

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 4

На тему: «Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання.
Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки
та робота з масивами та структурами.

з дисципліни: «Основи програмування»

до:

Практичних Робіт до блоку № 4

Виконав:

Студент групи ШІ-13

Матрунич Олександр Іванович

Тема роботи:

Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.

Мета роботи: вивчення пам'яті в C++, роботи з масивами, вказівниками, посиланнями та динамічним виділенням пам'яті, а також ознайомлення зі структурами даних і перевантаженням операторів для ефективного управління пам'яттю та обробки даних.

Теоретичні відомості:

1. Класи пам'яті у C++

- Статична пам'ять.
- Динамічна пам'ять.
- Поняття стеку.
- Виділення та вивільнення пам'яті.

2. Вступ до Масивів і Вказівників:

- Основи масивів: визначення, важливість, приклади використання.
- Різниця між статичними та динамічними масивами.
- Основи вказівників: що це таке, як вони працюють.
- Взаємозв'язок між масивами та вказівниками.
- Вступ до посилань: основні концепції та відмінності від вказівників.

3. Одновимірні Масиви:

- Створення та ініціалізація одновимірних масивів.
- Основні операції: індексація, присвоєння, читання.
- Цикли та обхід масивів.
- Використання функцій для роботи з масивами.
- Приклади алгоритмів сортування та пошуку.

4. Вказівники та Посилання:

- Використання вказівників для доступу до елементів масиву.
- Арифметика вказівників.
- Різниця між вказівниками та посиланнями в контексті функцій.
- Динамічне виділення пам'яті з використанням вказівників.
- Використання вказівників для створення складних структур даних.

5. Двовимірні Масиви:

- Оголошення та ініціалізація двовимірних масивів.
- Вкладені цикли для обходу двовимірних масивів.
- Практичні приклади використання двовимірних масивів.
- Передача двовимірних масивів у функції.
- Застосування двовимірних масивів для розв'язання задач.

6. Динамічні Масиви:

- Основи динамічного виділення пам'яті.
- Створення та управління динамічними масивами.
- Використання операторів `new` та `delete` для управління пам'яттю.
- Реалізація змінної розмірності масивів.
- Передача динамічних масивів у функції.

7. Структури Даних:

- Оголошення та використання структур.
- Використання масивів та вказівників у структурах.
- Функції для обробки даних у структурах.
- Використання структур для представлення складних даних.
- Вкладені структури та їх використання.
- Об'єднання (Union)
- Переліки (enumerations)

8. Вкладені Структури:

- Поняття вкладених структур та їх оголошення.
- Взаємодія з вкладеними структурами.
- Використання вкладених структур для моделювання складних даних.
- Передача вкладених структур у функції.
- Приклади реального використання вкладених структур.

9. Використання структур

- Перевантаження операторів у структурі.
- Вивід/ввід структури (operator<<,&&);
- Арифметичні операції з структурами (operator+, operator-);
- Практичні задачі на виведення структур та операції з ними

10. Алгоритми обробки та робота з масивами та структурами:

- Алгоритми пошуку та сортування в масивах.
- Обробка та маніпуляції з даними у структурах.
- Використання циклів та умовних операторів для роботи з масивами та структурами.
- Інтеграція масивів та структур у алгоритми.
- Розв'язання практичних задач з використанням масивів та структур.

Індивідуальний план опрацювання теорії:

- Класи пам'яті у C++
- Вступ до Масивів і Вказівників
- Одновимірні Масиви
- Вказівники та Посилання
- Двовимірні Масиви
- Динамічні Масиви
- Структури Даних

Джерела:

- Лекції О. Пшеничного
- Практичні М. Фаріон

- Chat gpt
- Список відтворення на YouTube (<https://youtube.com/playlist?list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&si=sXvmPdnGkwvJLXUi>)
- Власний досвід

Виконання роботи:

1) Опрацювання завдання та вимог до програми та середовища

Завдання №1 - VNS Lab 4 Task 1 (6)

- 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити елемент із заданим номером.
- 4) Додати після першого парного елемента масиву елемент зі значенням $M[I-1]+2$.
- 5) Роздрукувати отриманий масив.

Завдання №2 - VNS Lab 5 Task 1 (6)

Елемент матриці є сідловою точкою, якщо він є найменшим у своєму рядку й найбільшим у своєму стовпці (або навпаки: найбільшим у своєму рядку й найменшим у своєму стовпці). Для заданої матриці визначити всі сідлові точки.

Завдання №3 - Algotester Lab 2 (v2)

У вас є масив r розміром N . Також вам дано 3 цілих числа.

Спочатку ви маєте видалити з масиву ці 3 числа, які вам дані. Після цього перетворити цей

масив у масив сум, розміром $N_{\text{new}} - 1$ (розмір нового масиву після видалення елементів), який

буде відображати суми сусідніх елементів нового масиву.

Далі необхідно вивести масив сум на екран.

Завдання №4 - Algotester Lab 3 (v3)

Вам дана стрічка *s*.

Ваше завдання зробити компресію стрічки, тобто якщо якась буква йде більше одного разу

підряд у стрічці замінити її на букву + кількість входжень підряд.

Завдання №5 - Class Practice Work – Перевірка чи слово або число є паліндромом

Задача

Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.

Мета задачі

Навчитися користуватися механізмами перевантаження функції та використовувати рекурсію для вирішення задач обчислення.

Вимоги:

1. Визначення функції:

Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.

2. Приклад визначення функції:

bool isPalindrome(const string& str, int start, int end);

3. Перевантаження функцій:

Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.

bool isPalindrome(ціле число);

4. Рекурсія:

Рекурсивна функція для рядків перевірить символи в поточній початковій і кінцевій позиціях. Якщо вони збігаються, він буде рекурсивно перевіряти наступні позиції, поки початок не перевищить кінець, після чого рядок буде визначено як паліндром.

Кроки реалізації

- Визначте та реалізуйте рекурсивну функцію *isPalindrome* для рядків.
- Визначте та реалізуйте перевантажену функцію *isPalindrome* для цілих чисел. Використати математичний підхід щоб перевірити чи число є паліндромом.

Завдання №6 - Self Practice - Task Lab 4v2:

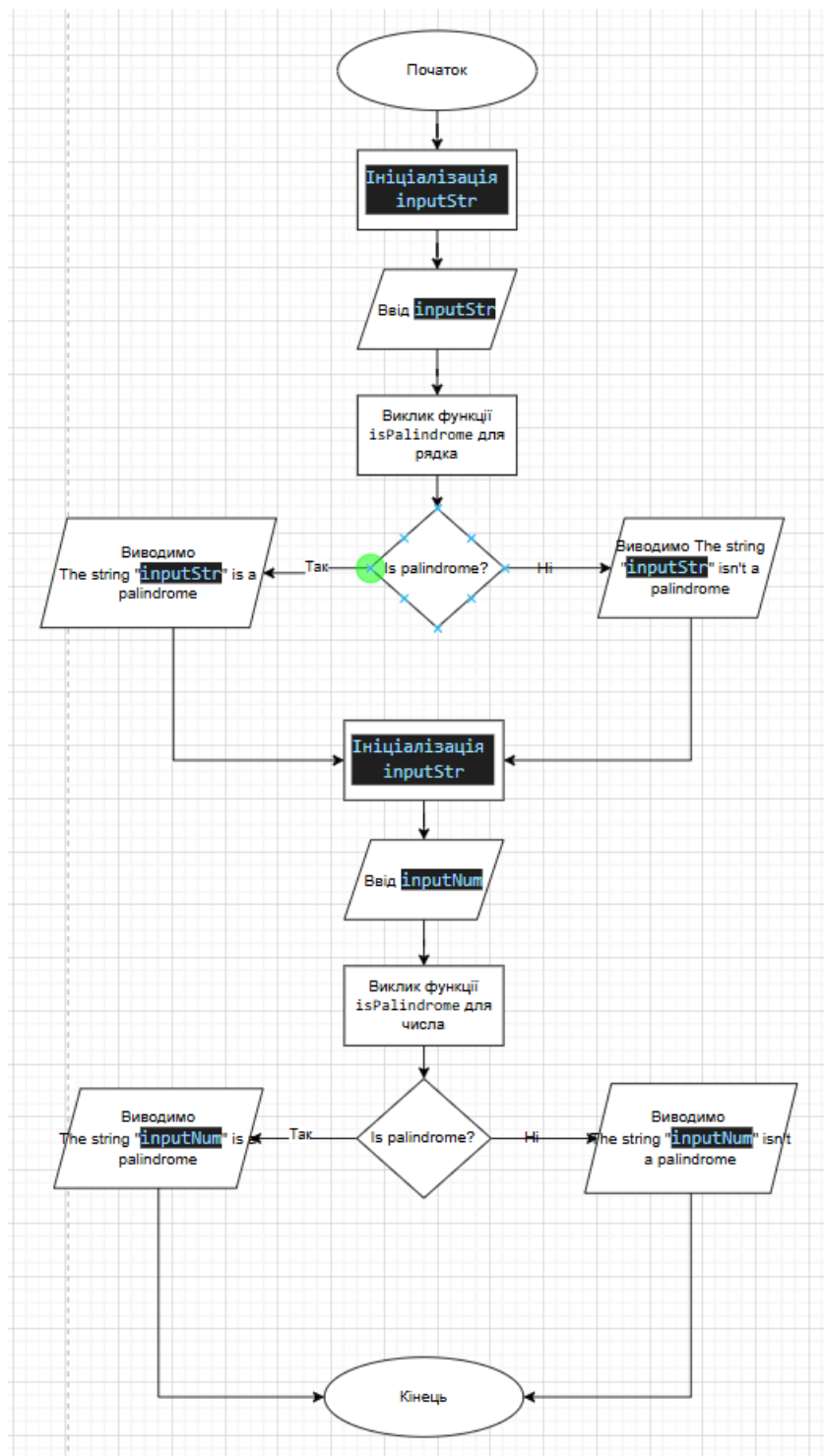
Вам дано масив aa з NN цілих чисел.

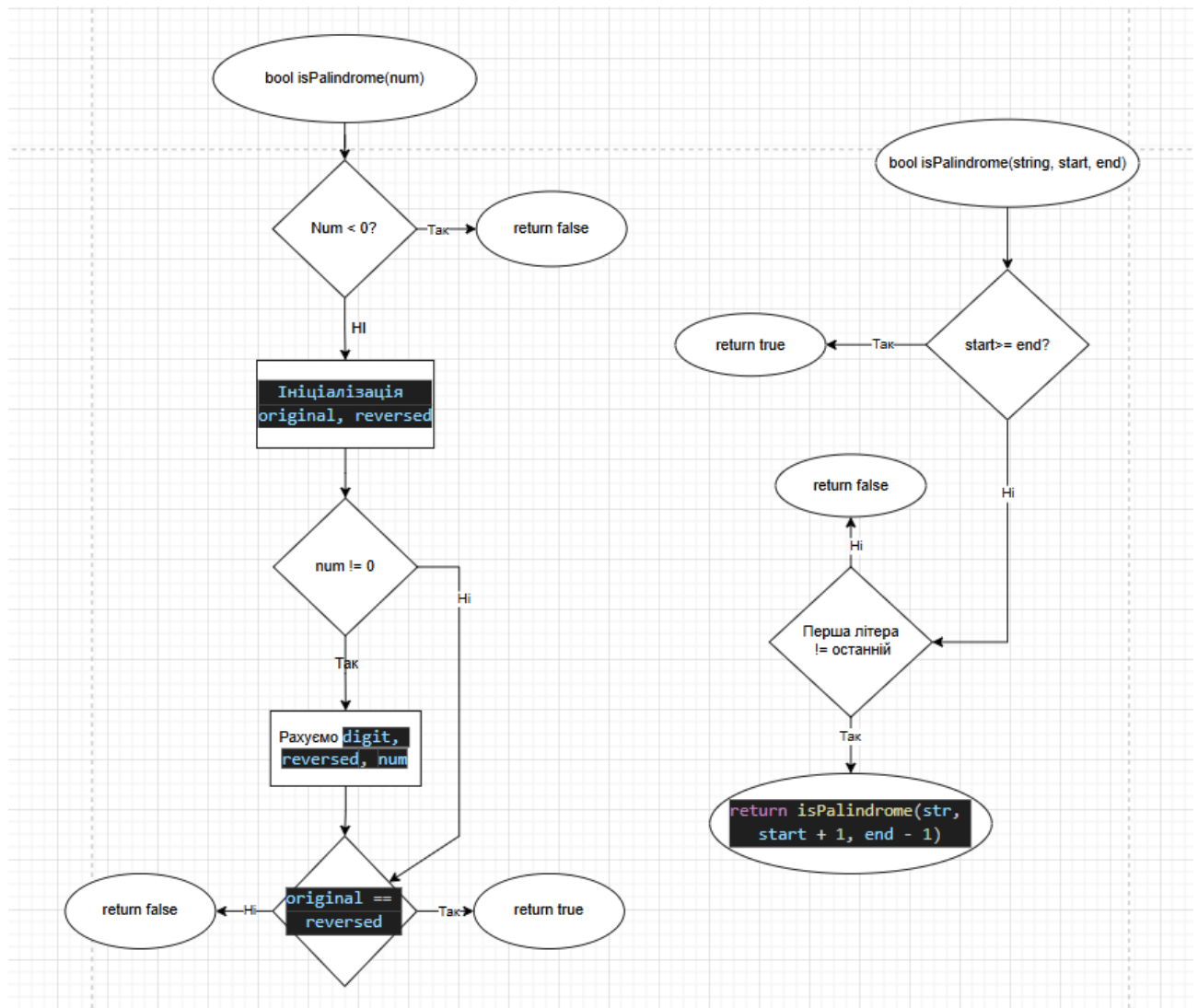
Спочатку видаліть масиву aa усі елементи що повторюються, наприклад масив $[1, 3, 3, 4]$ має перетворитися у $[1, 3, 4]$.

Після цього оберніть посортовану версію масиву aa на KK , тобто при $K=3$ масив $[1, 2, 3, 4, 5, 6, 7]$ перетвориться на $[4, 5, 6, 7, 1, 2, 3]$.

Виведіть результат.

2) Дизайн

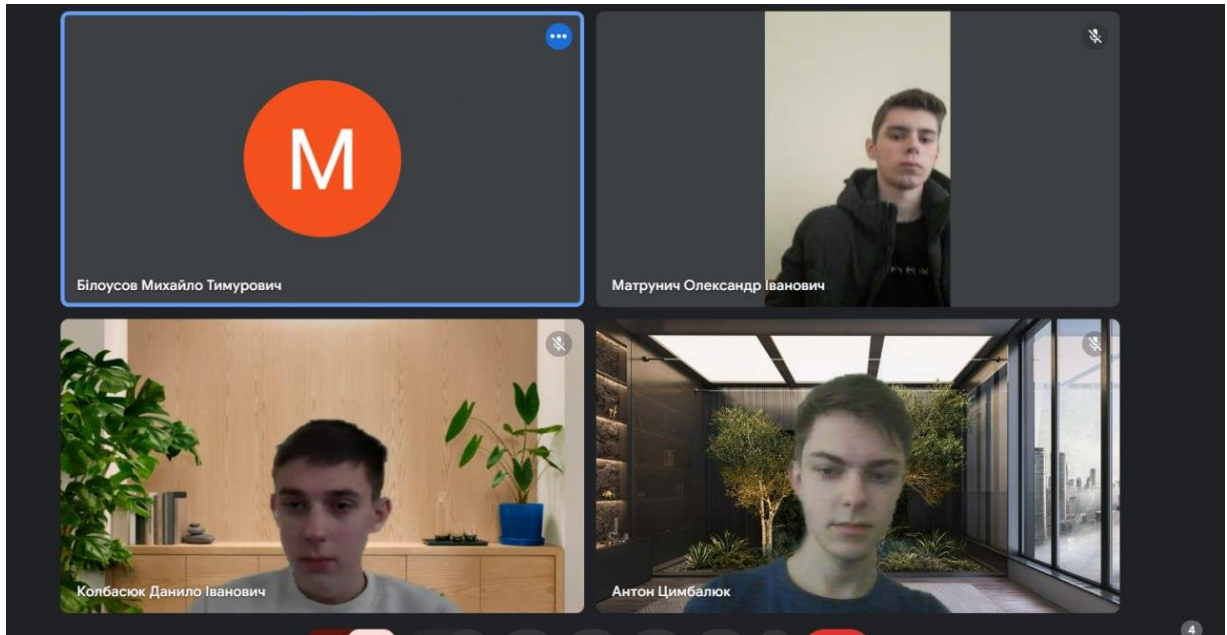




Завдання №5 - Class Practice Work – Перевірка чи слово або число є паліндромом

3) Конфігурація середовища до виконання завдань:

Зустріч з командою



4) Код програми з посиланням на зовнішні ресурси

До завдання №1 код `vns_lab_4_task_1_variant_6_oleksandr_matrunych.cpp`

До завдання №2 код `vns_lab_5_task_1_variant_6_oleksandr_matrunych.cpp`

До завдання №3 код `algotester_lab_2_task_2_oleksandr_matrunych.cpp`

До завдання №4 код `algotester_lab_3_task_3_oleksandr_matrunych.cpp`

До завдання №5 код `practice_work_oleksandr_matrunych.cpp`

До завдання №6 код `practice_work_self_oleksandr_matrunych.cpp`

5) Результати виконаних завдань, тестування та фактично затрачений час

Завдання №1 - VNS Lab 4 Task 1 (6)

```
Enter the size of the array: 10
Initial array: 32 15 5 47 60 88 24 88 85 33
Enter the index of the element to delete (0 - 9): 9
Resulting array: 32 87 15 5 47 60 88 24 88 85
```

Час виконання: 100 хв

Завдання №2 - VNS Lab 5 Task 1 (6)

```
Enter the number of rows (n): 3
Enter the number of columns (m): 3
Enter the elements of the matrix:
1 2 3
5 6 7
8 9 10
Saddle points are:
Saddle point at (0, 2) with value 3
Saddle point at (2, 0) with value 8
```

Час виконання: 120 хв

Завдання №3 - Algotester Lab 2 (v2)

```
6
1 2 3 4 5 7
4 5 6
3
3 5 10
```

Час виконання: 80 хв

декілька секунд тому	C++ 23	Зараховано	0.003	1.328	Перегляд
----------------------	--------	------------	-------	-------	--------------------------

Завдання №4 - Algotester Lab 3 (v3)

```
AAAABVBCQQQQ
A4V3CQ4
```

Час виконання: 100 хв

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.003	1.180	Перегляд

Завдання №5 - Class Practice Work – Перевірка чи слово або число є паліндромом

```
Enter a string to check: level
The string "level" is a palindrome
Enter a number to check: 12321
The number 12321 is a palindrome.
```

```
Enter a string to check: helpme
The string "helpme" is not a palindrome.
Enter a number to check: 103
The number 103 is not a palindrome.
```

Час виконання: 1.5 год

Завдання №6 - Self Practice – Lab 4v2

```
10 3
1 2 2 3 3 3 4 5 6 7
7
4 5 6 7 1 2 3
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.003	1.199	Перегляд
хвилину тому	C++ 23	Зараховано	0.003	1.191	Перегляд

Час виконання: 140 хв

Висновок: на лабораторній роботі вивчав пам'ять в C++, роботи з масивами, вказівниками, посиланнями та динамічним виділенням пам'яті, а також ознайомився зі структурами даних і перевантаженням операторів для ефективного управління пам'яттю та обробки даних.