

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



## **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:

ВНС Лабораторної Роботи №4  
ВНС Лабораторної Роботи №5  
Алготестер Лабораторної Роботи №2  
Алготестер Лабораторної Роботи №3  
Практичних Робіт до блоку №4

**Виконала:**

Студентка групи ШІ-11

Гуменюк Анастасія Олександрівна

**Тема роботи:** Ознайомлення з одновимірними та двовимірними масивами, їх оголошенням, ініціалізацією та використанням у програмах C++. Вивчення взаємодії між масивами та вказівниками, особливостей їх використання в контексті функцій, а також принципів роботи з посиланнями. Розгляд різниці між статичними та динамічними масивами, а також алгоритмів роботи з динамічною пам'яттю, використання операторів new та delete. Вивчення структур даних, включаючи вкладені структури та об'єднання (union), їх оголошення та практичне застосування. Аналіз алгоритмів обробки масивів і структур, їх інтеграції в більш складні алгоритми та застосування у вирішенні різних завдань.

**Мета роботи:** Опанувати методи створення та ініціалізації одновимірних і двовимірних масивів, виконання основних операцій, таких як індексація, обхід і маніпуляції з даними за допомогою циклів та функцій. Навчитися використовувати вказівники для доступу до елементів масивів, розуміти їхню арифметику та зв'язок з посиланнями. Дослідити різницю між вказівниками та посиланнями, особливо в контексті передачі параметрів у функції. Засвоїти динамічне виділення пам'яті та управління нею за допомогою операторів new і delete, створювати та використовувати динамічні масиви. Зрозуміти основи роботи зі структурами даних, навчитися оголошувати та використовувати вкладені структури та об'єднання, моделювати складні об'єкти. Ознайомитися з практичними прикладами алгоритмів пошуку та сортування в масивах, навчитися їх застосовувати для обробки даних у різних структурах. Розвинути вміння створювати, аналізувати та оптимізувати алгоритми для роботи з масивами та структурами даних у контексті вирішення практичних задач.

### **Теоретичні відомості:**

Теоретичні відомості з переліком важливих тем:

- Тема №1: Класи пам'яті в C++.
- Тема №2: Вступ до Масивів і Вказівників.
- Тема №3: Вказівники та Посилання.
- Тема №4: Динамічні Масиви.
- Тема №5: Структури Даних.
- Тема №6: Вкладені Структури.
- Тема №7: Алгоритми обробки та робота з Масивами та структурами.

## Індивідуальний план опрацювання теорії:

### Тема №1: Класи пам'яті в C++.

- Джерела:  
<https://acode.com.ua/urok-89-dynamichne-vydilennya-pam-yati/>  
<https://www.youtube.com/watch?v=NyOjKd5Qruk>
- Що опрацьовано:
  - Статична пам'ять.
  - Динамічна пам'ять.
  - Поняття стеку.
  - Виділення та вивільнення пам'яті.
- Статус: Ознайомлена
- Початок опрацювання теми: 7.11.2024.
- Звершення опрацювання теми: 7.11.2024 (35хв.).

### Тема №2: Вступ до Масивів і Вказівників.

- Джерела:  
<https://studfile.net/preview/5319824/page:10/>  
<https://www.youtube.com/watch?v=zopWRIYOXWw>
- Що опрацьовано:
  - 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
- Статус: Ознайомлена
- Початок опрацювання теми: 7.11.2024.
- Звершення опрацювання теми: 7.11.2024 (2 год.).

### Тема №3: Вказівники та Посилання.

- Джерела:  
<https://acode.com.ua/urok-84-vkazivnyky/>
- Що опрацьовано:
  - Використання вказівників для доступу до елементів масиву.
  - Арифметика вказівників.
  - Різниця між вказівниками та посиланнями в контексті функцій.

- Динамічне виділення пам'яті з використанням вказівників.
- Статус: Ознайомлена
- Початок опрацювання теми: 7.11.2024.
- Звершення опрацювання теми: 7.11.2024 (20хв.).

#### Тема №4: Динамічні Масиви.

- Джерела:
  - <https://www.youtube.com/watch?v=OGR9VJEh8Hk&t=553s>
- Що опрацьовано:
  - 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.
- Статус: Ознайомлена
- Початок опрацювання теми: 7.11.2024.
- Звершення опрацювання теми: 7.11.2024 (40хв.).

#### Тема №5: Структури Даних.

- Джерела:
  - <https://acode.com.ua/urok-64-struktury/>
  - [https://uk.wikibooks.org/wiki/C%2B%2B/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%B8/%D0%9E%D0%B1%27%D1%94%D0%B4%D0%BD%D0%B0%D0%BD%D0%BD%D1%8F\\_\(union\)](https://uk.wikibooks.org/wiki/C%2B%2B/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%B8/%D0%9E%D0%B1%27%D1%94%D0%B4%D0%BD%D0%B0%D0%BD%D0%BD%D1%8F_(union))
- Що опрацьовано:
  - 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)
- Статус: Ознайомлена
- Початок опрацювання теми: 7.11.2024.
- Звершення опрацювання теми: 7.11.2024 (30хв.).

#### Тема №6: Вкладені Структури.

- Джерела:

[https://www.youtube.com/watch?v=999IE-6b7\\_s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=64](https://www.youtube.com/watch?v=999IE-6b7_s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=64)

- Що опрацьовано:
  - 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.
- Статус: Ознайомлена
- Початок опрацювання теми: 7.11.2024.
- Звершення опрацювання теми: 7.11.2024 (30хв.).

Тема №7: Алгоритми обробки та робота з Масивами та структурами.

- Джерела:
  - <https://youtu.be/gncUL57AHwk?si=ghKiPwoPKUnAnJxZ>
  - <https://youtu.be/maB87eyn7h8?si=F3sXb8FzbExNl0Re>
  - <https://youtu.be/uQxG9gBROog?si=IGJUmHZ9fXluWVQh>
  - [https://youtu.be/YFLRN\\_Gmh4o?si=-88LKx2ta332u3ms](https://youtu.be/YFLRN_Gmh4o?si=-88LKx2ta332u3ms)
- Що опрацьовано:
  - 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.
- Статус: Ознайомлена
- Початок опрацювання теми: 7.11.2024.
- Звершення опрацювання теми: 7.11.2024 (30хв.).

### **Виконання роботи:**

*1) Опрацювання завдання та вимог до програми та середовища*

#### **Завдання №1 VNS Lab 4 - Task 1-10**

- 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити 5 перших елементів масиву.

4) Додати в кінець масиву 3 нових елементи.

5) Роздрукувати отриманий масив

**Методичні вказівки:** При виконанні роботи використовуються статичні масиви. Для організації статичних масивів із псевдозмінними межами необхідно оголосити масив досить великої довжини, наприклад, 100 елементів:

```
int N=100;
```

```
int a[N];
```

Потім користувач вводить реальну довжину масиву (не більше N) і працює з масивом тієї довжини, що він сам вказав. Інші елементи (хоча пам'ять під них і буде виділена) не розглядаються.

При зменшенні або збільшенні довжини масиву необхідно змінювати його реальну довжину.

### **Завдання №2 VNS Lab 5 - Task 1-10**

Написати функцію, що перевіряє чи є від'ємні елементи в зазначеному рядку двовимірного масиву. Знищити з масиву всі рядки з від'ємними елементами, знищений рядок заповнюється 0 і переноситься в кінець масиву.

### **Завдання №3 Algotester Lab 2 – v3**

Вам дано масив цілих чисел розміром NN, на першій та останній клітинці розміщено по дрону.

Вони одночасно взлітають.

На початку кожного ходу швидкість дрону стає рівною значенню клітинки, у якій він знаходиться.

Тобто лівий дрон у першу секунду з клітинки з індексом 11 перелетить у клітинку з індексом a1a1, тобто його наступна позиція рахується як поточна позиція + число у поточній позиції (перегляньте пояснення для візуалізації) Правий робить аналогічно в протилежну сторону.

Вони це роблять до моменту, коли трапиться одна з зазначених подій:

Якщо 2 дрони опиняються в одній клітинці - ви виводите Collision.

Якщо лівий дрон опиниться справа від правого - це Miss

У випадку якщо вони зупиняться один навпроти одного, тобто у клітинках  $a_i$  та  $a_{i+1}$  - виведіть Stopped

Врахуйте, що перевіряти треба також до взльоту.

Input

У першому рядку ціле число  $NN$  - розмір масиву

У другому рядку  $NN$  цілих чисел - елементи масиву

Output

У першому рядку фінальна позиція першого та другого дрона.

У другому рядку одне зі слів:

Collision

Miss

Stopped

#### **Завдання №4 Algotester Lab 3 – v2**

Вам дано 2 масиви розміром  $N$  та  $M$ . Значення у цих масивах унікальні.

Ваше завдання вивести у першому рядку кількість елементів, які наявні в обох масивах одночасно, у другому кількість унікальних елементів в обох масивах разом.

**Input**

У першому рядку ціле число  $NN$

у другому рядку  $NN$  цілих чисел  $a_1..a_n$

У третьому рядку ціле число  $MM$

у четвертому рядку  $MM$  цілих чисел  $b_1..b_n$

**Output**

У першому рядку одне ціле число - кількість елементів, які наявні в обох масивах одночасно.

У другому рядку кількість унікальних елементів в обох масивах (тобто кількість унікальних елементів у масиві, який буде об'єднанням двох даних).

**Notes**

Користуватися `std::set` та `std::map` та їх похідними ЗАБОРОНЕНО.

## Завдання №5 Class Practice Work

Перевірка чи слово або число є паліндромом

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

Паліндром — це послідовність символів, яка однаково читається вперед і назад (наприклад, «radar», «level», «12321»).

Мета Задачі

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

Вимоги:

1. Визначення функції:
  - a. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:
  - a. *bool isPalindrome(const string& str, int start, int end);*
3. Перевантаження функцій:
  - a. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
  - b. *bool isPalindrome(ціле число);*
4. Рекурсія:
  - a. 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 Work



Розробити програму, яка зберігає та обробляє інформацію про людей.  
Кожна людина має ім'я, вік та місто проживання.

Спочатку програма запитує кількість людей, а потім послідовно вводить для кожної з них ім'я, вік та місто. Після цього програма виводить список людей у порядку введення, сортує його за віком у порядку зростання та знову виводить, але вже відсортований список.

Вхідні дані

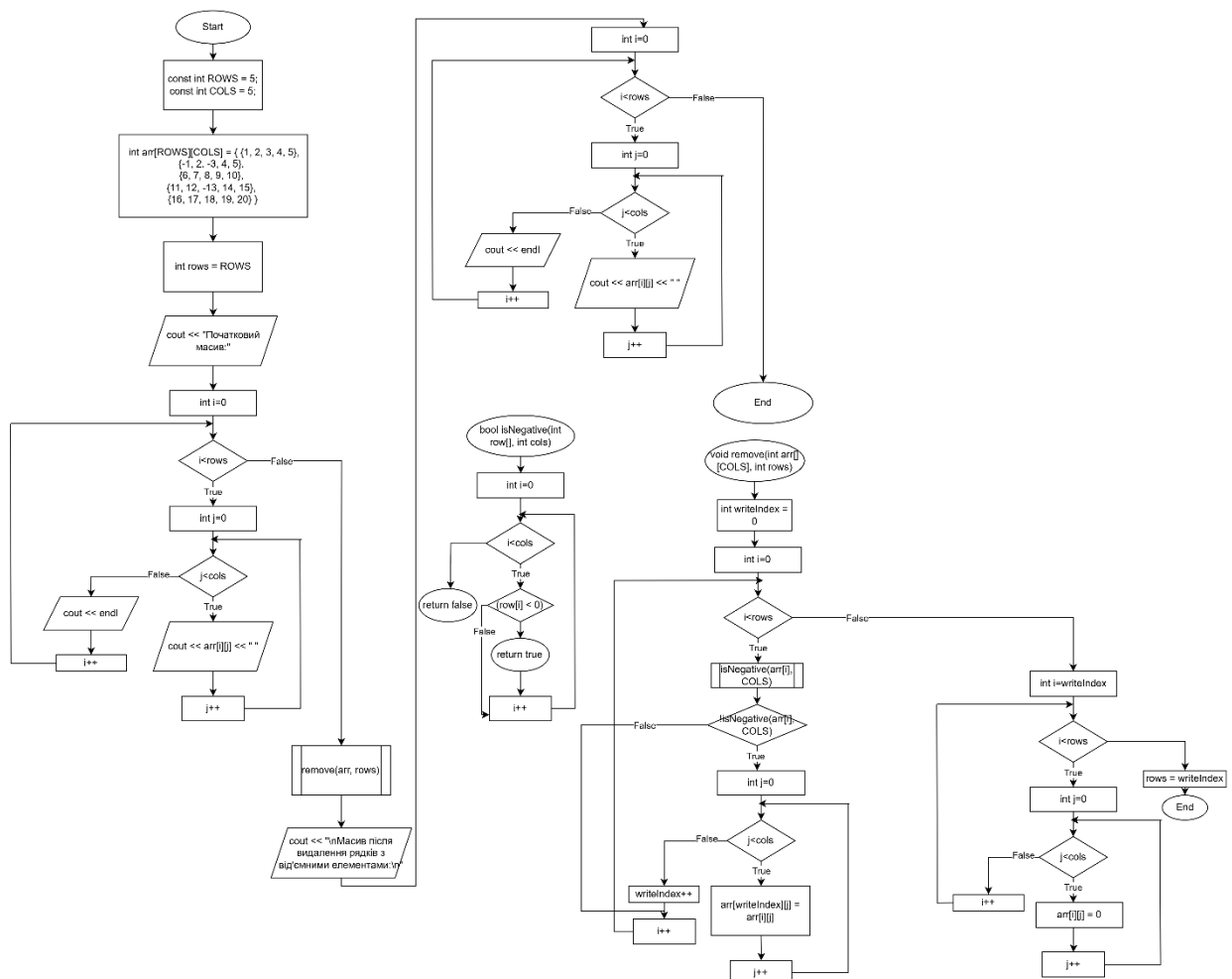
Число N — кількість людей, а потім N рядків із даними про кожну людину: ім'я, вік, місто.

Вихідні дані

Список людей до та після сортування за віком.

2) Дизайн виконання завдань:

## Завдання №2 VNS Lab 5 - Task 1-10



3) Код програм з посиланням на зовнішні ресурси

## Завдання №1 VNS Lab 4 - Task 1-10

```

#include <iostream>
#include <cstdlib>
using namespace std;

int main() {

    int a[100];
    int size;
    cout << "Введіть довжину масиву : ";
    cin >> size;

    for (int i = 0; i < size; i++) {
        a[i] = rand() % 100;
    }

    cout << "Початковий масив: ";
    for (int i = 0; i < size; i++) {
        cout << a[i] << " ";
    }
    cout << endl;

    if (size > 5) {
        size -= 5;
        for (int i = 0; i < size; i++) {
            a[i] = a[i + 5];
        }
    } else {
        cout << "Масив занадто малий для видалення 5 елементів." << endl;
    }

    cout << "Введіть 3 нових числа для додавання в кінець масиву: ";
    for (int i = 0; i < 3; i++) {
        cin >> a[size + i];
    }
    size += 3;

    cout << "Кінцевий масив: ";
    for (int i = 0; i < size; i++) {
        cout << a[i] << " ";
    }
    cout << endl;

    return 0;
}

```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/blob/epic\\_4\\_practice\\_and\\_labs\\_anastasiia\\_humeniuk/ai\\_11/anastasiia\\_humeniuk/epic4/vns\\_lab\\_4\\_task\\_1\\_variant\\_1\\_anastasiia\\_humeniuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_4_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic4/vns_lab_4_task_1_variant_1_anastasiia_humeniuk.cpp)

## Завдання №2 VNS Lab 5 - Task 1-10

```
#include <iostream>
using namespace std;

const int ROWS = 5;
const int COLS = 5;

bool isNegative(int row[], int cols) {
    for (int i = 0; i < cols; i++) {
        if (row[i] < 0)
            return true;
    }
    return false;
}

void remove(int arr[][COLS], int& rows) {
    int writeIndex = 0;

    for (int i = 0; i < rows; i++) {
        if (!isNegative(arr[i], COLS)) { // true, якщо немає від'ємних
            for (int j = 0; j < COLS; j++) {
                arr[writeIndex][j] = arr[i][j];
            }
            writeIndex++;
        }
    }

    for (int i = writeIndex; i < rows; i++) {
        for (int j = 0; j < COLS; j++) {
            arr[i][j] = 0;
        }
    }

    rows = writeIndex;
}

int main() {
    int arr[ROWS][COLS] = {
        {1, 2, 3, 4, 5},
        {-1, 2, -3, 4, 5},
        {6, 7, 8, 9, 10},
        {11, 12, -13, 14, 15},
        {16, 17, 18, 19, 20}
    };
    int rows = ROWS;

    cout << "Початковий масив:\n";
    for (int i = 0; i < ROWS; i++) {
```

```

        for (int j = 0; j < COLS; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }

    remove(arr, rows);

    cout << "\nМасив після видалення рядків з від'ємними елементами:\n";
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }

    return 0;
}

```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/blob/epic\\_4\\_practice\\_and\\_labs\\_anastasiia\\_humeniuk/ai\\_11/anastasiia\\_humeniuk/epic4/vns\\_lab\\_5\\_task\\_1\\_variant\\_1\\_anastasiia\\_humeniuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_4_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic4/vns_lab_5_task_1_variant_1_anastasiia_humeniuk.cpp)

### Завдання №3 Algotester Lab 2 – v3

```

#include <iostream>
using namespace std;

int main() {
    int n;
    cin >> n;

    int* a = new int [n];
    for (int i = 0; i < n; ++i) {
        cin >> a[i];
    }

    int left_pos = 0;
    int right_pos = n - 1;

    while (true) {

        if (left_pos == right_pos) {
            cout << left_pos + 1 << " " << right_pos + 1 << endl;
            cout << "Collision" << endl;
            return 0;
        }

        if (left_pos + 1 == right_pos) {
            cout << left_pos + 1 << " " << right_pos + 1 << endl;

```

```

        cout << "Stopped" << endl;
        return 0;
    }

    if (left_pos > right_pos) {
        cout << left_pos + 1 << " " << right_pos + 1 << endl;
        cout << "Miss" << endl;
        return 0;
    }

    left_pos += a[left_pos];
    right_pos -= a[right_pos];
}
delete[] a;
return 0;
}

```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/blob/epic\\_4\\_practice\\_and\\_labs\\_anastasiia\\_humeniuk/ai\\_11/anastasiia\\_humeniuk/epic4/algotester\\_lab\\_2\\_variant\\_3\\_anastasiia\\_humeniuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_4_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic4/algotester_lab_2_variant_3_anastasiia_humeniuk.cpp)

#### Завдання №4 Algotester Lab 3 – v2

```

#include <iostream>
using namespace std;

int main() {
    int N, M;
    cin >> N;

    int* a = new int[N];
    for (int i = 0; i < N; i++) {
        cin >> a[i];
    }

    cin >> M;
    int* b = new int[M];
    for (int i = 0; i < M; i++) {
        cin >> b[i];
    }

    int repetitive_count = 0;
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            if (a[i] == b[j]) {
                repetitive_count++;
                break;
            }
        }
    }
}

```

```

    }

    int unique_count = N;
    for (int j = 0; j < M; j++) {
        bool is_unique = true;
        for (int i = 0; i < N; i++) {
            if (b[j] == a[i]) {
                is_unique = false;
                break;
            }
        }
        if (is_unique) {
            unique_count++;
        }
    }

    cout << repetitive_count << endl;
    cout << unique_count << endl;

    delete[] a;
    delete[] b;

    return 0;
}

```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/blob/epic\\_4\\_practice\\_and\\_labs\\_anastasiia\\_humeniuk/ai\\_11/anastasiia\\_humeniuk/epic4/algotester\\_lab\\_3\\_variant\\_2\\_a\\_nastasiia\\_humeniuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_4_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic4/algotester_lab_3_variant_2_a_nastasiia_humeniuk.cpp)

## Завдання №5 Class Practice Work

```

#include <iostream>
using namespace std;

bool isPalindrome(const string& str, int start, int end) {
    if (start >= end) {
        return true;
    }
    if (str[start] == str[end]) {
        return isPalindrome(str, start + 1, end - 1);
    }
    return false;
}

bool isPalindrome(int number) {
    int reverseNumber = 0;
    int originalNumber = number;

```

```

        while (number > 0) {
            int digit = number % 10;
            reverseNumber = reverseNumber * 10 + digit;
            number /= 10;
        }

        return originalNumber == reverseNumber;
    }

int main() {
    int number;
    string word;
    char check;

    cout << "Enter the value you want to check for palindrome. W - word, N - number" << endl;
    cin >> check;

    if (check == 'W' || check == 'w') {
        cout << "Enter the word: ";
        cin >> word;
        cout << (isPalindrome(word, 0, word.length() - 1) ? "Is Palindrome" : "Is Not Palindrome") << endl;
    }
    else if (check == 'N' || check == 'n') {
        cout << "Enter the number: ";
        cin >> number;
        cout << (isPalindrome(number) ? "Is Palindrome" : "Is Not Palindrome") << endl;
    }
    else {
        cout << "Wrong value" << endl;
    }

    return 0;
}

```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/blob/epic\\_4\\_practice\\_and\\_labs\\_anastasiia\\_humeniuk/ai\\_11/anastasiia\\_humeniuk/epic4/practice\\_work\\_task\\_1\\_anastasiia\\_humeniuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_4_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic4/practice_work_task_1_anastasiia_humeniuk.cpp)

## Завдання №6 Self Practice Work

```

#include <iostream>
#include <string>
using namespace std;

struct Address {
    string city;

```

```

};

struct Person {
    string name;
    int age;
    Address address;
};

void inputPerson(Person* people, int size) {
    for (int i = 0; i < size; i++) {
        cout << "Введіть ім'я для людини " << i + 1 << ": ";
        cin >> people[i].name;
        cout << "Введіть вік для людини " << i + 1 << ": ";
        cin >> people[i].age;
        cout << "Введіть місто для людини " << i + 1 << ": ";
        cin >> people[i].address.city;
    }
}

void display(const Person* people, int size) {
    for (int i = 0; i < size; i++) {
        cout << "Людина " << i + 1 << ": " << people[i].name << ", Вік: " <<
people[i].age
        << ", Адреса: " << people[i].address.city << endl;
    }
}

void sortAge(Person* people, int size) {
    for (int i = 0; i < size - 1; i++) {
        for (int j = 0; j < size - i - 1; j++) {
            if (people[j].age > people[j + 1].age) {
                Person temp = people[j];
                people[j] = people[j + 1];
                people[j + 1] = temp;
            }
        }
    }
}

int main() {
    int size;
    cout << "Введіть кількість людей: ";
    cin >> size;

    Person* people = new Person[size];
    inputPerson(people, size);

    cout << "\nДо сортування:\n";
    display(people, size);

    sortAge(people, size);
}

```



```

cout << "\nПісля сортування за віком:\n";
display(people, size);

delete[] people;

return 0;
}

```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/blob/epic\\_4\\_practice\\_and\\_labs\\_anastasiia\\_humeniuk/ai\\_11/anastasiia\\_humeniuk/epic4/self\\_practice\\_work\\_anastasiia\\_humeniuk.cpp](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_4_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic4/self_practice_work_anastasiia_humeniuk.cpp)

4) *Результати виконання завдань, тестування та фактично затрачений час:*

### Завдання №1 VNS Lab 4 - Task 1-10

```

PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

PS C:\Users\User\Desktop\lpnu\epic4> & 'c:\Users\User\.vscode\extensions\ms-vscode.cpptools\bin\chex.exe' '--stdin=Microsoft-MIEngine-In-eni2txde.waq' '--stdout=Microsoft-MIEngine-Stdout-eni2txde.waq' '--pid=Microsoft-MIEngine-Pid-swihddzv.wd0' '--dbgExe=C:\msys64\bin\gdb.exe'
Введіть довжину масиву : 8
Початковий масив: 41 67 34 0 69 24 78 58
Введіть 3 нових числа для додавання в кінець масиву: 45 46 73
Кінцевий масив: 24 78 58 45 46 73
PS C:\Users\User\Desktop\lpnu\epic4>

```

Планований час: 20 хв. Фактичний: 35 хв.

### Завдання №2 VNS Lab 5 - Task 1-10

```

PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

wo.t2l' '--pid=Microsoft-MIEngine-Pid-glungptp.gbu' '--dbgExe=C:\msys64\bin\gdb.exe'
Початковий масив:
1 2 3 4 5
-1 2 -3 4 5
6 7 8 9 10
11 12 -13 14 15
16 17 18 19 20

Масив після видалення рядків з від'ємними елементами:
1 2 3 4 5
6 7 8 9 10
16 17 18 19 20
0 0 0 0 0
0 0 0 0 0
PS C:\Users\User\Desktop\lpnu\epic4>

```

Планований час: 30 хв. Фактичний: 30 хв.

## Завдання №3 Algotester Lab 2 – v3

```

PROBLEMS    OUTPUT    PORTS    DEBUG CONSOLE    TERMINAL
PS C:\Users\User\Desktop\lpnu\epic4> & 'c:\Users\User\.vscode\bin\nchcr.exe' '--stdin=Microsoft-MIEngine-In-vtjsojkk.tn2' '--stdout=Microsoft-MIEngine-Pid-lgi1gsy4.gp5' '--dbgE
1 3 1 1 5 1 1 2 1 2
PS C:\Users\User\Desktop\lpnu\epic4> ^C
PS C:\Users\User\Desktop\lpnu\epic4>
PS C:\Users\User\Desktop\lpnu\epic4> & 'c:\Users\User\.vscode\bin\nchcr.exe' '--stdin=Microsoft-MIEngine-In-451k11fv.hqa' '--stdout=Microsoft-MIEngine-Pid-lfhd5tha.qfm' '--dbgE
10
1 3 1 1 5 1 1 3 1 2
5 5
Collision
PS C:\Users\User\Desktop\lpnu\epic4> ^C
PS C:\Users\User\Desktop\lpnu\epic4>
PS C:\Users\User\Desktop\lpnu\epic4> & 'c:\Users\User\.vscode\bin\nchcr.exe' '--stdin=Microsoft-MIEngine-In-4jglwtlw.y03' '--stdout=Microsoft-MIEngine-Pid-nw02dlb4.r3m' '--dbgE
10
1 3 1 1 5 1 1 5 1 2
5 3
Miss

```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
3 hours ago	C++ 23	Accepted	0.003	1.203	<a href="#">View</a>
2 days ago	C++ 23	Accepted	0.003	1.203	<a href="#">View</a>

Планований час: 20 хв. Фактичний: 35 хв.

## Завдання №4 Algotester Lab 3 – v2

```
PS C:\Users\User\Desktop\lpnu\epic4> & 'c:\Users\User\
ncher.exe' '--stdin=Microsoft-MIEngine-In-rtvxegtu.gag'
rd.tc4' '--pid=Microsoft-MIEngine-Pid-m0zbnlvd.54u' '--d
5
1 2 3 4 5
5
4 5 6 7 8
2
8
PS C:\Users\User\Desktop\lpnu\epic4> ^C
PS C:\Users\User\Desktop\lpnu\epic4>
PS C:\Users\User\Desktop\lpnu\epic4> & 'c:\Users\User\
ncher.exe' '--stdin=Microsoft-MIEngine-In-catczemz.oes'
0m.z2a' '--pid=Microsoft-MIEngine-Pid-eceb14xd.tqm' '--d
5
1 2 3 4 5
5
6 7 8 9 10
0
10
PS C:\Users\User\Desktop\lpnu\epic4> |
```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
5 hours ago	C++ 23	Accepted	0.003	1.207	<a href="#">View</a>
2 days ago	C++ 23	Accepted	0.003	1.191	<a href="#">View</a>

Планований час: 30 хв. Фактичний: 30 хв.

## Завдання №5 Class Practice Work

```

PROBLEMS  OUTPUT  PORTS  DEBUG CONSOLE  TERMINAL

PS C:\Users\User\Desktop\lpnu\epic4> & 'c:\Users\User\.vscode\extensions\msncher.exe' '--stdin=Microsoft-MIEngine-In-iszr50uv.urt' '--stdout=Microsoft-Iux.vgt' '--pid=Microsoft-MIEngine-Pid-yb0qwnju.x5j' '--dbgExe=C:\msys64\ucrt
Enter the value you want to check for palindrome. W - word, N - number
W
Is Palindrome
PS C:\Users\User\Desktop\lpnu\epic4> ^C
PS C:\Users\User\Desktop\lpnu\epic4> & 'c:\Users\User\.vscode\extensions\msncher.exe' '--stdin=Microsoft-MIEngine-In-24vzgsyr.w4j' '--stdout=Microsoft-I5x.mtm' '--pid=Microsoft-MIEngine-Pid-jtlxrybq.dmq' '--dbgExe=C:\msys64\ucrt
Enter the value you want to check for palindrome. W - word, N - number
n
Enter the number: 12321
Is Palindrome
PS C:\Users\User\Desktop\lpnu\epic4> ^C
PS C:\Users\User\Desktop\lpnu\epic4>
PS C:\Users\User\Desktop\lpnu\epic4> & 'c:\Users\User\.vscode\extensions\msncher.exe' '--stdin=Microsoft-MIEngine-In-s0g1uuji.dsl' '--stdout=Microsoft-Iah.dnm' '--pid=Microsoft-MIEngine-Pid-d5vccucz.g4c' '--dbgExe=C:\msys64\ucrt
Enter the value you want to check for palindrome. W - word, N - number
n
Enter the number: 12345
Is Not Palindrome
PS C:\Users\User\Desktop\lpnu\epic4> █

```

Планований час: 45 хв. Фактичний: 40 хв.

## Завдання №6 Self Practice Work

```
PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

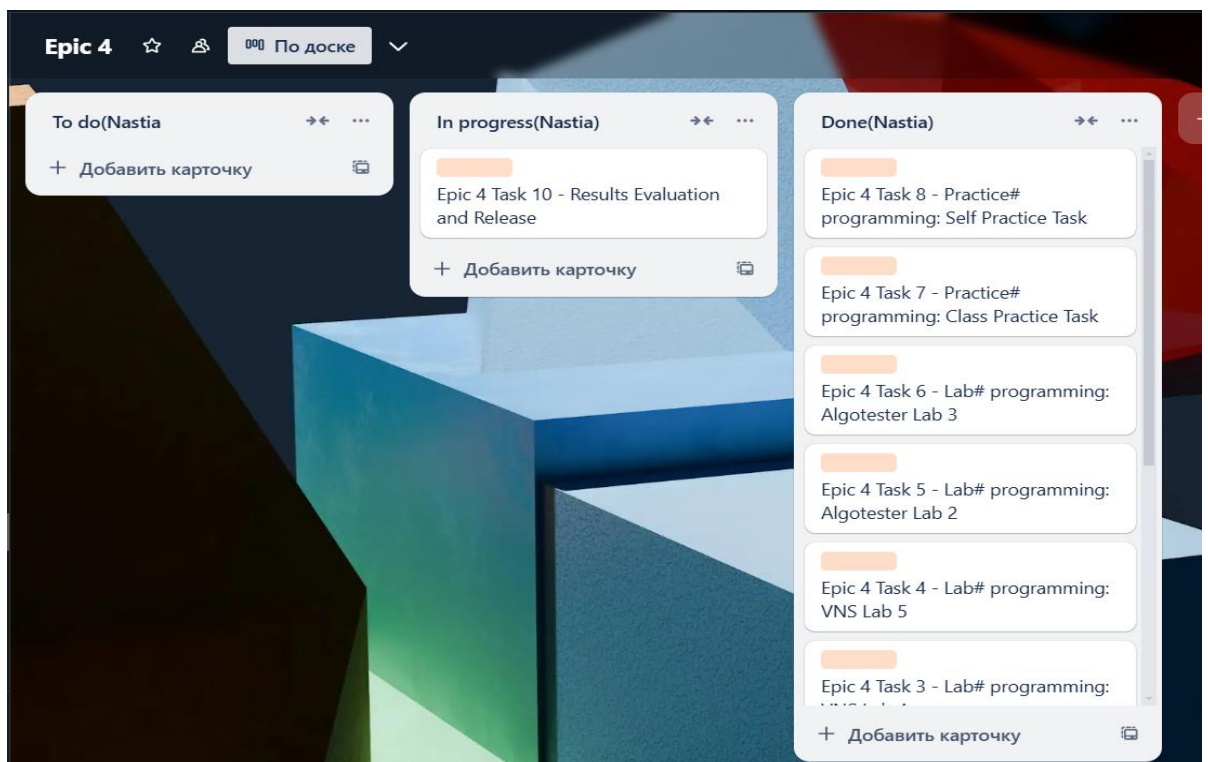
PS C:\Users\User\Desktop\lpnu\epic4> & 'c:\Users\User\.\ncher.exe' '--stdin=Microsoft-MIEngine-In-yzr3a2qk.zfz' z3.mww' '--pid=Microsoft-MIEngine-Pid-fhxyysfi.ice' '--d
Введіть кількість людей: 4
Введіть ім'я для людини 1: Vika
Введіть вік для людини 1: 17
Введіть місто для людини 1: Lviv
Введіть ім'я для людини 2: Ivan
Введіть вік для людини 2: 15
Введіть місто для людини 2: Kharkiv
Введіть ім'я для людини 3: Pavlo
Введіть вік для людини 3: 45
Введіть місто для людини 3: Lviv
Введіть ім'я для людини 4: Orest
Введіть вік для людини 4: 18
Введіть місто для людини 4: Lutsk

До сортування:
Людина 1: Vika, Вік: 17, Адреса: Lviv
Людина 2: Ivan, Вік: 15, Адреса: Kharkiv
Людина 3: Pavlo, Вік: 45, Адреса: Lviv
Людина 4: Orest, Вік: 18, Адреса: Lutsk

Після сортування за віком:
Людина 1: Ivan, Вік: 15, Адреса: Kharkiv
Людина 2: Vika, Вік: 17, Адреса: Lviv
Людина 3: Orest, Вік: 18, Адреса: Lutsk
Людина 4: Pavlo, Вік: 45, Адреса: Lviv
PS C:\Users\User\Desktop\lpnu\epic4> |
```

Планований час: 45 хв. Фактичний: 45 хв.

### 5) Кооперація з командою:





**Висновок:** Виконуючи четвертий епік, я ознайомилася з роботою з одновимірними та двовимірними масивами, а також їх оголошенням, ініціалізацією та використанням у C++. Вивчення взаємодії між масивами та вказівниками допомогло мені краще зрозуміти управління пам'яттю та використання функцій. Дослідження статичних і динамічних масивів, а також операторів new і delete дало глибше розуміння принципів роботи з динамічною пам'яттю. Вивчила структури даних та відпраювала знання на практиці, написавши код.