Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту  
  
A blue and white logo

Description automatically generated

**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 С++.
* 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 С++.

* + 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=zopWRlYOXWw>

* + - 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)>

* + - 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>

* + - 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>

* + - 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:

Якшо 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 aiai та ai+1ai+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 a1..ana1..an

У третьому рядку ціле число MM

у четвертому рядку MM цілих чисел b1..bnb1..bn

**Output**

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

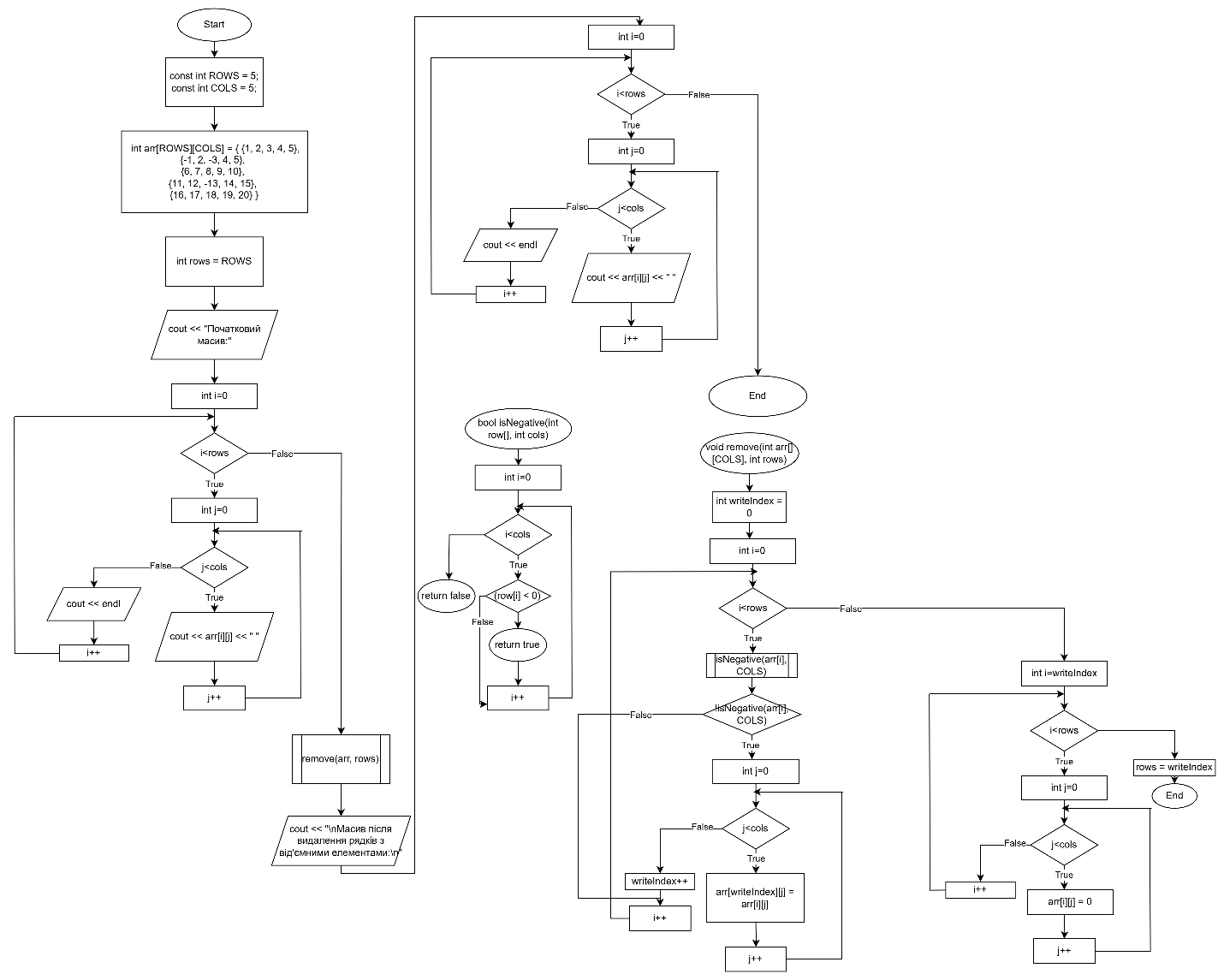
Спочатку програма запитує кількість людей, а потім послідовно вводить для кожної з них ім'я, вік та місто. Після цього програма виводить список людей у порядку введення, сортує його за віком у порядку зростання та знову виводить, але вже відсортований список.

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.

1. *Дизайн виконання завдань:*

**Завдання №2 VNS Lab 5 - Task 1-10**

****

1. *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>

**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>

**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>

**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_anastasiia_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>

**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;

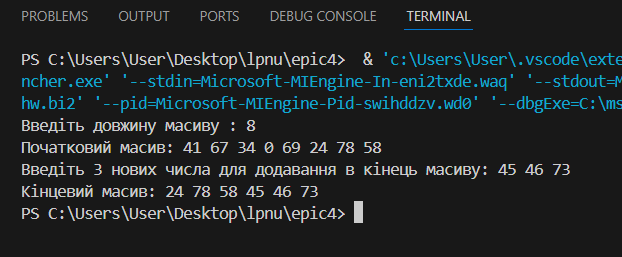
    return 0;

}

<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>

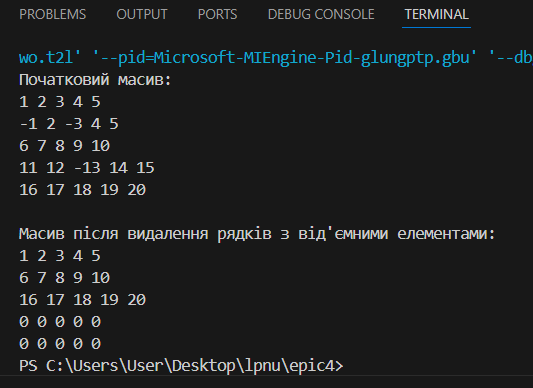
1. *Результати виконання завдань, тестування та фактично затрачений час:*

**Завдання №1 VNS Lab 4 - Task 1-10**

****

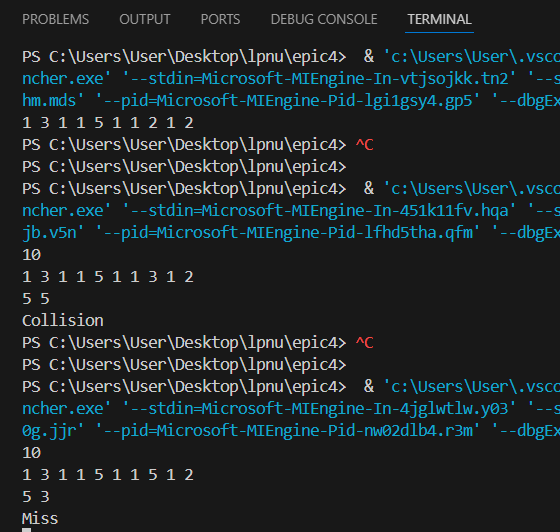
Планований час: 20 хв. Фактичний: 35 хв.

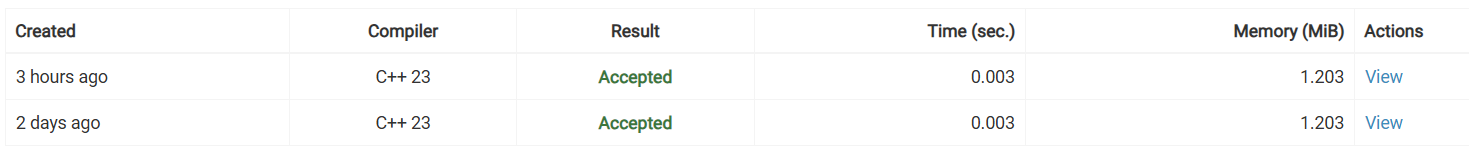
**Завдання №2 VNS Lab 5 - Task 1-10**

****

Планований час: 30 хв. Фактичний: 30 хв.

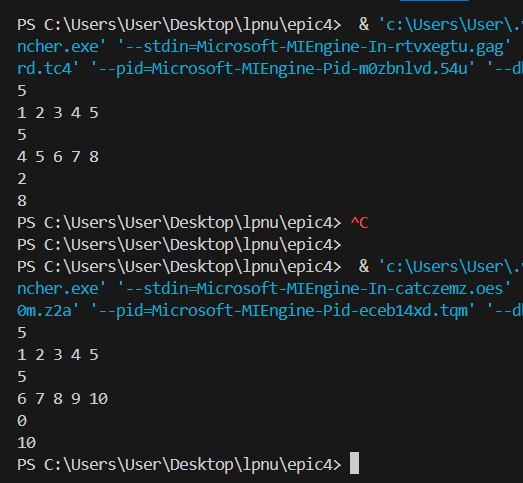
**Завдання №3 Algotester Lab 2 – v3**

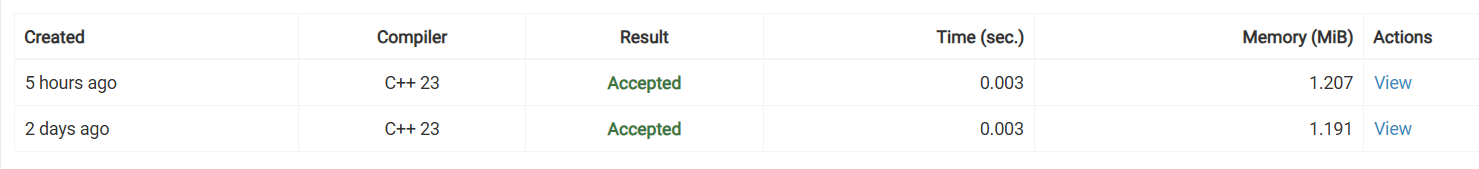
****

****

Планований час: 20 хв. Фактичний: 35 хв.

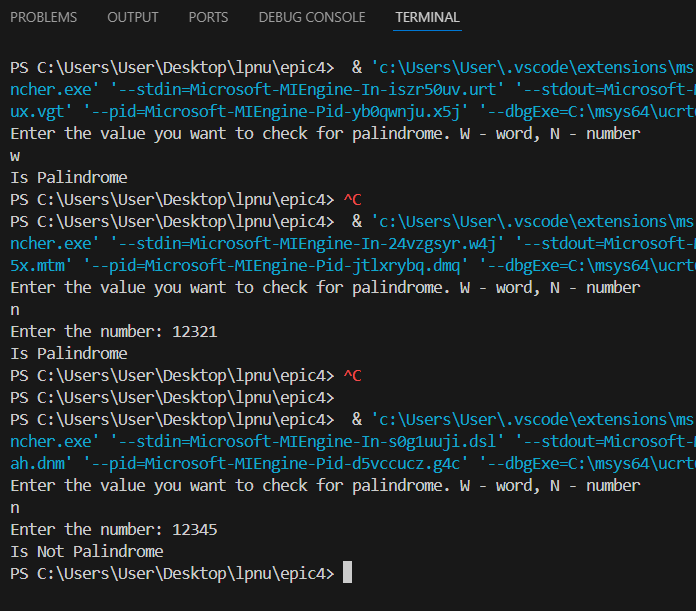
**Завдання №4 Algotester Lab 3 – v2**

****

****

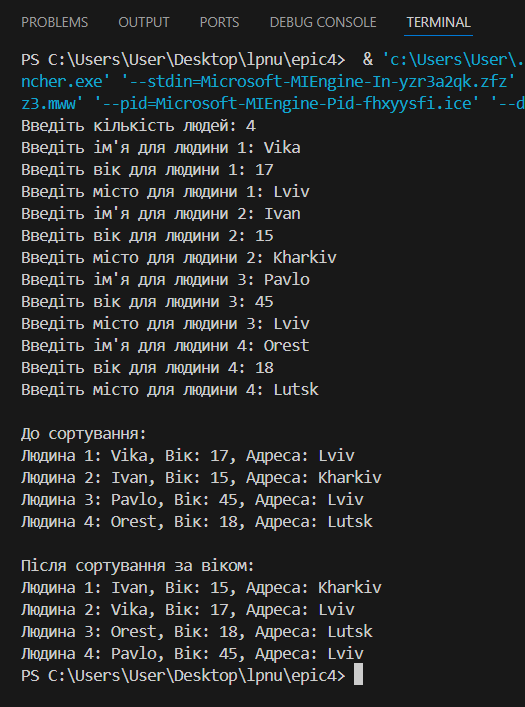
Планований час: 30 хв. Фактичний: 30 хв.

**Завдання №5 Class Practice Work**

****

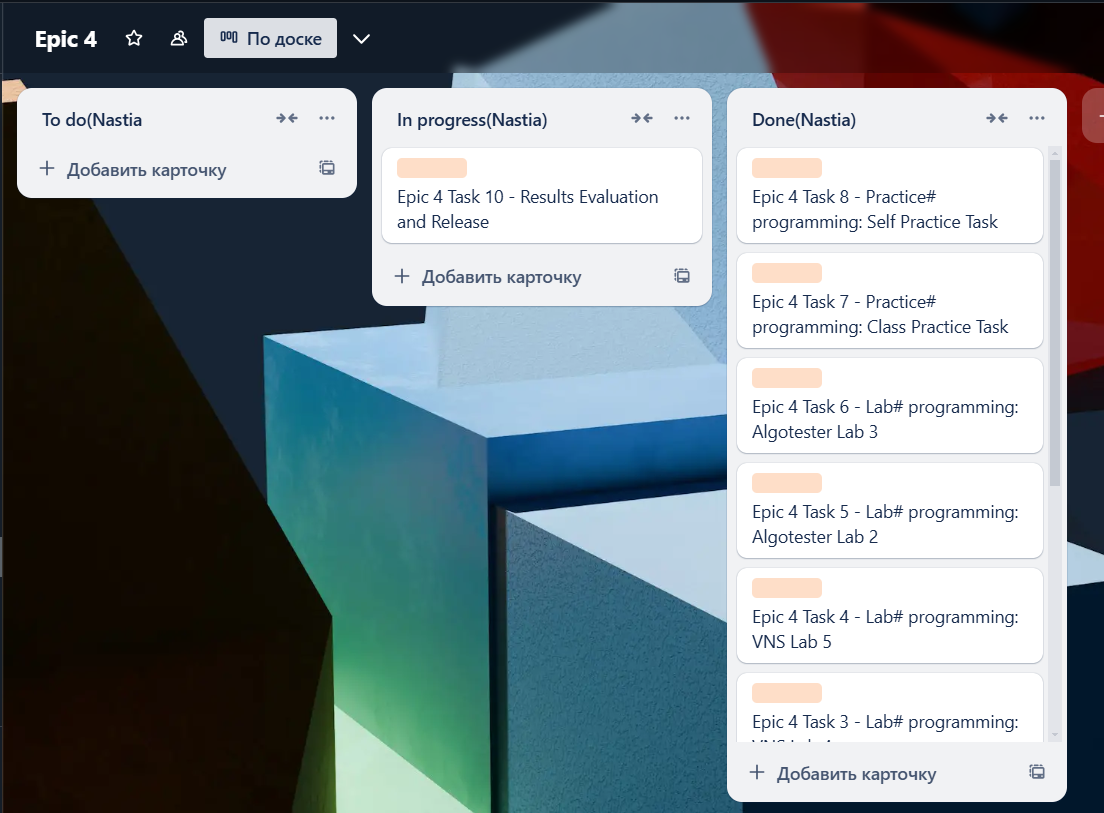
Планований час: 45 хв. Фактичний: 40 хв.

**Завдання №6 Self Practice Work**

****

Планований час: 45 хв. Фактичний: 45 хв.

1. *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.