Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**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:*** «Мови та парадигми програмування»

до:

ВНС Лабораторної Роботи №4

ВНС Лабораторної Роботи №5

Алготестер Лабораторної Роботи №2

Алготестер Лабораторної Роботи №3

Практичних Робіт №4

**Виконав:**

Студент групи ШІ-14

Михальчишин Лук’ян Павлович

**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:**

Навчитись використовувати одновимірні, двовимірні і динамічні масиви в програмах

**Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Theory Education Activities
* Тема №2:  Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-8)
* Тема №3: Lab# programming: VNS Lab 4
* Тема №4: Lab# programming: VNS Lab 5
* Тема №5  Lab# programming: Algotester Lab 2
* Тема №6 Lab# programming: Algotester Lab 3
* Тема №7 Practice# programming: Class Practice Task
* Тема №8 Practice# programming: Self Practice Task

1. Індивідуальний план опрацювання теорії:

* **Тема №1: Theory Education Activities**

1. Джерела Інформації

Статті: <https://acode.com.ua/urok-64-struktury/>

<https://acode.com.ua/urok-90-dynamichni-masyvy/>

Курс: <https://youtu.be/ULdbOaMBPYc?si=YRkP6qnF2A2X5_vJ>

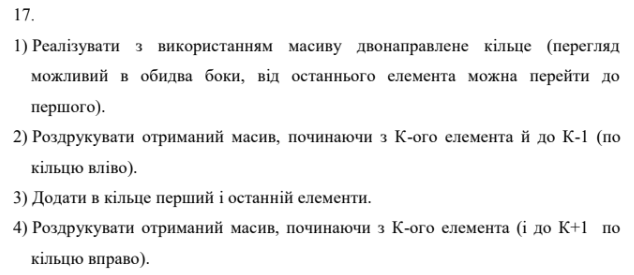
<https://youtu.be/V2g3B9Zbh4Q?si=uo519t4ZADSzDSRb>

<https://youtu.be/999IE-6b7_s?si=3QmnEKm7MDBcWVWL>

1. Що опрацьовано: Поняття які стосуються Одновимірних, двовимірних та динамічних масивів, що таке структури
2. Статус: Ознайомлений

**Виконання роботи:**

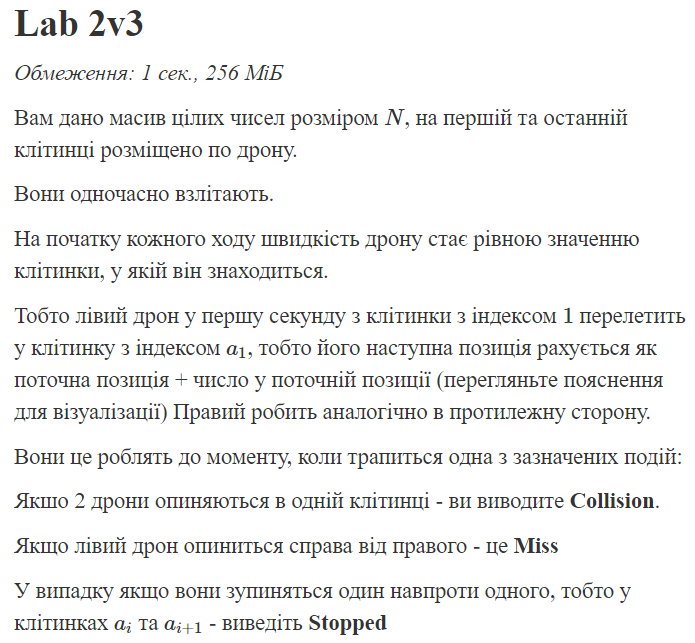
* 1. **Опрацювання завдання та вимог до програм та середовища:**
* Task 2 - Requirements management and design activities with Draw.io and Google Docs
  + Створити блок-схеми до кодів
* Task 3: Lab# programming: VNS Lab 4

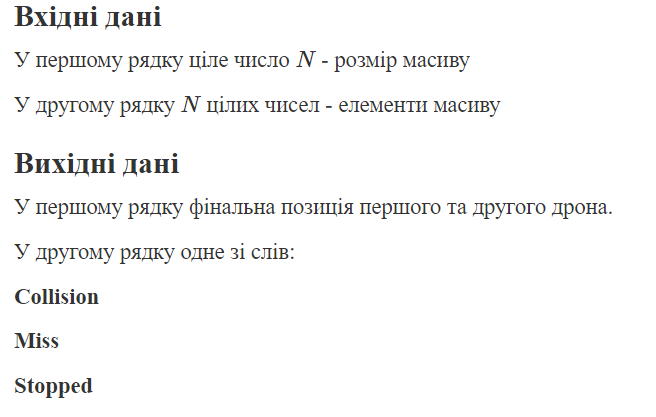


* Task 4: Lab# programming: VNS Lab 5

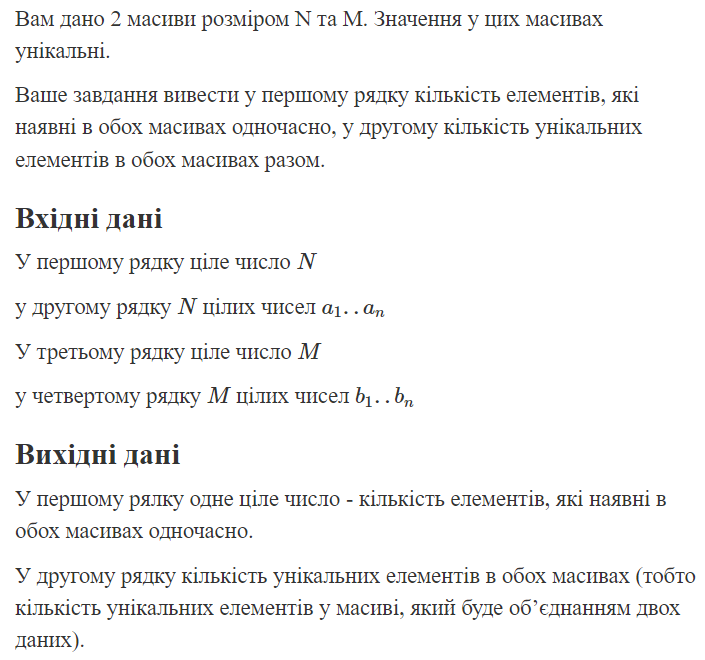


* Task 5: Lab# programming: Algotester Lab 2

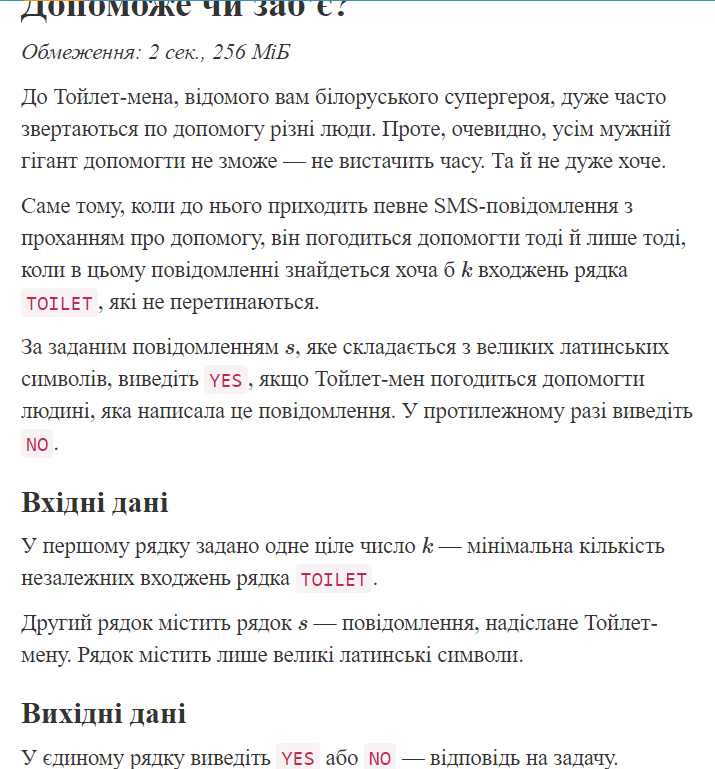




* Task 6: Lab# programming: Algotester Lab 3



* Task 7: Practice# programming: Class Practice Task
  + Визначення чи слово і число є паліндромом
* Task 8: Practice# programming: Self Practice Task
  + Допоможе чи заб’є



**2. Дизайн та планована оцінка часу виконання завдань:**

Блок-схема

* Task 5: Lab# programming: Algotester Lab 2

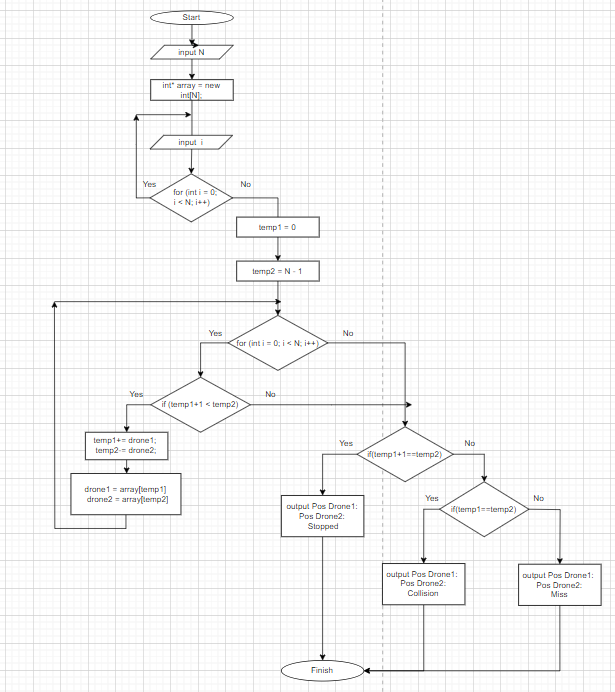


Рисунок : Блок схема до програми №1

* 1. **Код програм з посиланням на зовнішні ресурси:**
* Task 3  Lab# programming: VNS Lab 4

#include <iostream>

using namespace std;

int main()

{

    int N, K;

    cout << "Введіть розмір масиву: ";

    cin >> N;

    cout << "введіть значення К: ";

    cin >> K;

    int\* arr = new int [N];

    for (int i=0; i<N; i++)

    {

        cout << "Введіть " << i+1 << " елементи масиву: ";

        cin >> arr[i];

    }

    int temp = K-1;

    for (int i=0; i<N; i++)

    {

        cout  << arr[temp] << " ";

        temp++;

        if(temp == N)

        {

            temp = 0;

        }

    }

    int\* arr1 = new int[N+2];

    temp = K-1;

    for (int i=0; i < N+2; i++)

    {

        if (i < N)

        {

            arr1[i] = arr[temp];

        }

        else

        {

            if (i == N)

            {

                arr1[i] = arr[K-1];

            }

            else

            {

                arr1[i] = arr[K - 2];

            }

        }

        temp++;

        if(temp == N)

        {

            temp = 0;

        }

    }

    cout << endl;

    temp = K-1;

    for (int i=0; i < N+4; i++)

    {

        cout << arr1[temp] << " ";

        if(temp == N + 1)

        {

            temp = -1;

        }

        temp++;

    }

delete []arr;

    delete []arr1;

    return 0;

}

* Task 4: Lab# programming: VNS Lab 5

#include <iostream>

using namespace std;

int main()

{

    int N, M, Max\_Sum = 0, sum, Max\_row = 0, Max\_col = 0;

    cout << "Введіть кількість рядків: ";

    cin >> N;

    cout << "Введіть кількість стовпців: ";

    cin >> M;

    cout << endl;

    while (N < 3 || N > 100 || M < 3 || M > 100)

    {

        cout << "Розмір масивів має бути від 3 до 100\n";

        cout << "Введіть кількість рядків: ";

        cin >> N;

        cout << "Введіть кількість стовпців: ";

        cin >> M;

    }

    int arr[100][100];

    for (int i = 0; i < N; ++i)

    {

        for (int j = 0; j < M; ++j)

        {

            cout << "Введіть елементи масиву [" << i + 1 << "][" << j + 1 << "]: ";

            cin >> arr[i][j];

        }

    }

    Max\_Sum = arr[0][0] + arr[0][1] + arr[0][2] + arr[1][0] + arr[1][1] + arr[1][2] + arr[2][0] + arr[2][1] + arr[2][2]; // встановлення початкового значення для Max\_Sum

    for (int i = 0; i < N - 3; ++i)

    {

        for (int j = 0; j < M - 3; ++j)

        {

            sum = arr[i][j] + arr[i][j+1] + arr[i][j+2] + arr[i+1][j] + arr[i+1][j+1] + arr[i+1][j+2] + arr[i+2][j] + arr[i+2][j+1] + arr[i+2][j+2]; // сума елементів підмасиву

            if (sum > Max\_Sum) // поточна сума і максимальна

            {

                Max\_Sum = sum;

                Max\_row = i;

                Max\_col = j;

            }

        }

    }

    cout << "Максимальна сума елементів в підмасиві 3 на 3: " << Max\_Sum << endl;

    for (int i = 0; i < 3; ++i)

    {

        for (int j = 0; j < 3; ++j)

        {

            cout << arr[Max\_row + i][Max\_col + j] << " ";

        }

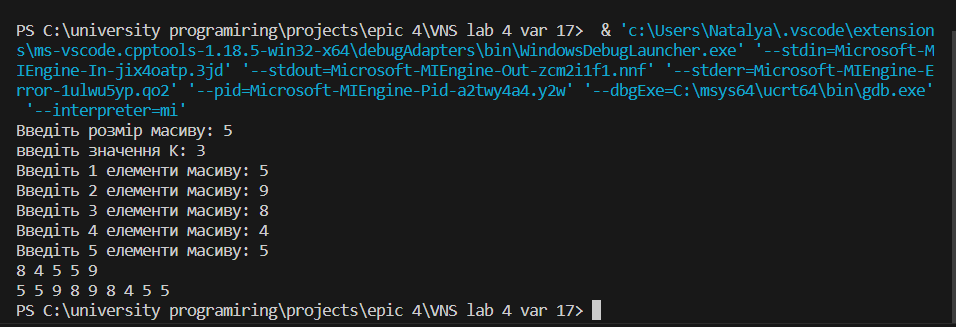
        cout << endl;

    }

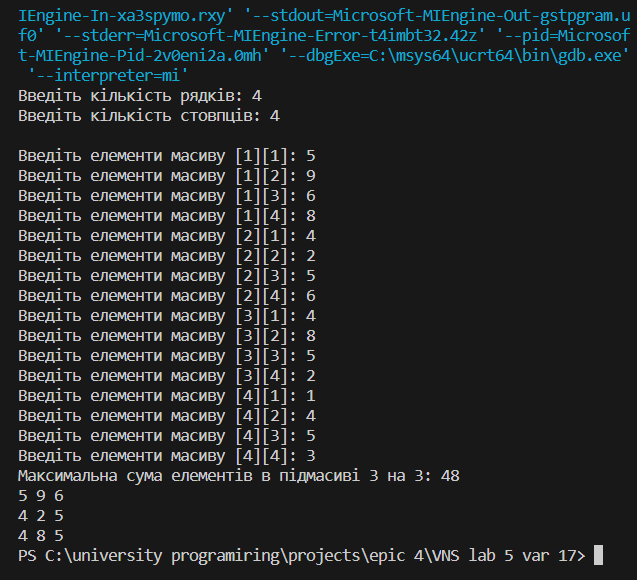
    return 0;

}

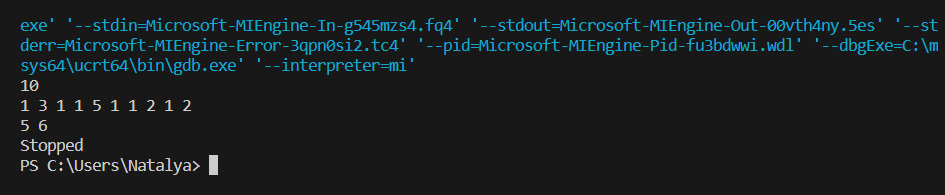
* Task 5: Lab# programming: Algotester Lab 2
* #include <iostream>
* #include <string>
* using namespace std;
* int main()
* {
* int N, drone1 = 0, drone2 = 0, temp1 = 0, temp2;
* cin >> N;
* int\* array = new int[N];
* for (int i = 0; i < N; i++)
* {
* cin >> array[i];
* }
* temp2 = N-1;
* for (int i = 0; i < N; i++)
* {
* if (temp1+1 < temp2)
* {
* temp1+= drone1; // додавання клітинки в які стоїть дрон 1
* temp2-= drone2; // віднімання клітинки в якій стоїть дрон 2
* drone1 = array[temp1]; //елемент на якому зупиниться дрон1
* drone2 = array[temp2]; //елемент на якому зупиниться дрон2
* }
* else
* {
* break;
* }
* }
* if(temp1+1==temp2)
* {
* cout << temp1+1 << " " << temp2+1 << endl; // дрони зупинилися один біля одного
* cout << "Stopped";
* }
* else
* {
* if(temp1==temp2)
* {
* cout << temp1+1 << " " << temp2+1 << endl; // дрони один на одному
* cout << "Collision";
* }
* else
* {
* cout  << temp1+1 << " " <<  temp2+1 << endl; // дрон перелетіли
* cout << "Miss";
* }
* }
* delete []array;
* return 0;
* }
* Task 6: Lab# programming: Algotester Lab 3
* #include <iostream>
* #include <string>
* #include <unordered\_set>
* using namespace std;
* int main ()
* {   int N, M;
* cout << "Введіть розмір масиву N: ";
* cin >> N;
* int\* array1 = new int[N];
* for (int i = 0; i < N; i++)
* {
* cin >> array1[i];
* }
* cout << "Введіть розмір масиву M: ";
* cin >> M;
* int\* array2 = new int[M];
* for (int j = 0; j < M; j++)
* {
* cin >> array2[j];
* }
* unordered\_set <int> common\_Elements; // використовуєм unordered\_set для зберігання: insert для додавання чисел
* for (int i = 0; i < N; i++)
* {
* common\_Elements.insert(array1[i]); // додавання елементів 1 масиву до множини
* }
* int common\_Count = 0;
* for (int j = 0; j < M; j++)
* {
* if (common\_Elements.find(array2[j]) != common\_Elements.end()) // пошук спільних елемнтів
* {
* common\_Count++;
* }
* }
* unordered\_set <int> unique\_Elements;
* for (int i = 0; i < N; i++)
* {
* unique\_Elements.insert(array1[i]); // додавання числа до множини
* }
* for (int j = 0; j < M; j++)
* {
* unique\_Elements.insert(array2[j]);
* }
* cout << "Cпільні елементи: " << common\_Count << endl;
* cout << "Унікальні елементи: " << unique\_Elements.size() << endl;
* delete []array1;
* delete []array2;
* return 0;
* }
* Task 7: Practice# programming: Class Practice Task
* #include <iostream>
* #include <string>
* using namespace std;
* bool isPalindrome(const string & str, int start, int end)
* {
* if (start >= end) // непарна довжина
* {
* return true;
* }
* return (str[start] == str[end]) && isPalindrome (str, start+1, end-1 ); // порівнюєм початкову і кінцеву позицію
* }
* bool isPalindrome (int num)
* {
* string numstr = to\_string(num); //зміна num на рядок
* return isPalindrome (numstr, 0, numstr.length()-1);
* }
* int main()
* {
* string word;
* int number;
* cout << "Введіть одне слово: ";
* cin >> word;
* cout << "Введіть число: ";
* cin >> number;
* cout << endl;
* if (isPalindrome(word, 0, word.length()-1)) // початкова і кінцева пизиція порівняння
* {
* cout << word << " (є паліндромом)\n";
* }
* else
* {
* cout << word << " (не є паліндромом)\n";
* }
* if (isPalindrome(number))
* {
* cout << number << " (є паліндромом)\n";
* }
* else
* {
* cout << number << " (не є паліндромом)\n";
* }
* return 0;
* }
* Task 8: Practice# programming: Self Practice Task
* #include <iostream>
* #include <string>
* using namespace std;
* int main()
* {
* int k;
* string message;
* cout << "Введіть кількість входжень\n";
* cin >> k;
* cout << "Введіть повідомлення\n";
* cin >> message;
* int amount = 0;  // кількість знайдених входжень
* size\_t found = 0; // знаходження позиції TOILET
* while ((found = message.find("TOILET")) != string::npos) //умова перевірки чи знайшовся TOILET
* {
* amount++;
* message = message.substr(found + 6); //стирання всіх символів після першого TOILET
* }
* if (amount >= k)
* {
* cout << "YES\n";
* }
* else
* {
* cout << "NO\n";
* }
* return 0;
* }
  1. **Результати виконання завдань, тестування та фактично затрачений час:**
* Task 3  Lab# programming: VNS Lab 4



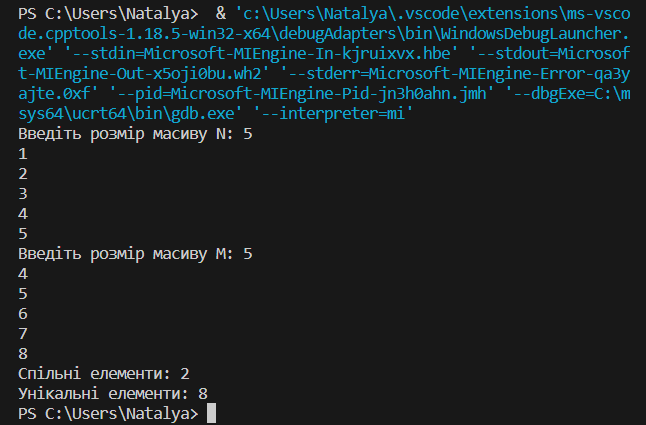
* Task 4: Lab# programming: VNS Lab 5



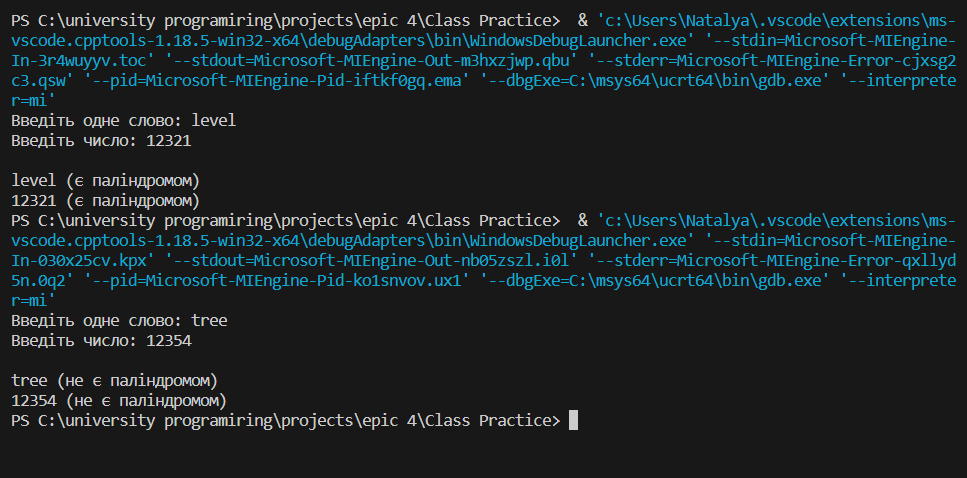
* Task 5: Lab# programming: Algotester Lab 2



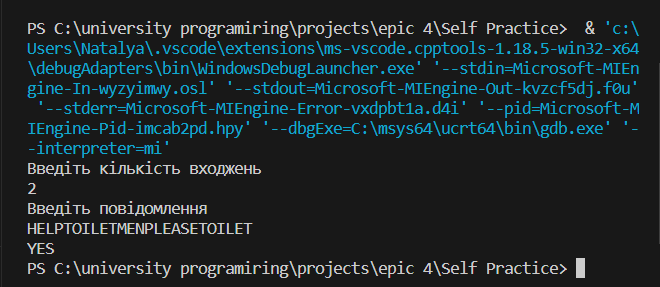
* Task 6: Lab# programming: Algotester Lab 3



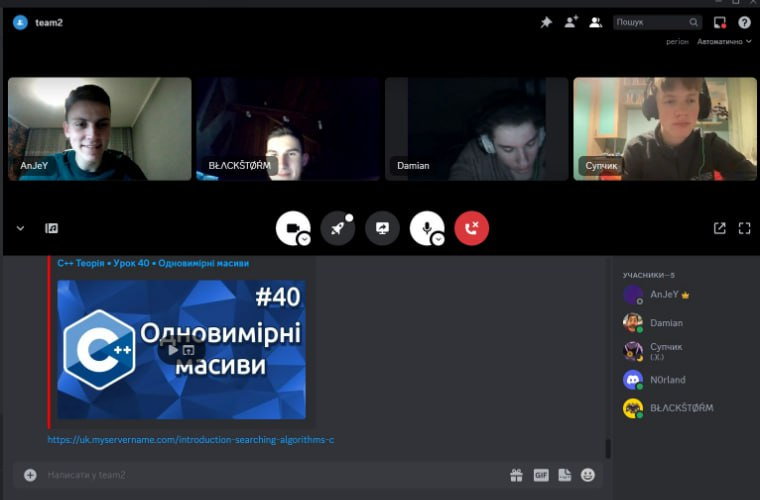
* Task 7: Practice# programming: Class Practice Task



* Task 8: Practice# programming: Self Practice Task



* 1. Онлайн зустріч з командою



Обговорено: Які є типи масивів, як їх застосовувати, поняття динамічних масивів

**Посилання на pull request**

[**https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/680**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/680)

**Висновки:** Навчився використовувати одновимірні, двовимірні, динамічні масиви для вирішення поставлених задач, так само навчитись використовувати структури для обробки інформації