

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:

ВНС Лабораторної Роботи №4

ВНС Лабораторної Роботи №5

Алготестер Лабораторної Роботи №2

Алготестер Лабораторної Роботи №3

Практичних Робіт до блоку №4

Виконав(ла):

Студент групи ШІ-12

Климишин Данило Ігорович

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:

Вивчити основи роботи з пам'яттю, масивами, вказівниками та структурами в мові програмування C++. Ознайомитись з поняттями статичної та динамічної пам'яті. Розібратись з одновимірними та двовимірними масивами, динамічними масивами, а також використанням структур даних, включаючи вкладені структури, для представлення складної інформації.

Теоретичні відомості:

1)Масиви

<https://acode.com.ua/urok-77-masyvy/>

<https://acode.com.ua/urok-81-bagatovymirni-masyvy/>

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

<https://acode.com.ua/urok-97-vkazivnyky-na-vkazivnyky/>

2)Структури

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

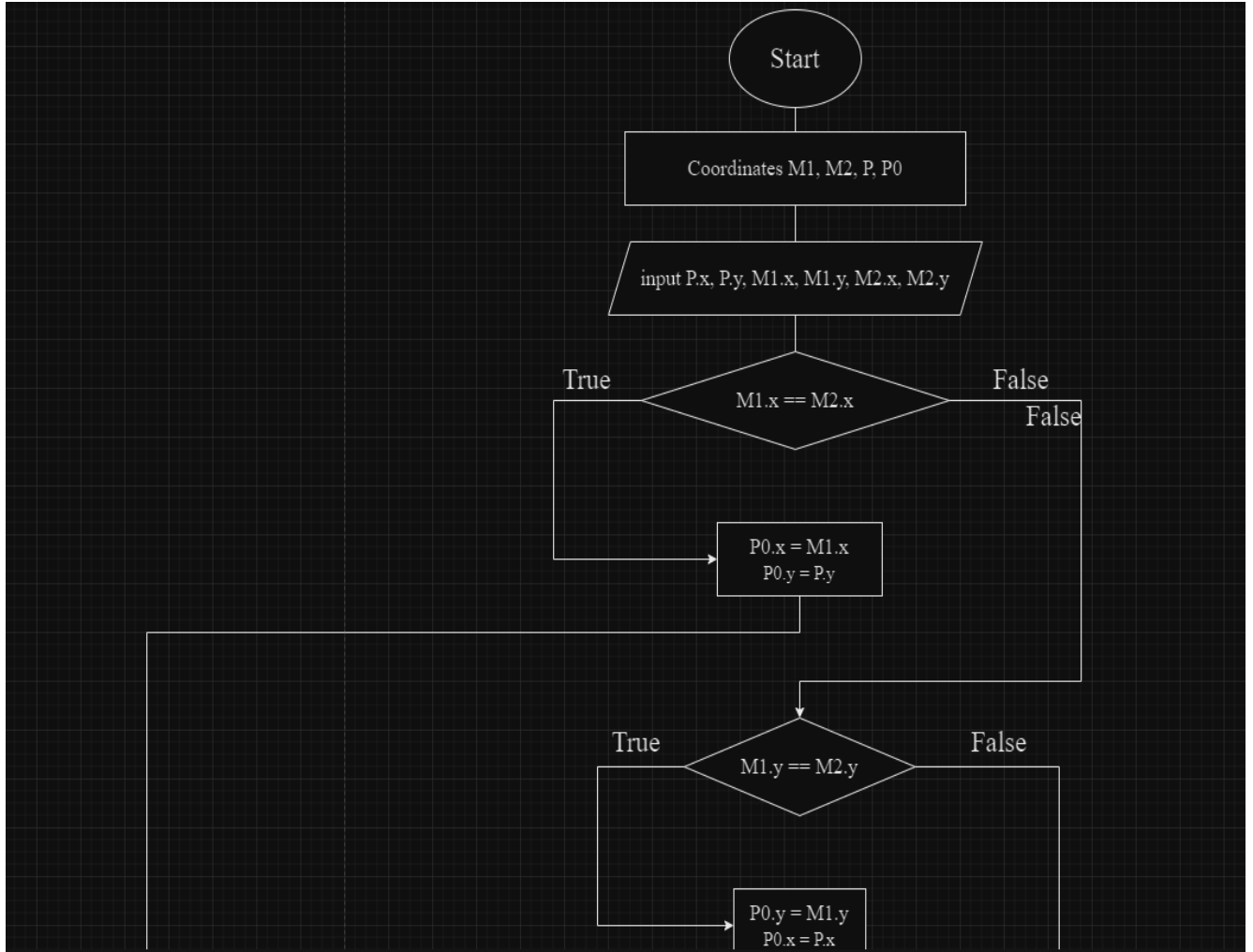
3)Виділення пам'яті

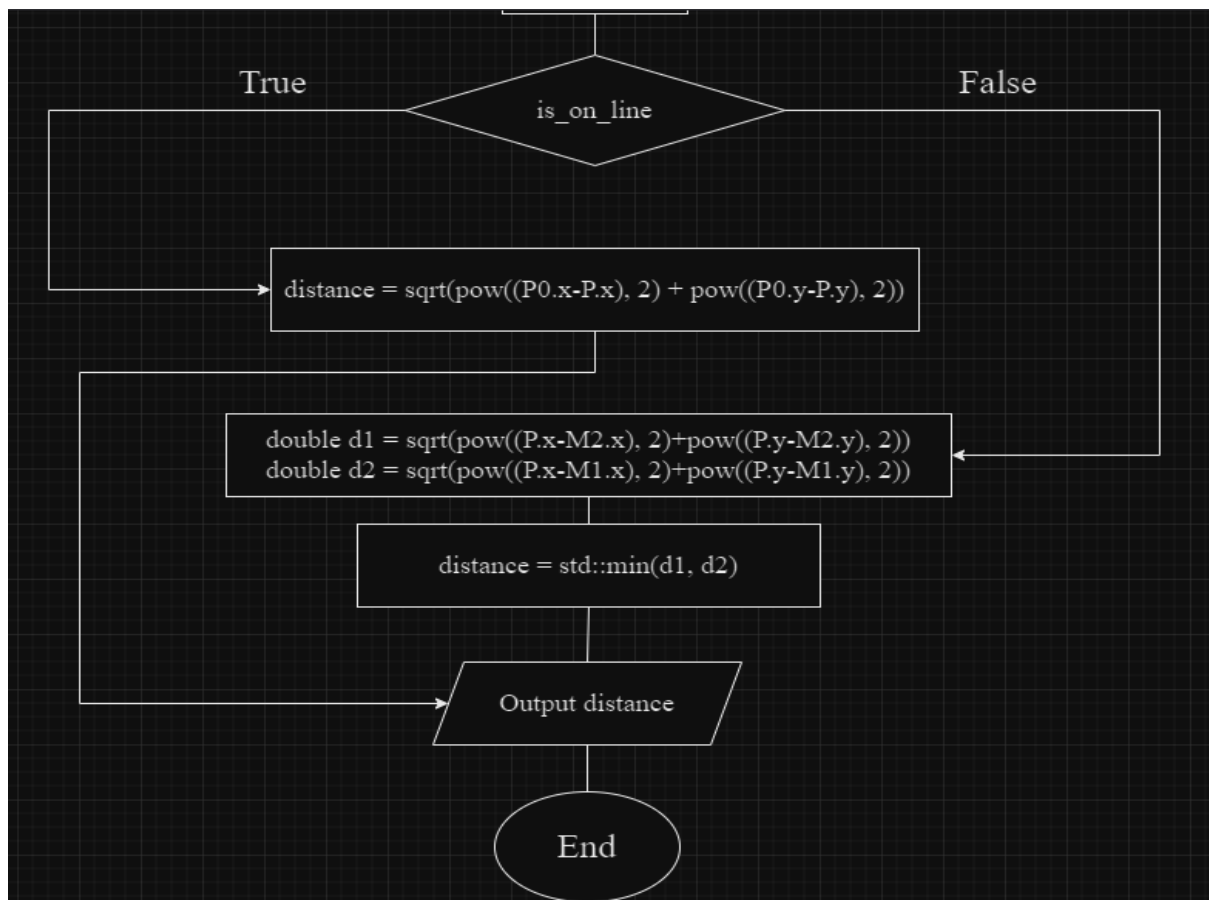
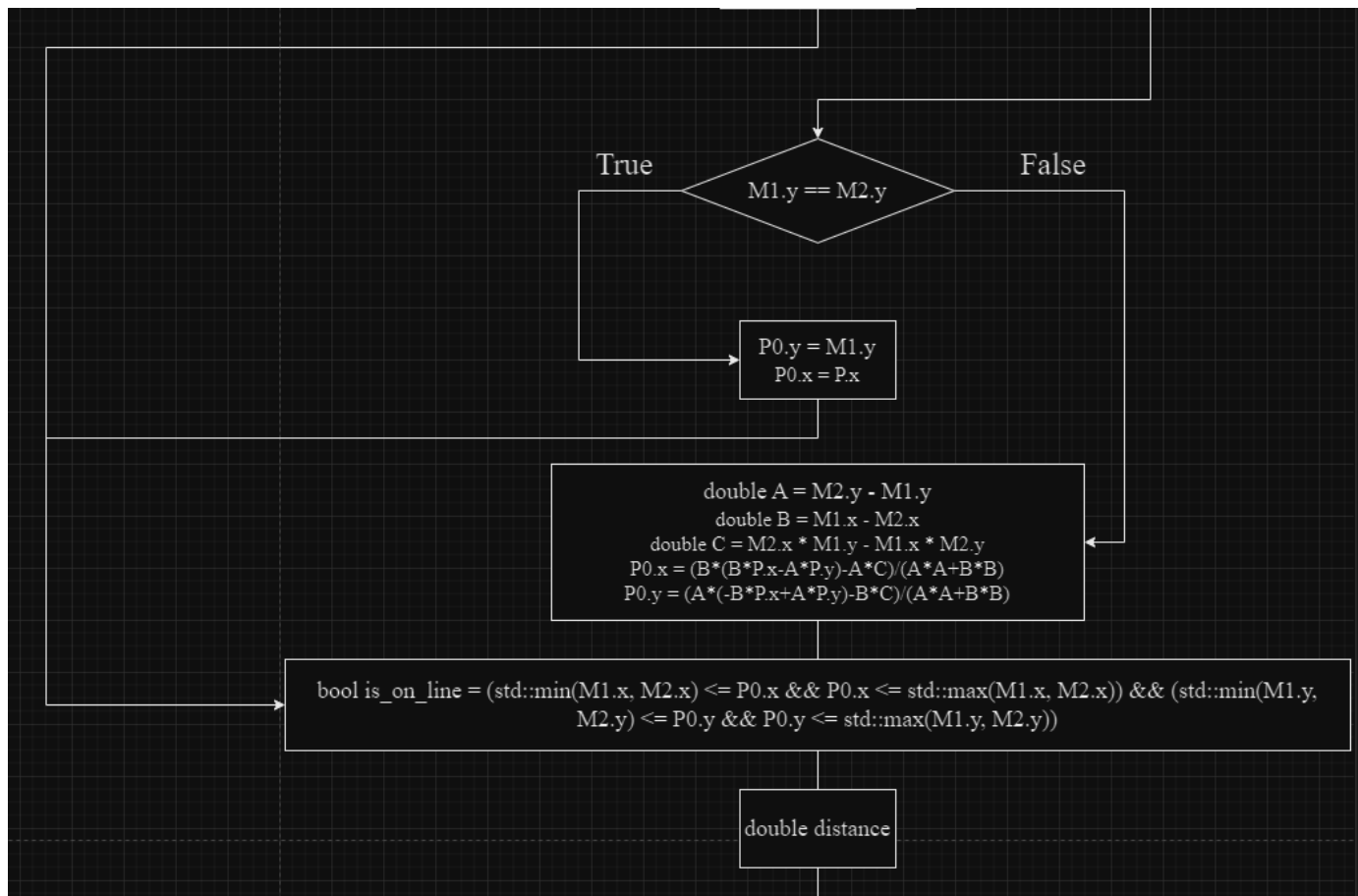
<https://acode.com.ua/urok-111-stek-i-kupa/>

Виконання роботи

Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-8)

Потрапив 30 хв на виконання





Task 3 - Lab# programming: VNS Lab 4

Потратив 40 хв на виконання

```
#include <iostream>

void print_stack(int *array, int &top){

    if(top >= 0){
        for(int i = 0; i <= top; ++i){
            std::cout << array[i] << ' ';
        }
        std::cout << '\n';
    }
    else
        std::cout << "Stack is empty." << std::endl;
}

void push(int *array, int value, int &top){

    if(top < 14){
        array[++top] = value;
        std::cout << "Element " << value << " added succesfully.";
        std::cout << '\n';
        print_stack(array, top);
    }
    else{
        std::cout << "No more space left. ";
    }
}
```

```

void pop(int *array, int &top){

    if(top >= 0){
        std::cout << "Element " << array[top--] << " was deleted.";
        std::cout << '\n';
        print_stack(array, top);
    }
    else{
        std::cout << "Stack is empty";
    }
}

int main(){

    int array[15];

    int top = -1;
    push(array, 10, top);
    push(array, 20, top);
    push(array, 40, top);
    push(array, 100, top);
    pop(array, top);
    pop(array, top);
    pop(array, top);

    return 0;
}

```

```

Element 10 added succesfully.
10
Element 20 added succesfully.
10 20
Element 40 added succesfully.
10 20 40
Element 100 added succesfully.
10 20 40 100
Element 100 was deleted.
10 20 40
Element 40 was deleted.
10 20
Element 20 was deleted.
10

```

Task 4 - Lab# programming: VNS Lab 5

Потратив 30 хв на виконання

```
#include <iostream>

int calculate(int **array){
    double S = 0;
    short counter = 0, n = 0;
    for(int i = 0; i < 6; ++i){
        S += array[i][0];
        ++counter;
    }
    double S_A = S/counter;

    for(int j = 1; j < 6; ++j){
        for(int i = 0; i < 6; ++i){
            if(array[i][j] > S_A)
                ++n;
        }
    }

    return n;
}

int main(){

    int **array = new int*[6];
    for(int i = 0; i < 6; ++i){
        array[i] = new int[6];
    }

    for(int i = 0; i < 6; ++i){
        for(int j = 0; j < 6; ++j){
            std::cin >> array[i][j];
        }
    }

    std::cout << "N = " << calculate(array);

    for(int i=0; i < 6; ++i){
        delete[] array[i];
    }

    delete[] array;
    return 0;
}
```

```
1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28
29 30 31 32 33 35 35 36
N = 17
```

Task 5 - Lab# programming: Algotester Lab 2

Потратив 30 хв на виконання

```
#include <iostream>
#include <vector>

int main(){

    int n, a1, a2, a3;
    std::cin >> n;
    std::vector<int> r {};
    r.resize(n);

    int erase_array[3];

    for(int i = 0; i < n; ++i)
        std::cin >> r[i];

    for(int i = 0; i < 3; ++i){
        std::cin >> erase_array[i];
    }

    for(int i = 0; i < r.size(); ++i){
        for(int j = 0; j < 3; ++j){
            if(r[i] == erase_array[j]){
                r.erase(r.begin() + i);
                --i;
            }
        }
    }

    if(r.size() == 0){
        std::cout << '0';
        return 0;
    }

    std::cout << r.size() - 1 << std::endl;
    for(int i = 0; i < r.size() - 1; ++i){
        std::cout << r[i] + r[i+1] << ' ';
    }

    return 0;
}
```

```
6
1 2 3 4 5 6
3 4 5
2
3 8
```


Task 6 - Lab# programming: Algotester Lab 3

Потратив 20 хв на виконання

```
#include <iostream>
#include <vector>

int main(){
    int N, M, counter1 = 0, counter2 = 0;

    std::vector<int> array1;
    std::vector<int> array2;
    std::vector<int> riz;

    std::cin >> N;
    array1.resize(N);
    for(int i = 0; i < array1.size(); ++i)
        std::cin >> array1[i];

    std::cin >> M;
    array2.resize(M);
    for(int i = 0; i < array2.size(); ++i)
        std::cin >> array2[i];

    for(int i = 0; i < array1.size(); ++i){
        for(int j = 0; j < array2.size(); ++j){
            if (array1[i] == array2[j])
                ++counter1;
        }
    }

    std::cout << counter1 << std::endl;
    std::cout << array1.size() + array2.size() - counter1;

    return 0;
}
```

```
3
4 5 8
5
6 7 8 2 12
1
7
```

Task 7 - Practice# programming: Class Practice Task

На виконання потратив 1.5 години

```
#include <iostream>
#include <string>

bool isPalindrome(const std::string &word, int current_position_1, int current_position_2){
    if(current_position_1 == current_position_2)
        return true;
    return (word[current_position_1] == word[current_position_2]) && isPalindrome(word, current_position_1 + 1, current_position_2 - 1);
}

bool isPalindrome(int a, int n){
    if( a < 10)
        return true;
    while (a / n >= 10) {
        n *= 10;
    }
    int firstDigit = a / n;
    int lastDigit = a % 10;

    return (firstDigit == lastDigit) && isPalindrome((a/n)/n, n*10);
}

int main(){
    std::string word;
    std::cout << "Enter you word: ";
    std::cin >> word;

    bool result_word = isPalindrome(word, 0, word.size()-1);

    if(result_word)
        std::cout << "Your word is a palindrome" << std::endl;
    else
        std::cout << "You word is not a palindrome" << std::endl;

    int digit;
    std::cout << "Enter you digit: ";
    std::cin >> digit;
```

```
bool result_digit = isPalindrome(digit, 10);
if(result_digit)
    std::cout << "Your digit is a palindrom";
else
    std::cout << "Your digit is not a palindrom";

    return 0;
```

```
Enter you word: radar
Your word is a palindrome
Enter you digit: 45654
Your digit is a palindrom
```

Task 8 - Practice# programming: Self Practice Task

На виконання потратив годину

```
#include <iostream>
#include <cmath>
#include <iomanip>

struct Coordinates{
    double x;
    double y;
};

int main(){
    Coordinates M1, M2, P, P0;

    std::cin >> P.x >> P.y >> M1.x >> M1.y >> M2.x >> M2.y;

    if(M1.x == M2.x){
        P0.x = M1.x;
        P0.y = P.y;
    }
    else if(M1.y == M2.y){
        P0.y = M1.y;
        P0.x = P.x;
    }
    else{
        double A = M2.y - M1.y;
        double B = M1.x - M2.x;
        double C = M2.x * M1.y - M1.x * M2.y;

        P0.x = (B*(B*P.x-A*P.y)-A*C)/(A*A+B*B);
        P0.y = (A*(-B*P.x+A*P.y)-B*C)/(A*A+B*B);
    }

    bool is_on_line = (std::min(M1.x, M2.x) <= P0.x && P0.x <= std::max(M1.x, M2.x)) && (std::min(M1.y, M2.y) <= P0.y && P0.y <= std::max(M1.y, M2.y));

    double distance;

    if(is_on_line)
        distance = sqrt(pow((P0.x-P.x), 2) + pow((P0.y-P.y), 2));

    else{
        double d1 = sqrt(pow((P.x-M2.x), 2)+pow((P.y-M2.y), 2));
        double d2 = sqrt(pow((P.x-M1.x), 2)+pow((P.y-M1.y), 2));
        distance = std::min(d1, d2);
    }

    std::cout << std::fixed << std::setprecision(4) << distance << std::endl;

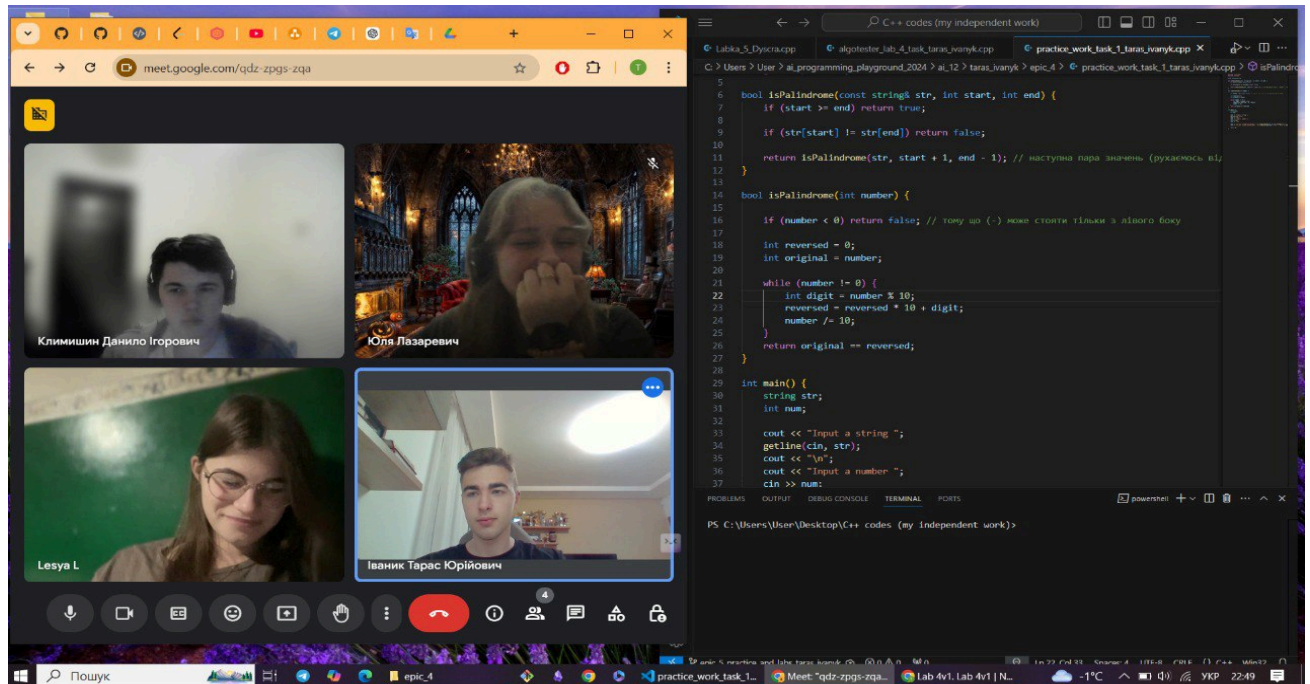
    return 0;
}
```

```
5 6
3 7 8 9
1.6713
```

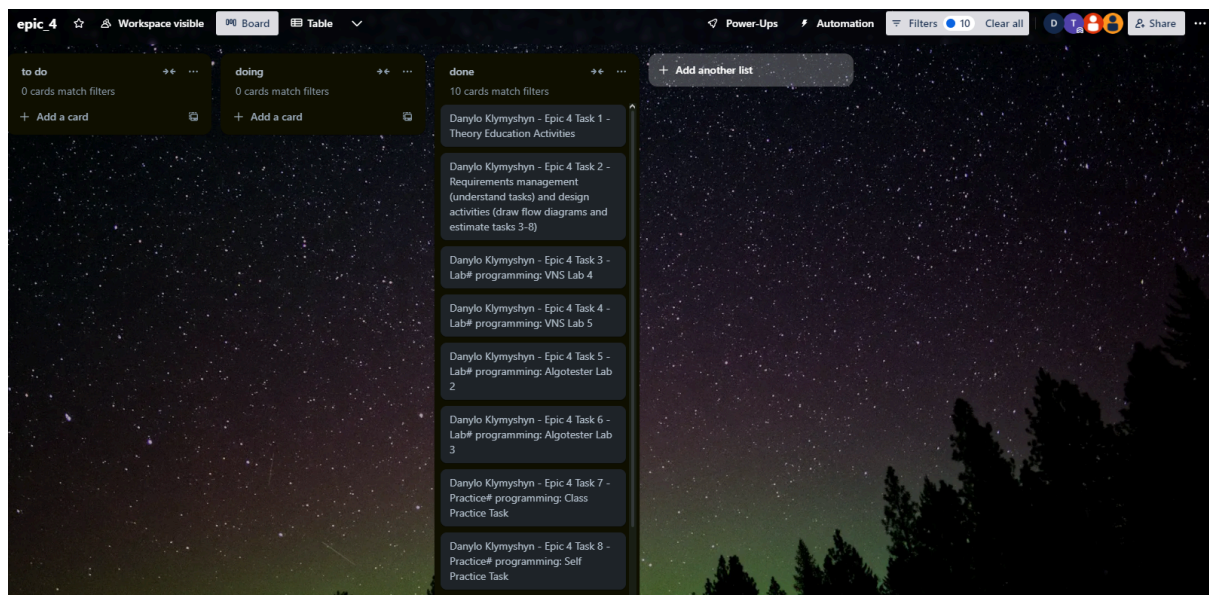
Task 9 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)

Потратив 40 хв на виконання

Зустрічі з командою:



Trello:



Висновок: Я вивчив основи роботи з масивами, вказівниками та динамічною пам'яттю в C++. Освоїв оголошення та ініціалізацію одновимірних і двовимірних масивів, а також різницю між статичними і динамічними масивами. Навчився працювати з вказівниками та посиланнями, а також управляти пам'яттю за допомогою new і delete. Вивчення вкладених структур допомогло зрозуміти представлення складних даних та маніпуляції з ними, що є важливим для розробки ефективних алгоритмів у C++.