру по журналу. Реалізувати у вигляді програмного коду та результати занести до звіту.

6.3.1. Блок завдань 1

Варіант 1.

Написати програму, яка зчитує з текстового файлу три речення і виводить їх в зворотному порядку.

Варіант 2.

Написати програму, яка зчитує текст з файлу і виводить на екран тільки речення, що містять введене з клавіатури слово.

Варіант 3.

Написати програму, яка зчитує текст з файлу і виводить на екран тільки стрінги, що містять двозначні числа.

Варіант 4.

Написати програму, яка зчитує текст англійською мовою з файлу і виводить на екран слова, що починаються з голосних букв.

Варіант 5.

Написати програму, яка зчитує текст з файлу і виводить його на екран, міняючи місцями кожні два сусідніх слова.

Варіант 6.

Написати програму, яка зчитує текст з файлу і виводить на екран тільки речення, які не містять ком.

Варіант 7.

Написати програму, яка зчитує текст з файлу і визначає, скільки в ньому слів, що складаються не більше ніж з чотирьох літер.

Варіант 8.

Написати програму, яка зчитує текст з файлу і виводить на екран тільки цитати, тобто речення, що лапках.

Варіант 9.

Написати програму, яка зчитує текст з файлу і виводить на екран тільки речення, що складаються з певної кількості слів.

Варіант 10.

Написати програму, яка зчитує текст англійською мовою з файлу і виводить на екран слова тексту, що починаються і закінчуються голосними літерами.

Варіант 11.

Написати програму, яка зчитує текст з файлу і виводить на екран тільки стрінги, що не містять двозначних чисел.

Варіант 12.

Написати програму, яка зчитує текст з файлу і виводить на екран тільки речення, що починаються з тире, перед яким можуть перебувати тільки пробільні символи.

Варіант 13.

Написати програму, яка зчитує текст англійською мовою з файлу і виводить його на екран, замінивши кожну першу букву слів, що починаються з голосної букви, на прописну.

Варіант 14.

Написати програму, яка зчитує текст з файлу і виводить його на екран, замінивши цифри від 0 до 9 на слова «нуль», «один», ..., «дев'ять», починаючи кожне речення з нового рядка.

Варіант 15.

Написати програму, яка зчитує текст з файлу, знаходить найдовше слово і визначає, скільки разів воно зустрілося в тексті.

Варіант 16.

Написати програму, яка зчитує текст з файлу і виводить на екран спочатку речення зі знаком питання, а потім зі знаком оклику.

Варіант 17.

Написати програму, яка зчитує текст з файлу і виводить його на екран, додаючи після кожного речення, скільки разів зустрілося в ньому введене з клавіатури слово.

Варіант 18.

Написати програму, яка зчитує текст з файлу і виводить на екран всі його речення в зворотному порядку.

Варіант 19.

Написати програму, яка зчитує текст з файлу і виводить на екран спочатку речення, що починаються з однолітерних слів, а потім всі інші.

Варіант 20.

Написати програму, яка зчитує текст з файлу і виводить на екран речення, що містять максимальну кількість знаків пунктуації.

6.3.2. Блок завдань 2

Функції бібліотеки для працювання з стрінгами та символами.

Варіант 1

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна

перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ після натискання довільної клавіші по черзі виділяє кожне речення тексту;
- ☐ визначає кількість речень в тексті.

Варіант 2

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ після натискання довільної клавіші по черзі виділяє кожне слово тексту;
- ☐ визначає кількість слів у тексті.

Варіант 3

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ після натискання довільної клавіші по черзі виділяє кожне слово тексту, що закінчується на голосну букву;
- ☐ визначає кількість слів у тексті, що закінчуються на голосну букву.

Варіант 4

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів. Текст має складатися з трьох пропозицій (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ після натискання довільної клавіші по черзі виділяє кожне речення тексту в послідовності 2, 1, 3.

Варіант 5

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ після натискання довільної клавіші по черзі виділяє кожне зі слів текста, у яких перший і останній символи збігаються;
- ☐ визначає кількість слів у тексті, у яких перший і останній символи збігаються.

Варіант 6

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ після натискання довільної клавіші по черзі виділяє кожне слово тексту, що починається на голосну букву;
- ☐ визначає кількість слів у тексті, що починаються на голосну букву.

Варіант 7

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ визначає кількість символів в найдовшому слові;
- ☐ після натискання довільної клавіші по черзі виділяє кожне слово тексту, що містить максимальну кількість символів.

Варіант 8

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ визначає кількість символів в самому короткому слові;
- ☐ після натискання довільної клавіші по черзі виділяє кожне слово тексту, що містить мінімальну кількість символів.

Варіант 9

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ визначає в кожному реченні тексту кількість символів, відмінних від букв і пробілу;
- ☐ після натискання довільної клавіші по черзі виділяє кожне речення тексту, а в виділеному реченні - по черзі всі символи, відмінні від букв і пробілу.

Варіант 10

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна

перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ визначає кількість речень тексту і кількість слів у кожному реченні;
- ☐ після натискання довільної клавіші по черзі виділяє кожне речення тексту, а в виділеному реченні - по черзі всі слова.

Варіант 11

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ визначає кількість букв 'а' в останньому слові тексту;
- ☐ після натискання довільної клавіші виділяє останнє слово тексту, а в виділеному слові по черзі всі букви 'а'.

Варіант 12

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ визначає найдовшу послідовність цифр в тексті (вважати, що будь-яку кількість прогалів між двома цифрами не перериває послідовність цифр);
- ☐ після натискання довільної клавіші по черзі виділяє кожен послідовність цифр, що містить максимальну кількість символів.

Варіант 13

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ визначає порядковий номер заданого слова в кожному реченні тексту (задане слово вводиться з клавіатури);
- ☐ після натискання довільної клавіші по черзі виділяє кожне речення тексту, а в виділеному реченні - задане слово.

Варіант 14

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 700 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;

☐ після натискання довільної клавіші по черзі виділяє в тексті задане слово (задане слово вводити з клавіатури);

☐ виводить текст на екран дисплея ще раз, викидаючи з нього задане слово і видаляючи зайві пробіли.

Варіант 15

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 700 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

☐ виводить текст на екран дисплея;

☐ після натискання довільної клавіші по черзі виділяє в тексті задані слова, які потрібно поміняти місцями (задані слова вводити з клавіатури);

☐ виводить текст на екран дисплея ще раз, змінюючи в ньому місцями задані слова і видаляючи зайві пробіли.

Варіант 16

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 700 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

☐ виводить вихідний текст на екран дисплея;

☐ після натискання довільної клавіші по черзі виділяє в тексті задане слово (заданий слово вводити з клавіатури);

☐ виводить текст на екран дисплея ще раз, укладаючи задане слово в лапки, і по черзі виділяє задане слово разом з лапками.

Варіант 17

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 700 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT.

Написати програму, яка:

☐ виводить вихідний текст на екран дисплея;

☐ виводить текст на екран дисплея ще раз, вставляючи в кожне речення в якості останнього заданий слово, введене з клавіатури в якості вихідних даних.

☐ після натискання довільної клавіші по черзі виділяє в тексті вставлене слово.

Варіант 18

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 500 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

☐ виводить текст на екран дисплея;

- ☐ після натискання довільної клавіші по черзі виділяє в тексті зайві пропуски між словами;
- ☐ виводить текст на екран дисплея ще раз, прибираючи зайві пропуски між словами і починаючи кожне речення з нового рядка.

Варіант 19

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 700 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ після натискання довільної клавіші по черзі виділяє в тексті задане слово (задане слово вводиться з клавіатури);
- ☐ виводить текст на екран дисплея ще раз, замінюючи в заданому слові малі літери прописними.

Варіант 20

За допомогою текстового редактора створити файл, що містить текст, довжина якого не перевищує 1000 символів (довжина рядка тексту не повинна перевищувати 70 символів). Назва файлу повинна мати розширення DAT. Написати програму, яка:

- ☐ виводить текст на екран дисплея;
- ☐ визначає найбільшу кількість послідовних пробілів в тексті;
- ☐ після натискання довільної клавіші по черзі виділяє кожну з послідовностей пробілів максимальної довжини.

6.4. Зміст звіту

Звіт повинен містити:

1. Титульний аркуш.
2. Лістинги програм, що вирішують завдання.
3. При необхідності навести екранні форми, або консольний виведення результатів роботи програми.
4. Розширені висновки з лабораторної роботи.

Лабораторна робота 7. Динамічні структури даних.

7.1. Мета роботи

Вивчення технологій керування пам'яттю за допомогою динамічних структур даних. Навчитися реалізовувати динамічні об'єкти для вирішення практичних задач.

7.2. Теоретичний матеріал

Необхідно вивчити матеріали лекції 10 [13].

Виділемо такі ключові моменти, що стосуються технологій програмування динамічних структур даних, які є ключовими при вирішенні практичних задач:

1. Динамічними структурами даних називаються блоки в динамічній пам'яті, що пов'язані один з одним за допомогою вказівників. Структури розрізняються способами зв'язку окремих елементів і допустимими операціями над ними. Найбільш поширеними структурами є лінійний список (однозв'язний або двузв'язний), стек, черга і бінарне дерево.

2. Елемент будь динамічної структури даних складається з інформаційних полів і полів вказівників.

3. Для стека визначені операції приміщення елемента в вершину і вибірки елемента з вершини. Для черги визначені операції приміщення елемента в кінець черги і вибірка елемента з її початку. Допускається вставляти і видаляти елементи в довільне місце лінійного списку.

4. Бінарне дерево складається з вузлів, кожен з яких містить, крім даних, не більше двох посилань на різні бінарні дерева. На кожен вузол є рівно одне посилання. Кожен вузол дерева характеризується унікальним ключем. Допускається вставляти і видаляти елементи в довільне місце дерева.

5. Якщо для кожного вузла всі ключі його лівого піддерева менше ключа цього вузла, а всі ключі його правого піддерева - більше, то таке дерево називається деревом пошуку. У дереві пошуку можна знайти елемент по ключу, рухаючись від кореня і переходячи на ліве або праве піддерево в залежності від значення ключа в кожному вузлі, що набагато ефективніше пошуку в списку.

6. Динамічні структури в деяких випадках більш ефективно реалізовувати за допомогою масивів.

7.3. Порядок виконання роботи

Оберіть варіант завдань відповідно до порядкового номеру по журналу. Реалізувати у вигляді програмного коду та результати занести до звіту.

Завдання:

Варіант 1

Скласти програму, яка містить динамічну інформацію про наявність автобусів в автобусному парку. Відомості про кожний автобус включають номер

автобуса, прізвище та ініціали водія і номер маршруту. Програма повинна забезпечувати:

- початкове формування даних про всі автобуси в парку у вигляді списку;
- при виїзді кожного автобуса з парку вводиться номер автобуса, і програма видаляє дані про це автобусі зі списку автобусів, які перебувають в парку, і записує ці дані в список автобусів, які перебувають на маршруті;
- при в'їзді кожного автобуса в парк вводиться номер автобуса, і програма видаляє дані про це автобусі зі списку автобусів, які перебувають на маршруті, і записує ці дані в список автобусів, які перебувають в парку;
- на вимогу видаються відомості про автобуси, які перебувають в парку, або про автобуси, які перебувають на маршруті.

Варіант 2

Скласти програму, яка містить поточну інформацію про книги в бібліотеці. Відомості про книги включають:

- номер УДК;
- прізвище та ініціали автора;
- назва;
- рік видання;
- кількість примірників цієї книги в бібліотеці.

Програма повинна забезпечувати:

- початкове формування даних про книгах у вигляді двійкового дерева;
- додавання даних про книгах, тих, хто влаштовується в бібліотеку;
- видалення даних про списуються книгах;
- на вимогу видаються відомості про наявність книг в бібліотеці, впорядковані по роках видання.

Варіант 3

Скласти програму, яка містить поточну інформацію про заявки на авіаквитки.

Кожна заявка включає:

- пункт призначення;
- номер рейсу;
- прізвище та ініціали пасажирів;
- бажану дату вильоту.

Програма повинна забезпечувати:

- зберігання всіх заявок у вигляді списку;
- додавання заявок в список;
- видалення заявок;
- виведення заявок по заданому номеру рейсу і дату вильоту;
- виведення всіх заявок.

Варіант 4

Скласти програму, яка містить поточну інформацію про заявки на авіаквитки.

Кожна заявка включає:

- пункт призначення;
- номер рейсу;
- прізвище та ініціали пасажирів;
- бажану дату вильоту.

Програма повинна забезпечувати:

- зберігання всіх заявок у вигляді двійкового дерева;
- додавання і видалення заявок;
- по заданому номеру рейсу і дату вильоту виведення заявок з їх подальшим видаленням;
- виведення всіх заявок.

Варіант 5

Скласти програму, яка містить поточну інформацію про книги в бібліотеці.

Відомості про книгах включають:

- номер УДК;
- прізвище та ініціали автора;
- назва;
- рік видання;
- кількість примірників цієї книги в бібліотеці.

Програма повинна забезпечувати:

- початкове формування даних про всі книги в бібліотеці у вигляді списку;
- при видачі кожної книги на руки вводиться номер УДК, і програма зменшує значення кількості книг на одиницю або видає повідомлення про те, що необхідної книги в бібліотеці немає або необхідна книга знаходиться на руках;
- при поверненні кожної книги вводиться номер УДК, і програма збільшує значення кількості книг на одиницю;
- на вимогу видаються відомості про наявність книг в бібліотеці.

Варіант 6

Скласти програму, яка містить динамічну інформацію про наявність автобусів в автобусному парку.

Відомості про кожному автобусі включають:

- номер автобуса;
- прізвище та ініціали водія;
- номер маршруту;
- ознака того, де знаходиться автобус - на маршруті або в парку.

Програма повинна забезпечувати:

- початкове формування даних про всі автобусах у вигляді списку;

- при виїзді кожного автобуса з парку вводиться номер автобуса, і програма встановлює значення ознаки «автобус на маршруті»;
- при в'їзді кожного автобуса в парк вводиться номер автобуса, і програма встановлює значення ознаки «автобус в парку»;
- на вимогу видаються відомості про автобуси, які перебувають в парку, або про автобуси, які перебувають на маршруті.

Варіант 7

Створити програму, відшукувати прохід по лабіринту. Лабіринт представляється у вигляді матриці, що складається з квадратів. Кожен квадрат або відкритий, або закритий. Вхід в закритий квадрат заборонений. Якщо квадрат відкритий, то вхід в нього можливий з боку, але не з кута. Кожен квадрат визначається його координатами в матриці. Програма знаходить прохід через лабіринт, рухаючись від заданого входу. Після відшукування проходу програма виводить знайдений шлях у вигляді координат квадратів. Для зберігання шляху використовувати стек.

Варіант 8

Гаражна стоянка має одну стояночну смугу, причому єдиний в'їзд і він же виїзд знаходиться в одному кінці смуги. Якщо власник автомобіля приходить забрати автомобіль, який не є найближчим до виходу, то всі автомобілі, що загороджують проїзд, видаляються, машина даного власника виводиться зі стоянки, а інші машини повертаються на стоянку в вихідному порядку. Написати програму, яка моделює процес прибуття і від'їзду машин. Прибуття або від'їзд автомобіля задається командним рядком, яка містить ознаку прибуття або від'їзду і номер машини. Програма повинна виводити повідомлення під час вїзду або виїзду будь-якої машини. При виїзді автомобіля зі стоянки повідомлення повинно містити число випадків, коли машина була вилучена зі стоянки для забезпечення виїзду інших автомобілів.

Варіант 9

Написати програму, що моделює заповнення гнучкого магнітного диска. Загальний обсяг пам'яті на диску 360 Кбайт. Файли мають довільну довжину від 18 байт до 32 Кбайт. В процесі роботи файли або записуються на диск, або видаляються з нього.

На початку роботи файли записуються підряд один за одним. Після видалення файлу на диску утворюється вільна ділянка пам'яті, і знову записується файл або розміщується на вільній ділянці, або, якщо файл не поміщається в вільну ділянку, розміщується після останнього записаного файлу.

У разі, коли файл перевершує довжину найбільшої вільної ділянки, видається аварійне повідомлення. Вимога на запис або видалення файлу задається в командному рядку, яка містить ім'я файлу, його довжину в байтах, ознака записи або видалення. Програма повинна видавати за запитом відомості про зайнятих і вільних ділянках пам'яті на диску.

Вказівка: створити списки зайнятих і вільних ділянок пам'яті на диску.

Варіант 10

У файловій системі каталогфайлів, що організована у вигляді лінійного списку. Для кожного файлу в каталозі міститься ім'я файлу, дата створення і кількість звернень до файлу. Написати програму, яка забезпечує:

- початкове формування каталогу файлів;
- виведення каталогу файлів;
- видалення файлів, дата створення яких менше заданої;
- вибірку файлу з найбільшою кількістю звернень.

Програма повинна забезпечувати діалог за допомогою меню і контроль помилок введення.

Варіант 11

Покажчик організований у вигляді лінійного списку. Кожен компонент покажчика містить слово і номери сторінок, на яких це слово зустрічається. Кількість номерів сторінок, що відносяться до одного слова, - від одного до десяти.

Написати програму, яка забезпечує:

- початкове формування предметного покажчика;
- виведення предметного покажчика;
- виведення номерів сторінок для заданого слова.

Програма повинна забезпечувати діалог за допомогою меню і контроль помилок введення.

Варіант 12

Текст допомоги для деякої програми організований у вигляді лінійного списку. Кожен компонент тексту допомоги містить термін (слово) і текст, що містить пояснення до цього терміну. Кількість рядків тексту, що відносяться до одного терміну, становить від однієї до п'яти. Написати програму, яка забезпечує:

- початкове формування тексту допомоги;
- виведення тексту допомоги;
- виведення пояснюючого тексту для заданого терміну.

Програма повинна забезпечувати діалог за допомогою меню і контроль помилок введення.

Варіант 13

Картотека в бюро обміну квартир організована у вигляді лінійного списку. Відомості про кожну квартиру включають кількість кімнат, поверх, площа та адреса. Написати програму, яка забезпечує:

- початкове формування картотеки;
- введення заявки на обмін;
- пошук в картотеці відповідного варіанту: за однакової кількості кімнат і поверху і відмінності площ в межах 10%

відповідна картка виводиться і видаляється зі списку, в іншому випадку надійшла заявка включається в список;

- виведення всього списку.

Програма повинна забезпечувати діалог за допомогою меню і контроль помилок введення.

Варіант 14

Англо-російський словник побудований у вигляді двійкового дерева. Кожен компонент містить англійське слово, відповідне йому російське слово і лічильник кількості звернень до даної компоненті.

Спочатку дерево формується в порядку англійського алфавіту. В процесі експлуатації словника при кожному зверненні до компоненту до лічильника звернень додається одиниця. Написати програму, яка

- забезпечує початковий введення словника з конкретними значеннями лічильників звернень;
- формує нове представлення словника у вигляді двійкового дерева за наступним алгоритмом: а) в старому словнику шукається компонент з найбільшим значенням лічильника звернень; б) знайдений компонент заноситься в новий словник і видаляється зі старого; в) перехід до п. а) до вичерпання вихідного словника.
- забезпечує виведення вихідного і нового словників.

Програма повинна забезпечувати діалог за допомогою меню і контроль помилок введення.

Варіант 15

Анкета для опитування населення містить дві групи питань. Перша група містить відомості про респондента: вік; підлога; освіти (початкова, середня, вища). Друга група містить власне питання анкети, відповіддю на який може бути або ТАК, або НІ.

Написати програму, яка

- забезпечує початковий введення анкет і формує з них лінійний список;
- на основі аналізу анкет видає відповіді на наступні питання: а) скільки чоловіків старше 40 років, які мають вищу освіту, відповіли ТАК на питання анкети; б) скільки жінок молодше 30 років, що мають середню освіту, відповіли НІ на питання анкети; в) скільки чоловіків молодше 25 років, що мають початкову освіту, відповіли ТАК на питання анкети;
- забезпечує виведення всіх анкет і відповідей на питання.

Програма повинна забезпечувати діалог за допомогою меню і контроль помилок введення.

Варіант 16

Написати програму, яка містить поточну інформацію про книги в бібліотеці.

Відомості про книги включають:

- номер УДК;
- прізвище та ініціали автора;
- назва;
- рік видання;
- кількість примірників цієї книги в бібліотеці.

Програма повинна забезпечувати:

- початкове формування даних про всі книги в бібліотеці у вигляді списку;
- додавання даних про книги, тих, хто надійшов до бібліотеки;
- видалення даних про книги, що списуються;
- на вимогу видаються відомості про наявність книг в бібліотеці, впорядковані по роках видання.

Варіант 17

На міжміської телефонної станції картотека абонентів, що містить відомості про телефонах і їх власників, організована у вигляді лінійного списку. Написати програму, яка

- забезпечує початкове формування картотеки у вигляді лінійного списку;
- забезпечує виведення всієї картотеки;
- вводить номер телефону і час розмови;
- виводить повідомлення на оплату телефонної розмови.

Програма повинна забезпечувати діалог за допомогою меню і контроль помилок введення.

Варіант 18

На міжміської телефонної станції картотека абонентів, що містить відомості про телефонах і їх власників, організована у вигляді двійкового дерева. Написати програму, яка

- забезпечує початкове формування картотеки у вигляді двійкового дерева; виробляє висновок всієї картотеки;
- вводить номер телефону і час розмови;
- виводить повідомлення на оплату телефонної розмови.

Програма повинна забезпечувати діалог за допомогою меню і контроль помилок введення.

Варіант 19

Автоматизована інформаційна система на залізничному вокзалі містить відомості про відправлення поїздів. Для кожного поїзда вказується номер поїзда; станція призначення; час відправлення. Дані в інформаційну систему організовані у вигляді лінійного списку. Написати програму, яка

- забезпечує початковий введення даних в інформаційну систему і формування лінійного списку;
- забезпечує висновок всього списку;

- вводить номер поїзда і виводить всі дані про цей потяг;
- вводить назву станції призначення і виводить дані про всі поїздах, що прямують до цієї станції.

Програма повинна забезпечувати діалог за допомогою меню і контроль помилок введення.

Варіант 20.

Автоматизована інформаційна система на залізничному вокзалі містить відомості про відправлення поїздів. Для кожного поїзда вказується номер поїзда; станція призначення; час відправлення. Дані в інформаційну систему організовані у вигляді двійкового дерева. Написати програму, яка

- забезпечує початковий введення даних в інформаційну систему і формування двійкового дерева;
- забезпечує висновок всього дерева;
- вводить номер поїзда і виводить всі дані про цей потяг;
- вводить назву станції призначення і виводить дані про всіх поїздах, що прямують до цієї станції.

Програма повинна забезпечувати діалог за допомогою меню і контроль помилок введення.

7.4. Зміст звіту

Звіт повинен містити:

1. Титульний аркуш.
2. Лістинги програм, що вирішують завдання.
3. При необхідності навести екрані форми, або консольний виведення результатів роботи програми.
4. Розширені висновки з лабораторної роботи.

Лабораторна робота 8. Використання чисельних бібліотек мови C/C++ для вирішення практичних задач.

8.1. Мета роботи

Навчитися застосовувати чисельні бібліотеки мови C/C++ для реалізації програмних продуктів, що оброблюють чисельні дані. Навчитися реалізовувати алгоритми обробки чисельної інформації.

8.2. Теоретичний матеріал

Необхідно вивчити матеріали лекції 15 [15].

Виділемо такі ключові моменти, що стосуються використання числових бібліотек мови C/C++, які є ключовими при вирішенні практичних задач:

1. Компоненти, які програма C++ може використовувати для виконання напівчислових операцій.

- `<complex>`. Заголовок `<complex>` визначає шаблонний клас, і набір функцій для роботи з комплексними числами.
- `<random>`. Можливість генерувати (псевдо-)випадкові числа
- `<valarray>` Визначає п'ять шаблонів класів (`valarray`, `slice_array`, `gslice_array`, `mask_array`, і `indirect_array`), два класи (`slice` і `gslice`), і множину пов'язаних з ними шаблонів функцій для представлення і маніпуляції над масивами значень.
- `<numeric>`. Узагальнені числові операції.

2. Цілочисельні типи (`int`, `long` та `in`.) являють собою лише наближення математичного поняття цілих чисел, а числа з плаваючою точкою (`float`, `double` і `in`.) є лише наближенням математичного поняття дійсних чисел. Звідси випливає, що з математичної точки зору деякі обчислення виявляються неточними або неправильними.

3. Вплив фіксованого розміру цілих чисел може проявитися більш вражаюче. Справа в тому, що числа з плаваючою точкою за визначенням є наближеннями дійсних чисел, тому вони можуть втрачати точність (тобто втрачати наймолодші значущі біти). Цілі ж числа часто переповнюються (тобто втрачають найстарші значущі біти). В результаті помилки, пов'язані з числами з плаваючою точкою, мають більш складний характер (які новачки часто не помічають), а помилки, що пов'язані з цілими числами, кидаються в очі.

8.3. Порядок виконання роботи

Оберіть варіант завдання у кожному блоці відповідно до порядкового номеру журналу, що наведені нижче. Реалізувати у вигляді програмного коду та результати занести до звіту.

8.3.1. Блок завдань 1

Варіант 1

Виведіть на екран розміри типів char, short, int, long, float, double, int * і double * (використовуйте оператор sizeof, а не заголовки <limits>).

Варіант 2

Використовуючи оператор sizeof, виведіть на екран розміри об'єктів:

```
Matrix <int> a (10),  
Matrix <int> b (10),  
Matrix <double> c (10),  
Matrix <int, 2> d (10,10),  
Matrix <int, 3> e (10,10,10).
```

Виведіть на друк кількість елементів в кожному з об'єктів.

Варіант 3

Напишіть програму, яка вводить числа типу int і з потоку cin і виводить результат застосування функції sqrt() до кожного з цих чисел int або повідомлення «квадратний корінь не існує», якщо функцію sqrt(x) не можна застосовувати до деяких значень x (тобто перевіряйте значення, що повертаються функцією sqrt ()).

Варіант 4

Вважайте десять чисел з плаваючою точкою з потоку введення і запишіть їх у матрицю Matrix<double>. Клас Matrix не має функції push_back (), тому будьте обережні і передбачте реакцію на спробу ввести невірне кількість чисел типу double. Виведіть цю матрицю на екран.

Варіант 5

Обчисліть таблицю множення $[0, n) * [0, m)$ і уявіть її у вигляді двовимірної матриці Matrix. Вводите числа n і m з потоку cin і акуратно виводьте на екран отриману таблицю (передбачається, що число m досить мало, щоб результати помістилися в одному рядку).

Варіант 6

Введіть з потоку cin десять об'єктів класу complex <double> (клас cin підтримує оператор >> для типу complex) і помістіть їх в матрицю. Обчисліть і виведіть на екран суму десяти комплексних чисел.

Варіант 7

Зчитати шість чисел типу int в матрицю Matrix <int, 2> m (2,3) і виведіть їх на екран.

Варіант 8.

Аргументи функції f в виразах `a.apply(f)` і `apply(f,a)` є різними. Напишіть функцію `triple()` для кожного варіанту і застосуєте їх для потроєння елементів масиву `{1 2 3 4 5}`. Визначте єдину функцію `triple()`, яку можна було б використовувати як в вираженні `a.apply(triple)`, так і в вираженні `apply(triple,a)`. Поясніть, чому писати для використання в якості аргументу функції `apply()` всі функції саме таким чином - погана ідея.

Варіант 9.

Виконайте Варіант 8, використовуючи не функції, а функціональні об'єкти. Вказівка: приклади можна знайти в заголовки `Matrix.h`.

Варіант 10.

Напишіть функцію `apply(f,a)`, що приймає в якості аргументу функції `void(T&)`, `T(const&)`, а також еквівалентні їм функціональні об'єкти. Вказівка: див. `boost::bind`.

Варіант 11.

Виконайте програму методу виключення Гауса, скомпілюйте і протестуйте на простому прикладі.

Варіант 12.

Застосуйте програму методу виключення Гауса до системи $A = \begin{Bmatrix} 0 & 1 \\ 1 & 0 \end{Bmatrix}$ і $b = \begin{Bmatrix} 5 \\ 6 \end{Bmatrix}$ і переконайтеся, що програма не в змозі знайти рішення. Потім спробуйте застосувати в ній функцію `elim_with_partial_pivot()`.

Варіант 13.

Замініть циклами векторні операції `dotp_roduct()` і `scale_and_add()` в програмі методу виключення Гауса. Протестуйте і прокоментуйте цю програму.

8.3.2. Блок завдань 2

Варіант 1

Вивчити алгоритм і реалізувати метод послідовних наближень для розв'язання алгебраїчних рівнянь.

Варіант 2

Вивчити алгоритм і реалізувати метод половинного ділення для розв'язання алгебраїчних рівнянь.

Варіант 3

Вивчити алгоритм і реалізувати метод дотичних для вирішення алгебраїчних рівнянь.

Варіант 4

Вивчити алгоритм і реалізувати інтерполяційний поліном Лагранжа.

Варіант 5.

Вивчити алгоритм і реалізувати інтерполяційний поліном Ньютона.

Варіант 6.

Вивчити алгоритм і реалізувати метод обчислення інтегралів методом Сімпсона.

Варіант 7.

Вивчити алгоритм і реалізувати метод обчислення інтегралів методом Монте-Карло.

8.4. Зміст звіту

Звіт повинен містити:

1. Титульний аркуш.
2. Лістинги програм, що вирішують завдання.
3. При необхідності навести екрані форми, або консольний виведення результатів роботи програми.
4. Розширені висновки з лабораторної роботи.

Література

1. Пелешко Д.Д., Теслюк В.М. Об'єктні технології C++11: навч. Посібник. – Львів: Видавництво Львівської політехніки, 2013. – 360 с.
2. Страуструп, Б. Программирование: принципы и практика с использованием C++ (C++11 и C++14), 2-е изд.: Пер. с англ. - М.: ООО "И.Д. Вильямс", 2016. - 1328 с.
3. Павловская, Т.А. C/C++. Программирование на языке высокого уровня. — СПб.: Питер, 2003. - 461 с.
4. Павловская, Т.А., Щупак, Ю.А. C/C++. Структурное и объектно-ориентированное программирование: Практикум. — СПб.: Питер, 2011. - 352 с.
5. A Repository for Libraries Meant to Work Well with the C++ Standard Library. Електронний ресурс: www.boost.org.
6. ISO International Standard ISO/IEC 14882:2014(E) – Programming Language C++ Електронний ресурс: <https://isocpp.org/std/the-standard>
7. ISO /IEC 9899:2011. Programming Languages - C. The C standard.
8. ISO/IEC 1 4882:2011. Programming Languages - C++. The C++ standard.
9. Кёниг, Э., Му, Б.Э. Эффективное программирование на C++. Серия C++ In-Depth, т.2.: Пер. с англ. — М.: Издательский дом "Вильямс", 2002. — 384 с.
10. Саттер, Г. Решение сложных задач на C++. Серия C++ In-Depth: Пер. с англ. — М.: Издательский дом "Вильямс", 2008. - 400 с.
11. Страуструп, Б. Дизайн и эволюция C++: Пер. с англ. - М.: ДМК Пресс. - 448 с.
12. Уоррен, Г.С. Алгоритмические трюки для программистов. – 2-е изд.: Пер. с англ. – М.: Издательский дом "Вильямс", 2014. – 512 с.
13. Конспект лекцій з дисципліни «Вступ до програмування ч.2». ДВНЗ «Комп'ютерна академія шаг», Львів, 2017.