Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт №4, 5**

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

***з розділу***: «Прості структури даних. Одновимірні масиви. Двовимірні Масиви. Алгоритми обробки»

***Виконав:***

студент групи ШІ-14

Шелеп Андрій Іванович

# **Тема роботи:**

Прості структури даних у С++. Ознайомлення з одновимірними та двовимірними масивами. Знайомство з базовими алгоритми обробки даних

# **Мета роботи:**

Ознайомитися з одновимірними та двовимірними у С++. Опрацювати та застосувати алгоритми обробки для масивів.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Одновимірні масиви.
* Тема №2: Двовимірні масиви.
* Тема №3: Алгоритми обробки.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Одновимірні масиви.
  + Джерела Інформації:
    - Відео.

<https://www.youtube.com/watch?v=ULdbOaMBPYc>

* + - Стаття.

<http://cpp.dp.ua/masyvy-yak-parametry-funktsij/>

<https://cherto4ka.xyz/2020/01/16/massive_begin/>

* + Що опрацьовано:
    - Одновимірні масиви, їх задання, ініціалізація, прості операції обробки масивів
    - Передання масивів як параметрів функції
  + Статус: Ознайомлений
  + Початок опрацювання теми: 23.11.2023
  + Звершення опрацювання теми: 28.11.2023
* Тема №2: Двовимірні масиви.
  + Джерела Інформації:
    - Відео.

<https://www.youtube.com/watch?v=V2g3B9Zbh4Q>

* + - Стаття.

<https://www.toppr.com/guides/computer-science/programming-in-c-/structured-data-type/two-dimensional-array/>

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

* + Що опрацьовано:
    - Поняття двовимірного масиву. Задання двовимірного масиву, передача у функцію
    - Операції з двовимірним масивом
  + Статус: Ознайомлений
  + Початок опрацювання теми: 23.11.2023
  + Звершення опрацювання теми: 28.11.2023
* Тема №3 Алгоритми обробки.
  + Джерела Інформації:
    - Стаття.

<https://cherto4ka.xyz/2020/01/21/%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D1%96-%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D1%97-%D0%BE%D0%B1%D1%80%D0%BE%D0%B1%D0%BA%D0%B8-%D0%BE%D0%B4%D0%BD%D0%BE%D0%B2%D0%B8%D0%BC%D1%96%D1%80%D0%BD%D0%B8/>

<https://www.w3resource.com/c-programming-exercises/array/c-array-exercise-9.php>

<https://www.programiz.com/dsa/selection-sort>

<https://www.geeksforgeeks.org/insertion-sort/>

<https://www.programiz.com/dsa/bubble-sort>

<https://uk.myservername.com/introduction-searching-algorithms-c>

* + Що опрацьовано:
    - Базові алгоритми обробки одновимірних масивів
    - Лінійний пошук максимума та мінімума
    - Декілька видів сортування масивів (BubbleSort, InsertionSort, SelectionSort)
    - Лінійний та бінарний пошук
  + Статус: Ознайомлений
  + Початок опрацювання теми: 24.11.2023
  + Звершення опрацювання теми: 28.11.2023

# **Виконання роботи:**

## **1. Опрацювання завдання та вимог до програм та середовища:**

Завдання №1 VNS Lab 4 - Task 1

* Варіант 14
* Деталі завдання: 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор

випадкових чисел.

2) Роздрукувати отриманий масив.

3) Перевернути масив і, якщо кількість елементів масиву непарна, знищити

його середній елемент.

4) Додати на початок масиву 3 елементи зі значенням M[I+10]-2.

5) Роздрукувати отриманий масив.

* Важливі деталі для врахування в імплементації програми

Використано генератор випадкових чисел

Завдання №2 VNS Lab 5 - Task 1

* Варіант 14
* Деталі завдання

Обчислити добуток всіх стовпців масиву, у яких перший елемент більший

від елементів розташованих на головній і бічній діагоналі.

* Важливі деталі для врахування в імплементації програми

Використовував двовимірні масиви та вкладені цикли

Завдання №3 Algotester Lab 2

* Варіант 1
* Деталі завдання

Є дорога з N чисел. Потрібно порахувати мінімальну втому (різниця між мінімальним і максимальним числом), якщо можна відкинути 1 елемент з масиву

* Важливі деталі для врахування в імплементації програми

Використовував сортування

Завдання №4 Algotester Lab 3

* Варіант 2
* Деталі завдання

Дано 2 масиви розміром N та M. Значення у цих масивах унікальні. Вивести у першому рядку кількість елементів, які наявні в обох масивах одночасно, у другому кількість унікальних елементів в обох масивах разом.

* Важливі деталі для врахування в імплементації програми

Використовував сортування та вкладені цикли

Завдання №5 Class Practice Work

* Без варіанту
* Деталі завдання

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

* Важливі деталі для врахування в імплементації програми

Створюю функцію логічного типу повернення для рядка, яка рекурсивно викликає себе. Організовую перевантажену функцію для

Завдання №6 Self Practice Work

* Задача з алготестеру - 0022 Верховна Рада
* Деталі завдання

У виборах до Верховної Ради взяли участь n партій. Вам відома кількість голосів, що отримала кожна з них. Вивести мінімальну загальну кількість депутатських місць.

Відношення кількості голосів до кількості місць у Верховній Раді має бути однаковим для усіх партій.

* Важливі деталі для врахування в імплементації програми

1<=n<=1000

1<=votes[i]<=10^9

Використовую масив і створюю функцію, яка рахує найменший спільний дільник двох чисел

Завдання №7 Self Practice Work

* Задача з алготестеру - 1681 соціальна дистанція
* Деталі завдання

Потрібно сказати, чи між кожною парою людей відстань є не меншою за 2 метри.

* Важливі деталі для врахування в імплементації програми

1<=n<=1000

1<=pos[i]<=10^9

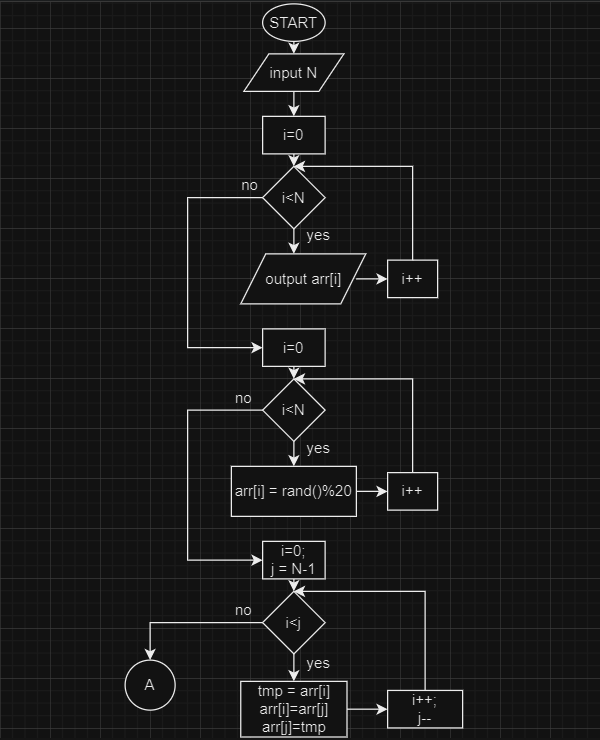
pos[i]<=pos[i+1]

Використовую масив і цикл для перевірки дистанції

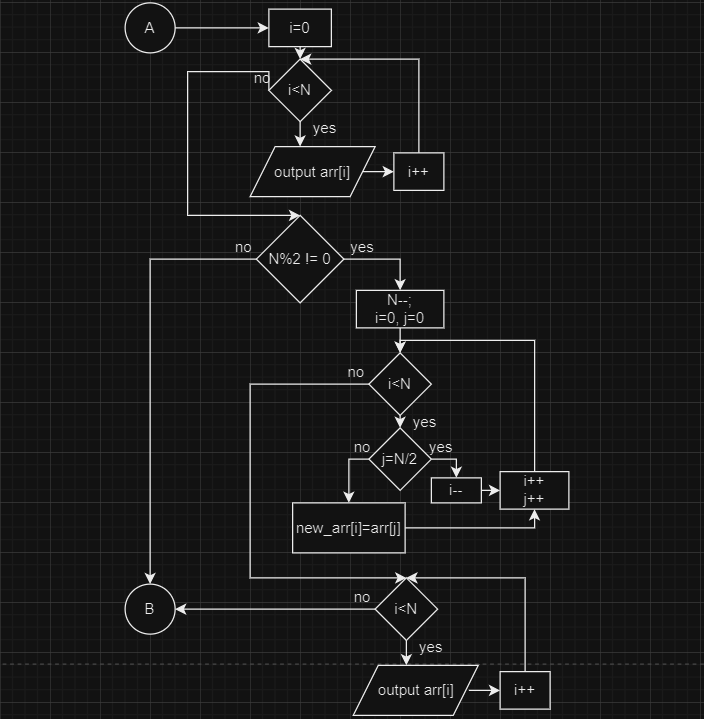
## **2. Дизайн та планована оцінка часу виконання завдань:**

Програма №1 VNS Lab 4 - Task 1

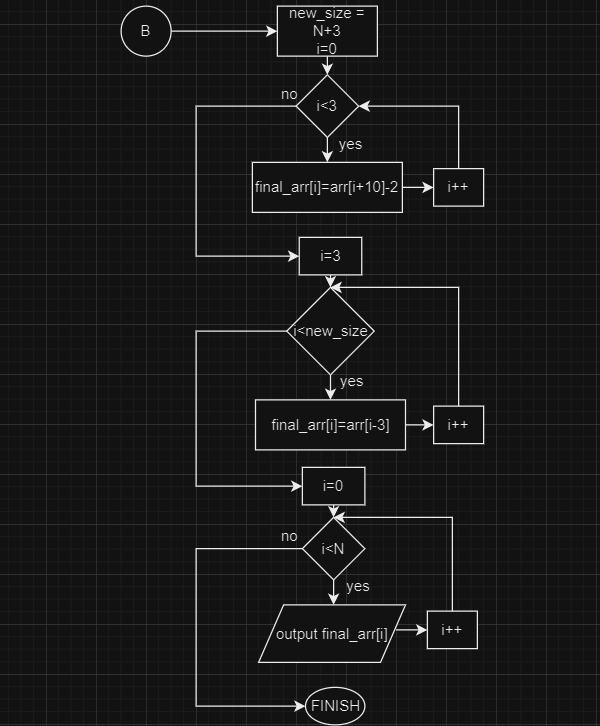
* Блок-схема



*Зображення №1. Блок-схема до програми №1 VNS Lab 4 - Task 1*



*Зображення №2. Блок-схема до програми №1 VNS Lab 4 - Task 1*



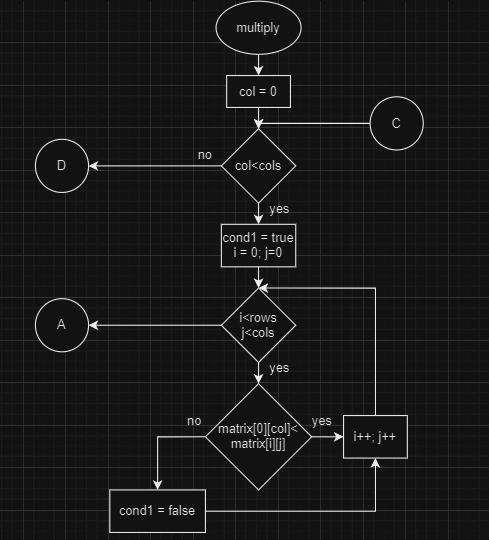
*Зображення №3. Блок-схема до програми №1 VNS Lab 4 - Task 1*

* Планований час на реалізацію: 45 хв
* Важливі деталі для врахування в імплементації

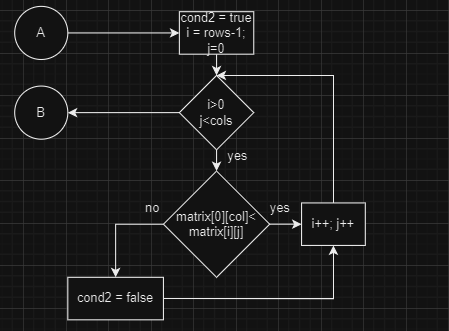
Використано генератор випадкових чисел

Програма №2 VNS Lab 5 - Task 1

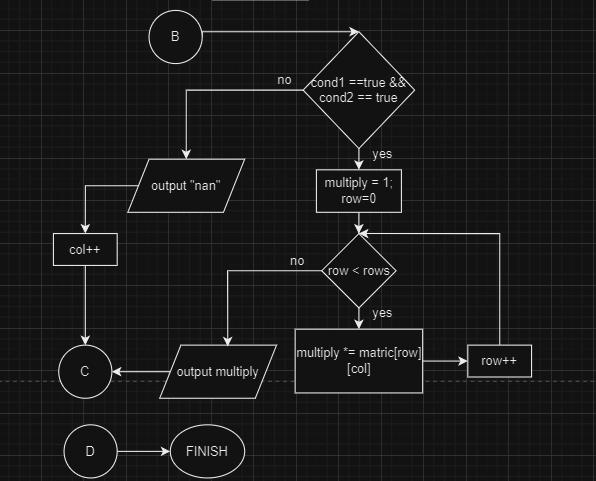
* Блок-схема



*Зображення №4. Блок-схема до програми №2 VNS Lab 5 - Task 1*



*Зображення №5. Блок-схема до програми №2 VNS Lab 5 - Task 1*



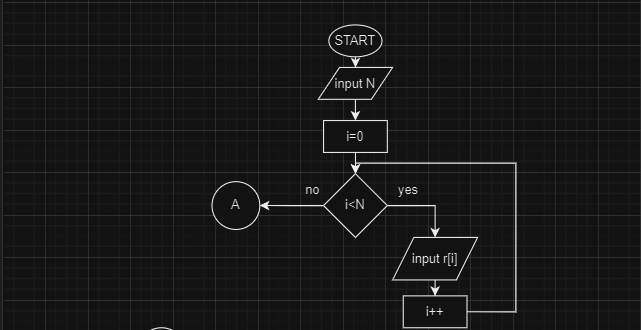
*Зображення №6. Блок-схема до програми №2 VNS Lab 5 - Task 1*

* Планований час на реалізацію: 1 год
* Важливі деталі для врахування в імплементації

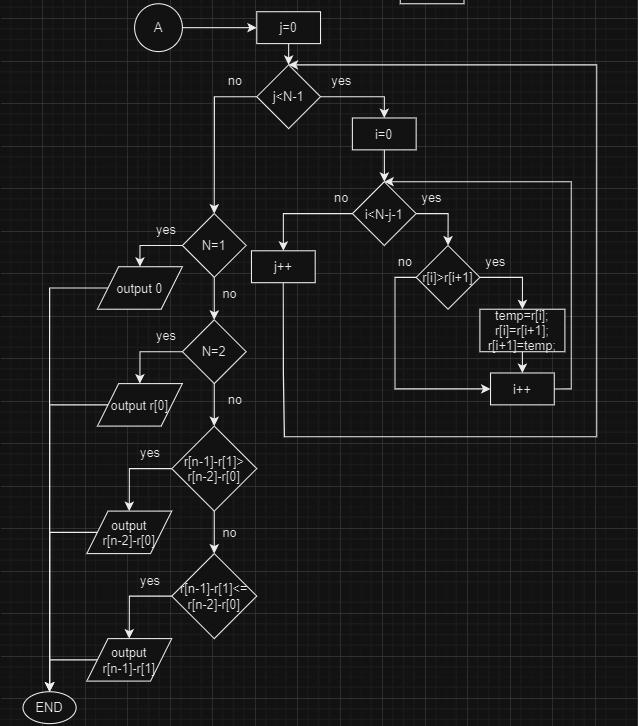
Використовував двовимірні масиви та вкладені цикли

Програма №3 Algotester Lab 2

* Блок-схема



*Зображення №7. Блок-схема до програми №3 ALgotester Lab 2*



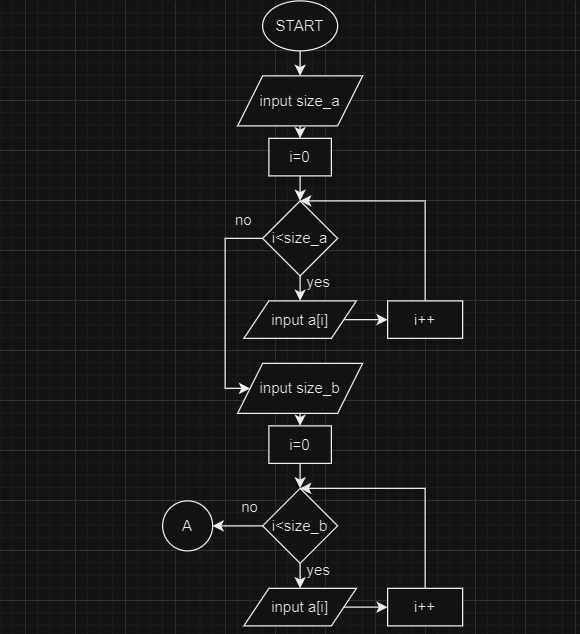
*Зображення №8. Блок-схема до програми №3 ALgotester Lab 2*

* Планований час на реалізацію: 1 год
* Важливі деталі для врахування в імплементації

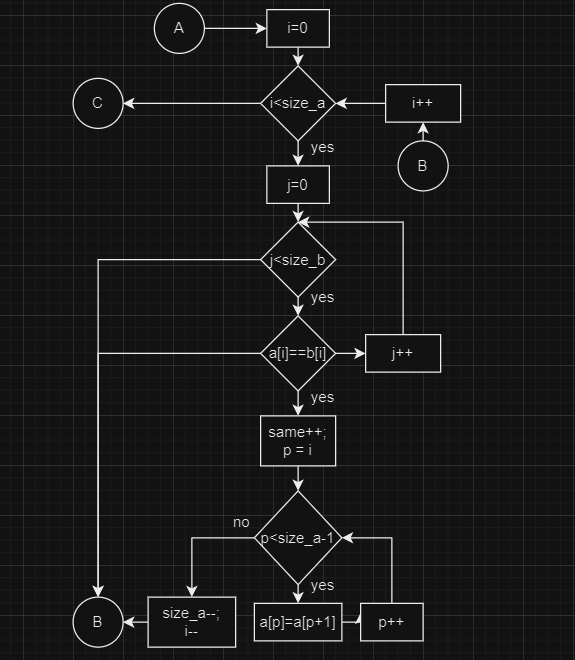
Використовував сортування

Програма №4 Algotester Lab 3

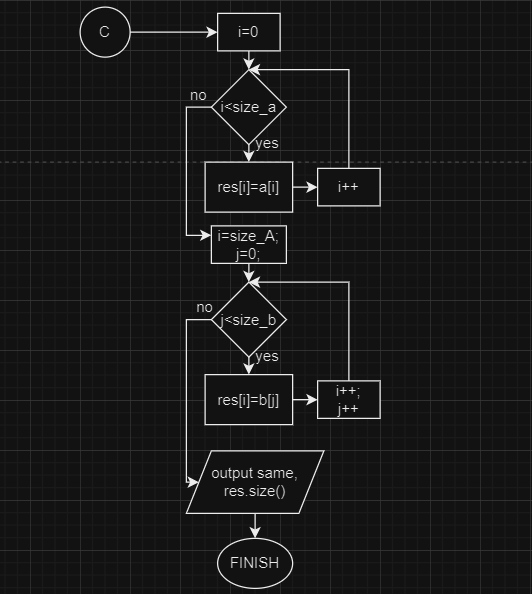
* Блок-схема



*Зображення №9. Блок-схема до програми №4 Algotester Lab 3*



*Зображення №10. Блок-схема до програми №4 Algotester Lab 3*



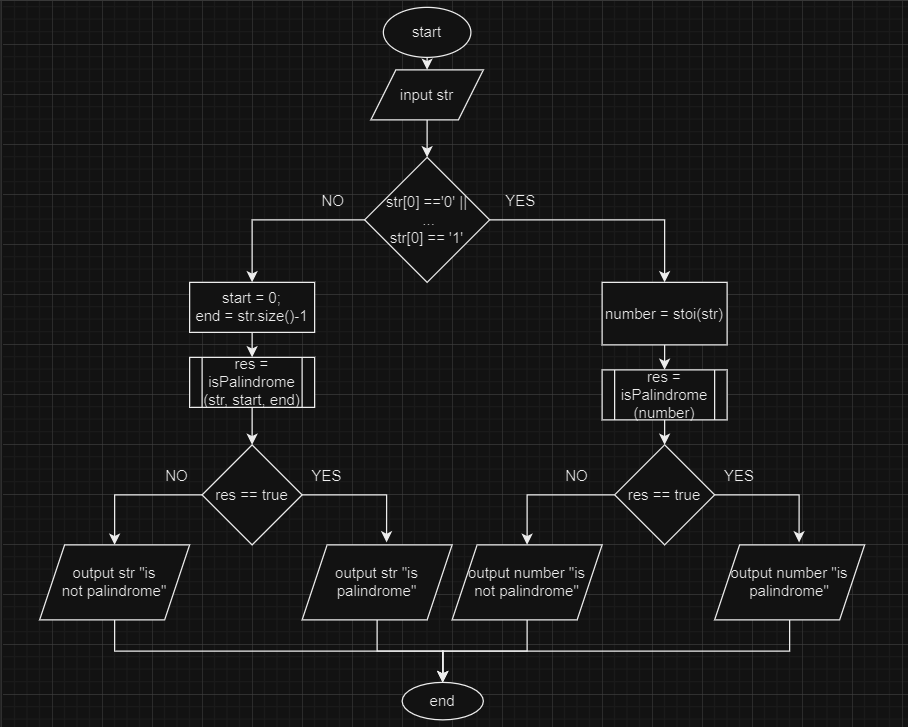
*Зображення №11. Блок-схема до програми №4 Algotester Lab 3*

* Планований час на реалізацію: 1 год
* Важливі деталі для врахування в імплементації

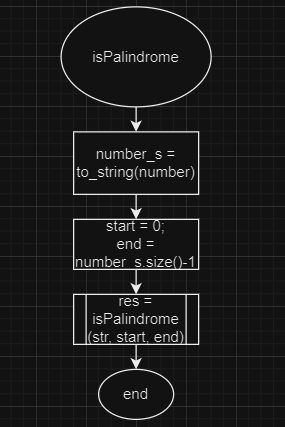
Використовував сортування та вкладені цикли

Програма №5 Class Practice Work

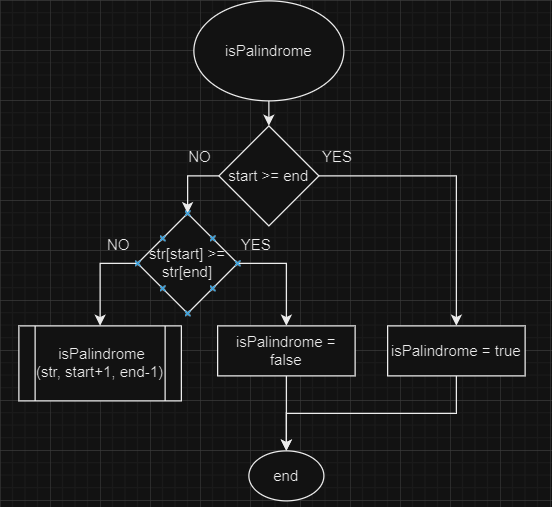
* Блок-схема



*Зображення №12. Блок-схема до програми №5 VNS Class Practice Task*



*Зображення №13. Блок-схема до програми №5 VNS Class Practice Task*



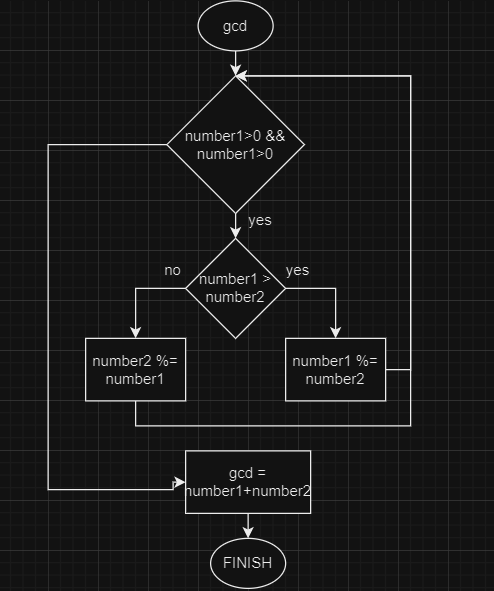
*Зображення №13. Блок-схема до програми №5 VNS Class Practice Task*

* Планований час на реалізацію: 1 год
* Важливі деталі для врахування в імплементації

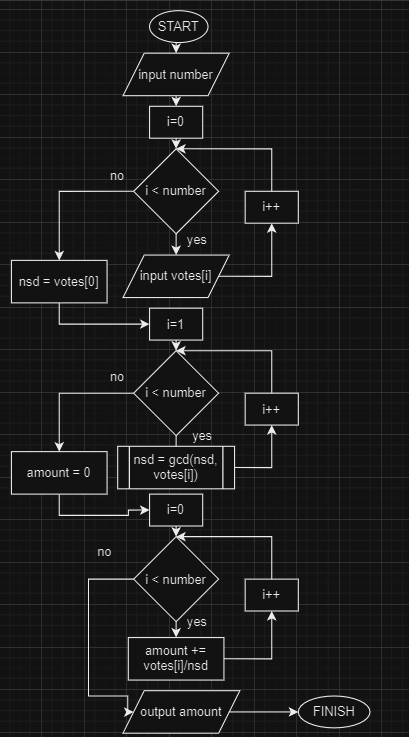
Створюю функцію логічного типу повернення для рядка, яка рекурсивно викликає себе. Організовую перевантажену функцію для

Програма №6 Self Practice Work

* Блок-схема



*Зображення №14. Блок-схема до програми №6 VNS Self Practice Task*



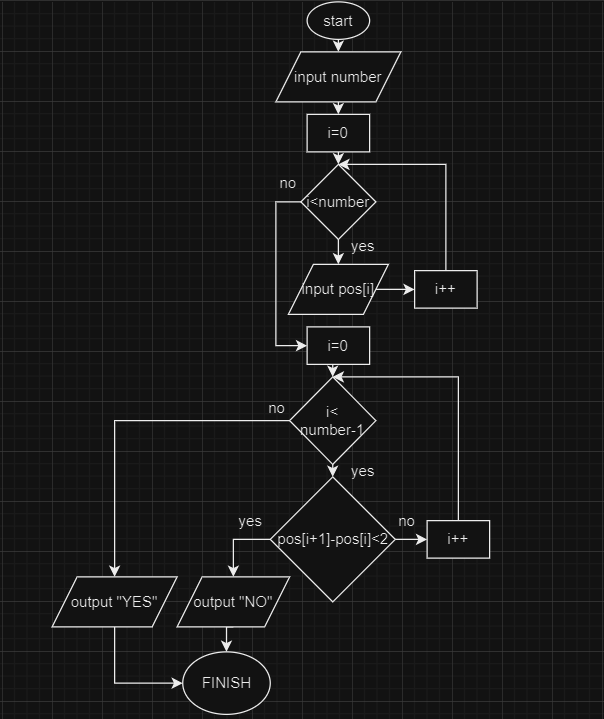
*Зображення №15. Блок-схема до програми №6 VNS Self Practice Task*

* Планований час на реалізацію: 45 хв
* Важливі деталі для врахування в імплементації

Використовую масив і створюю функцію, яка рахує найменший спільний дільник двох чисел

Програма №7 Self Practice Work

* Блок-схема



*Зображення №16. Блок-схема до програми №7 VNS Self Practice Task*

* Планований час на реалізацію: 45 хв
* Важливі деталі для врахування в імплементації

Використовую масив і цикл для перевірки дистанції

## **3. Конфігурація середовища до виконання завдань:**

Використано налаштування з попередніх лабораторних робіт.

## **4. Код програм з посиланням на зовнішні ресурси:**

Посилання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/559>

Завдання №1 VNS Lab 4 - Task 1

#include <iostream>

#include <ctime>

using *namespace* std;

*int* main(){

*int* N;

cin >> N;

*int* \* arr = new *int* [N];

srand(time(NULL));

for(*int* i = 0; i < N; i++){

arr[i] = rand()%20;

}

cout << "array: ";

for(*int* i = 0; i < N; i++){

cout << arr[i] << " ";

}

for(*int* i = 0, j = N-1; i<j ;i++, j--){

*int* tmp = arr[i];

arr[i]=arr[j];

arr[j]=tmp;

}

cout << endl << "rotated array: ";

for(*int* i = 0; i < N; i++){

cout << arr[i] << " ";

}

if(N%2 != 0){

N--;

*int* \* new\_arr = new *int* [N];

for(*int* i = 0, j=0; i<=N; i++, j++){

if(j==N/2){

i--;

} else{

new\_arr[i]=arr[j];

}

}

delete[] arr;

arr = new\_arr;

cout << endl << "array without element: ";

for(*int* i = 0; i < N; i++){

cout << arr[i] << " ";

}

}

*int* new\_size = N+3;

*int* \*final\_arr = new *int* [new\_size];

for(*int* i = 0; i < 3; i++){

final\_arr[i]=arr[i+10]-2;

}

for(*int* i = 3; i < new\_size; i++){

final\_arr[i]=arr[i-3];

}

cout << endl << "final array: ";

for(*int* i = 0; i < new\_size; i++){

cout << final\_arr[i] << " ";

}

delete[] final\_arr;

delete[] arr;

return 0;

}

*Блок №1. Код до програми №1 VNS Lab 4 - Task 1*

Завдання №2 VNS Lab 5 - Task 1

#include <iostream>

#include <ctime>

using *namespace* std;

*void* multiply(*int* *matrix*[][4], *int* *rows*, *int* *cols*){

for (*int* col = 0; col < *cols*; col++) {

*bool* cond1 = true;

for(*int* i = 0, j = 0; i < *rows* && j < *cols*; i++, j++){

if(*matrix*[0][col]<*matrix*[i][j]){

cond1 = false;

}

}

*bool* cond2 = true;

for(*int* i = *rows*-1, j = 0; i > 0 && j < *cols*; i--, j++){

if(*matrix*[0][col]<*matrix*[i][j]){

cond2 = false;

}

}

if (cond1 == true && cond2 == true){

*int* multiply = 1;

for(*int* row = 0; row < *rows*; row++){

multiply \*= *matrix*[row][col];

}

cout << multiply << " ";

} else{

cout << "nan ";

}

}

}

*int* main(){

const *int* rows = 3, cols = 4;

*int* matrix[rows][cols];

srand(time(NULL));

for(*int* i = 0; i < rows; i++){

for(*int* j = 0; j < cols; j++){

matrix[i][j] = rand() % 10;

}

}

cout << "matrix:" << endl;

for(*int* i = 0; i < rows; i++){

for(*int* j = 0; j < cols; j++){

cout << matrix[i][j] << " ";

}

cout << endl;

}

multiply(matrix, rows, cols);

return 0;

}

*Блок №2. Код до програми №2 VNS Lab 5 - Task 1*

Завдання №3 Algotester Lab 2

#include <iostream>

using *namespace* std;

*int* main(){

*int* r[100000];

*unsigned* *int* N;

cin >> N;

for (*int* i = 0; i < N; i++) {

cin >> r[i];

}

for (*int* j = 0; j < N - 1 ; j++){

for (*int* i = 0; i < N - j - 1; i++) {

if (r[i] > r[i+1]){

*int* temp = r[i];

r[i] = r[i+1];

r[i+1] = temp;

}

}

}

switch (N)

{

case 1:

cout << 0;

break;

case 2:

cout << r[0];

break;

default:

if (r[N-1] - r[1] > r[N-2] - r[0]){

cout << r[N-2] - r[0];

} else if (r[N-1] - r[1] <= r[N-2] - r[0]){

cout << r[N-1] - r[1];

}

}

return 0;

}

*Блок №3. Код до програми №3 Algotester Lab 2*

Завдання №4 Algotester Lab 3

#include <iostream>

#include <vector>

using *namespace* std;

*int* main(){

*int* size\_a, size\_b, same = 0;

cin >> size\_a;

*vector*<*int*> a(size\_a);

for(*int* i = 0; i < size\_a; i++){

cin >> a[i];

}

cin >> size\_b;

*vector*<*int*> b(size\_b);

for(*int* i = 0; i < size\_b; i++){

cin >> b[i];

}

for(*int* i = 0; i < size\_a; i++){

for(*int* j = 0; j < size\_b; j++){

if(a[i]==b[j]){

same++;

for(*int* p = i; p < size\_a - 1; p++){

a[p] = a[p+1];

}

size\_a--;

i--;

break;

}

}

}

*vector*<*int*> res(size\_a+size\_b);

for(*int* i = 0; i < size\_a; i++){

res[i]=a[i];

}

for(*int* i = size\_a, j = 0; j < size\_b; i++, j++){

res[i]=b[j];

}

cout << same << endl << res.size();

return 0;

}

*Блок №3. Код до програми №4 Algotester Lab 3*

Завдання №5 Class Practice Work

#include <iostream>

#include <string>

using *namespace* std;

*bool* isPalindrome(*string* *str*, *int* *start*, *int* *end*);

*bool* isPalindrome(*int* *number*);

*int* main(){

*string* str;

*int* start, end;

cout << "enter a word or number:" << endl;

cin >> str;

if(str[0] == '0' || str[0] == '1' || str[0] == '2' || str[0] == '3' || str[0] == '4' || str[0] == '5' || str[0] == '6' || str[0] == '7' || str[0] == '8' || str[0] == '9'){

*int* number = stoi(str);

*bool* res = isPalindrome(number);

if (res == true){

cout << number << " is palindrome" << endl;

} else {

cout << number << " is not palindrome" << endl;

}

} else{

start = 0;

end = str.size() - 1;

*bool* res = isPalindrome(str, start, end);

if (res == true){

cout << "\"" << str << "\"" << " is palindrome" << endl;

} else {

cout << "\"" << str << "\"" << " is not palindrome" << endl;

}

}

return 0;

}

*bool* isPalindrome(*string* *str*, *int* *start*, *int* *end*){

if (*start* >= *end*){

return true;

} else if (*str*[*start*] != *str*[*end*]){

return false;

} else {

return isPalindrome(*str*, *start* + 1, *end* - 1);

}

}

*bool* isPalindrome(*int* *number*){

*string* s\_number = to\_string(*number*);

*int* start = 0;

*int* end = s\_number.size() - 1;

return isPalindrome(s\_number, start, end);

}

*Блок №5. Код до програми №5 Class Practice Work*

Завдання №6 Self Practice Work

#include <iostream>

using *namespace* std;

*int* gcd(*int* *number1*, *int* *number2*)

{

while(*number1* > 0 && *number2* > 0){

if(*number1* > *number2*){

*number1* %= *number2*;

} else{

*number2* %= *number1*;

}

}

return *number1* + *number2*;

}

*int* main(){

*int* number;

*long* *long* votes[1000];

cin >> number;

for(*int* i = 0; i < number; i++){

cin >> votes[i];

}

*long* *long* nsd = votes[0];

for(*int* i = 1; i < number; i++){

nsd = gcd(nsd, votes[i]);

}

*long* *long* amount = 0;

for(*int* i = 0; i < number; i++){

amount += votes[i]/nsd;

}

cout << amount;

return 0;

}

*Блок №6. Код до програми №6 Self Practice Work*

Завдання №7 Self Practice Work

//algotester 1681 - Соціальна дистанція

#include <iostream>

using *namespace* std;

*int* main(){

*long* *long* position[10000];

*int* number;

cin >> number;

for(*int* i = 0; i < number; i++){

cin >> position[i];

}

for(*int* i = 0; i < number-1; i++){

if(position[i+1]-position[i]<2){

cout << "NO";

return 0;

}

}

cout << "YES";

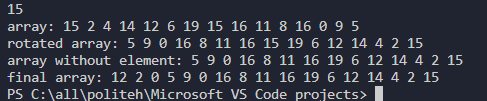
return 0;

}

*Блок №7. Код до програми №7 Self Practice Work*

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

Завдання №1 VNS Lab 4 - Task 1

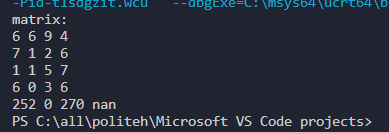


*Зображення №17. Результати виконання завдання №1 VNS Lab 4 - Task 1*

Задали масив з випадкових чисел. Оскільки розмір масиву непарний, то видалили середній елемент. Перевернули масив та добавили елементи на початок.

Час затрачений на виконання завдання: 1 год

Завдання №2 VNS Lab 5 - Task 1

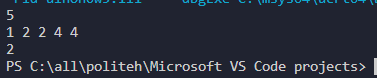


*Зображення №18. Результати виконання завдання №2 VNS Lab 5 - Task 1*

У перших трьох стовпцях перші елементи не менші за алементи на головній та бічній діагоналях, тому отримали добуток елементів цих стовпців

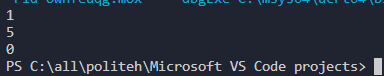
Час затрачений на виконання завдання: 1,5 год

Завдання №3 Algotester Lab 2



*Зображення №19. Результати виконання завдання №3 Algotester Lab 2*

4-1>4-2, тому відкидаємо одиницю і отримує різницю між 4 і 2

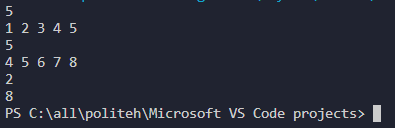


*Зображення №20. Результати виконання завдання №3 Algotester Lab 2*

був один елемент, відкинули його і наш результат - 0

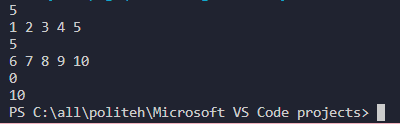
Час затрачений на виконання завдання: 1,2 год

Завдання №4 Algotester Lab 3



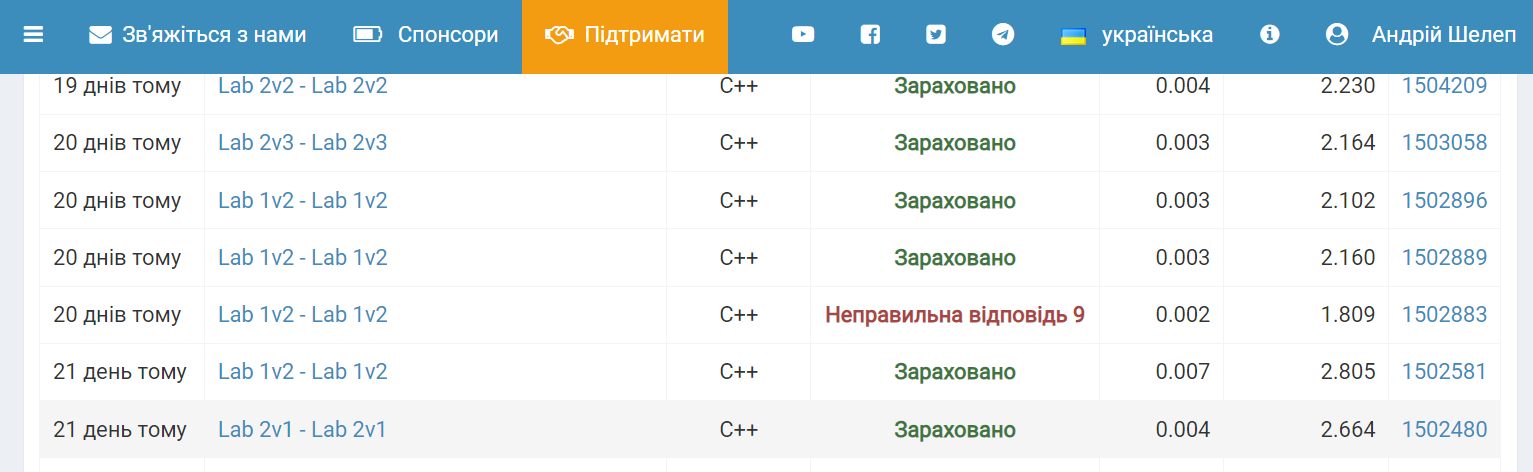
*Зображення №21. Результати виконання завдання №4 Algotester Lab 3*

2 елементи, що повторюються (4 і 5). Кількість унікальних елементів в об’єднаному масиві буде 8



*Зображення №22. Результати виконання завдання №4 Algotester Lab* *3*

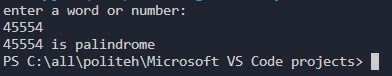
0 елементів, що повторюються



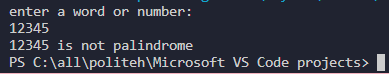
*Зображення №23. Результати виконання завдання №4 Algotester Lab 3*

Час затрачений на виконання завдання: 1,2 год

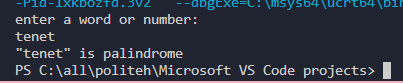
Завдання №5 Class Practice Work



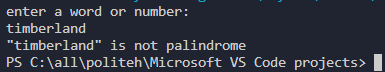
*Зображення №24. Результати виконання завдання №5 Class Practice Task*



*Зображення №25. Результати виконання завдання №5 Class Practice Task*



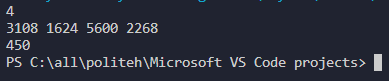
*Зображення №26. Результати виконання завдання №5 Class Practice Task*



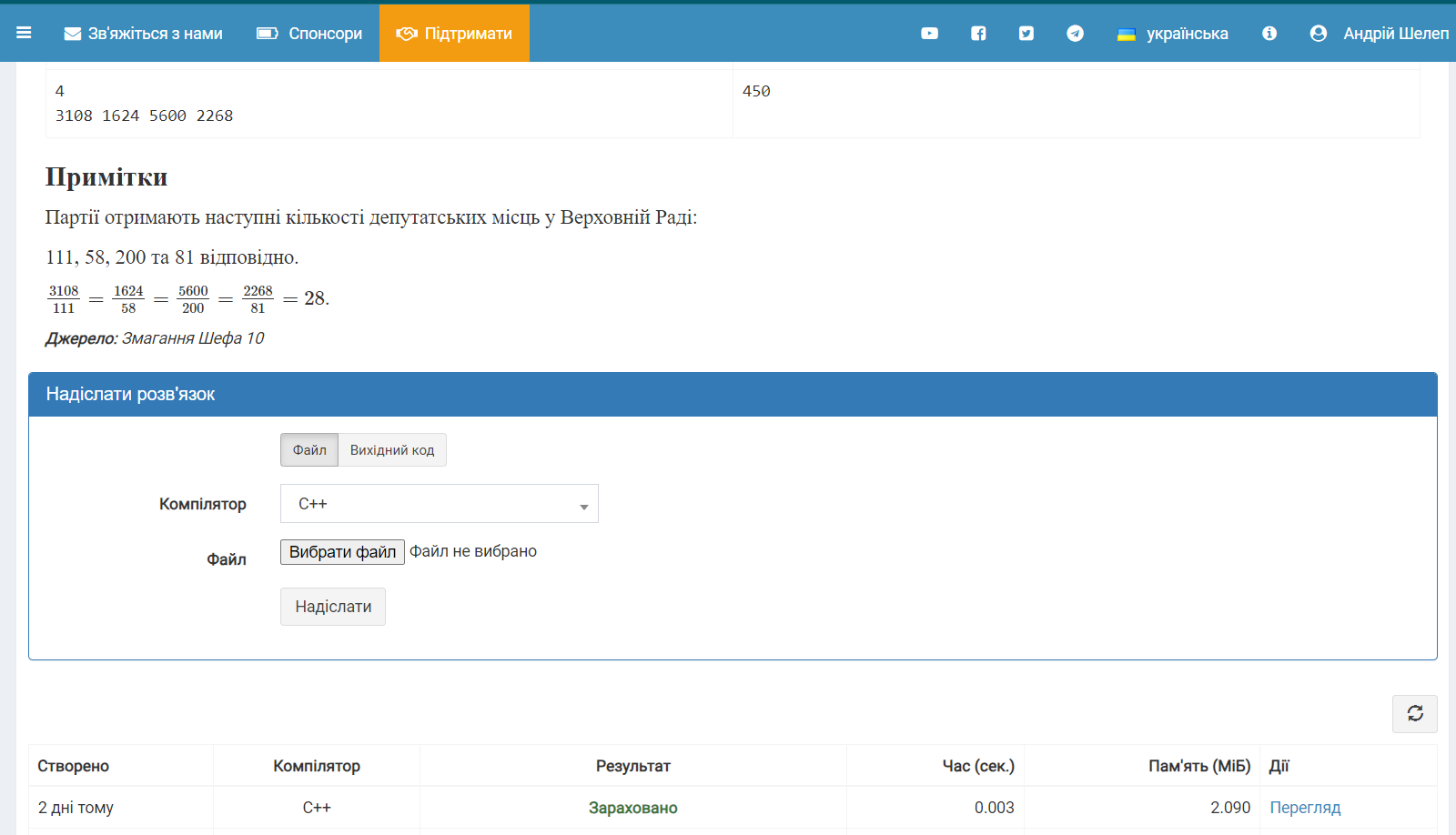
*Зображення №27. Результати виконання завдання №5 Class Practice Task*

Час затрачений на виконання завдання: 1 год

Завдання №6 Self Practice Work



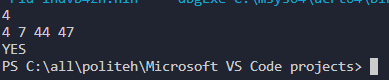
*Зображення №28. Результати виконання завдання №6 Self Practice Task*



*Зображення №29. Результати виконання завдання №6 Self Practice Task*

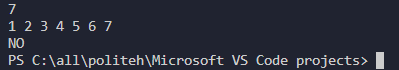
Час затрачений на виконання завдання: 1 год

Завдання №7 Self Practice Work



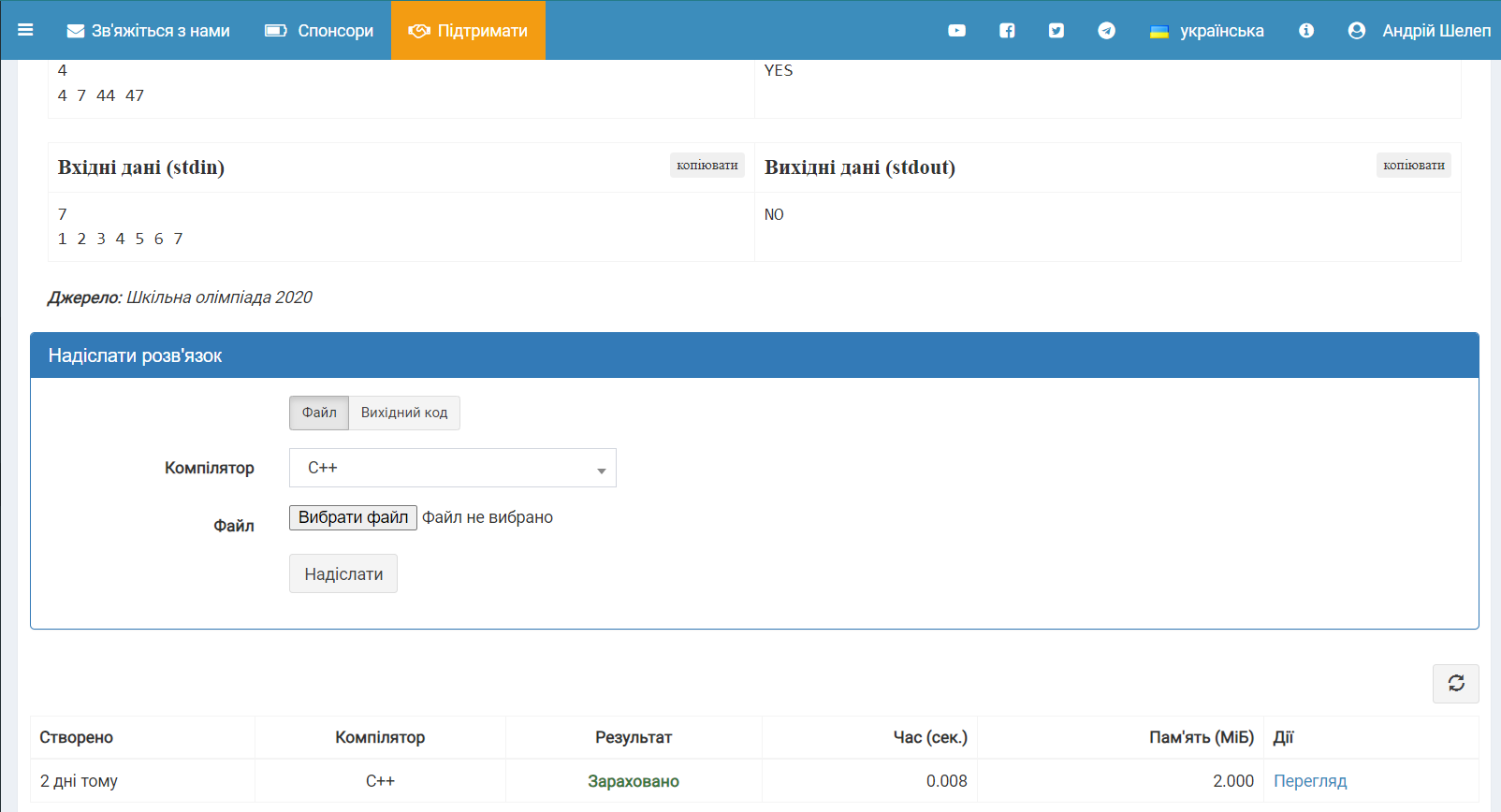
*Зображення №30. Результати виконання завдання №7 Self Practice Task*

дистанція між усіма більша 2 метрів



*Зображення №31. Результати виконання завдання №7 Self Practice Task*

є дистанція менша 2 метрів



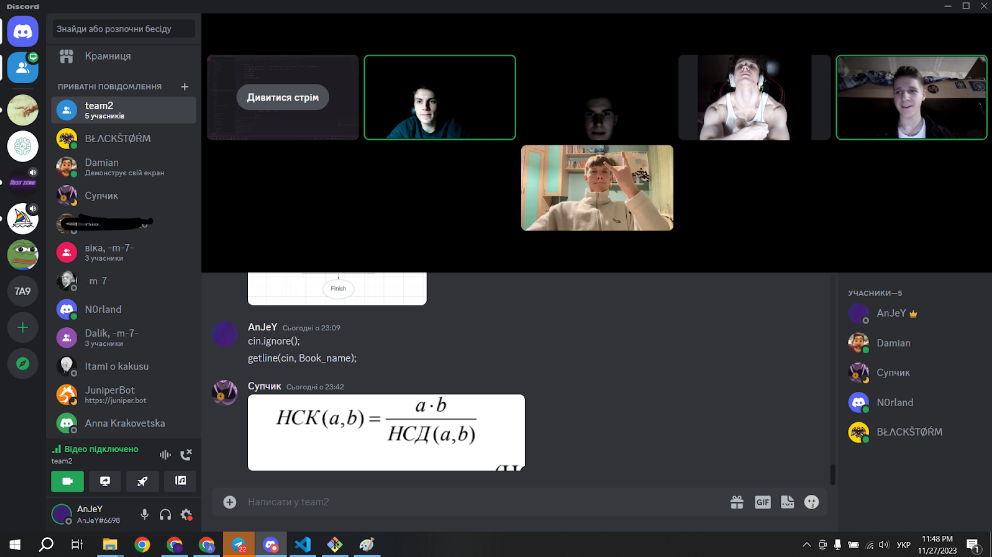
*Зображення №32. Результати виконання завдання №7 Self Practice Task*

зараховано на алготестері

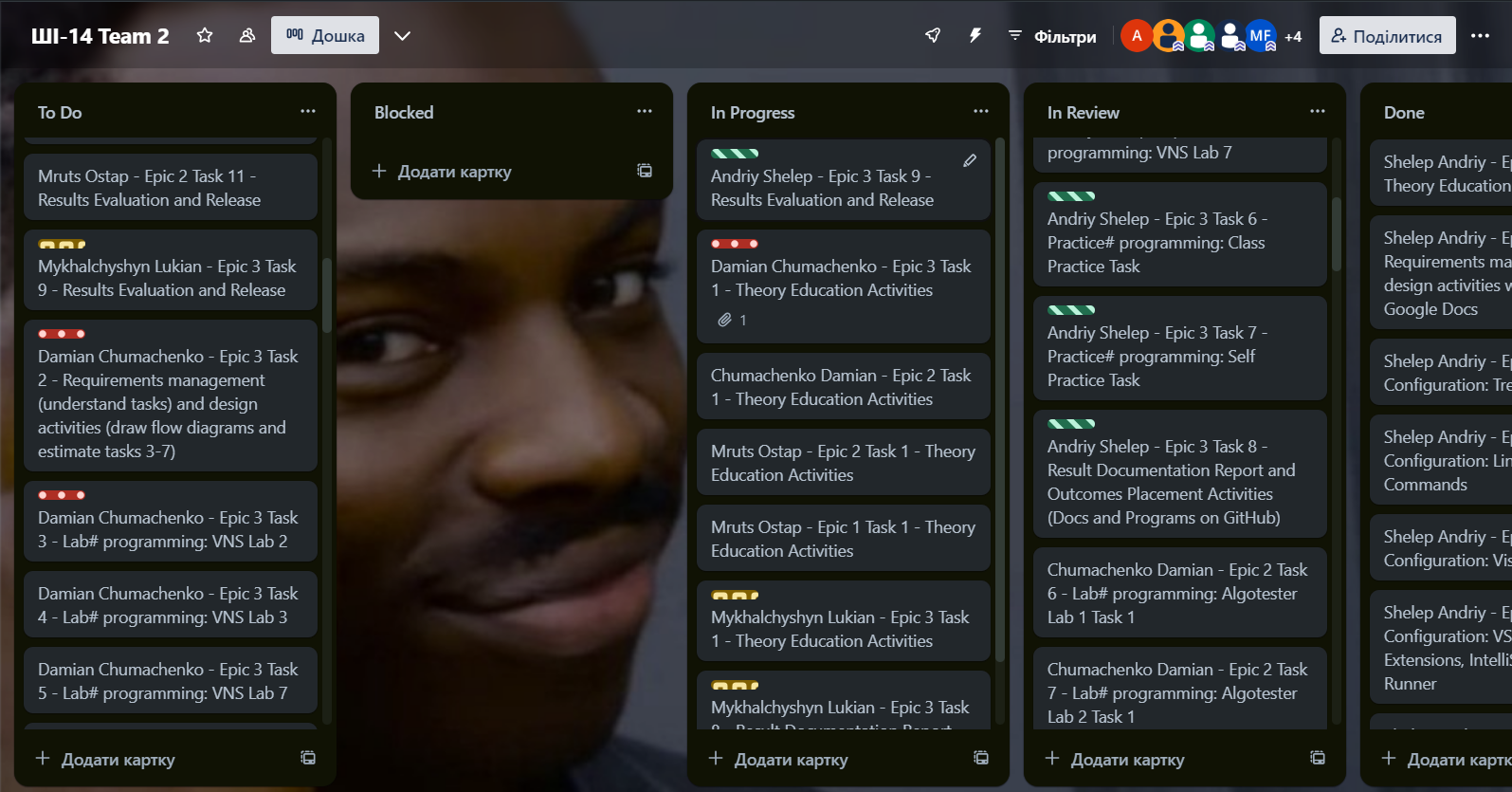
Час затрачений на виконання завдання: 20 хв

## **6. Кооперація з командою:**

* Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло

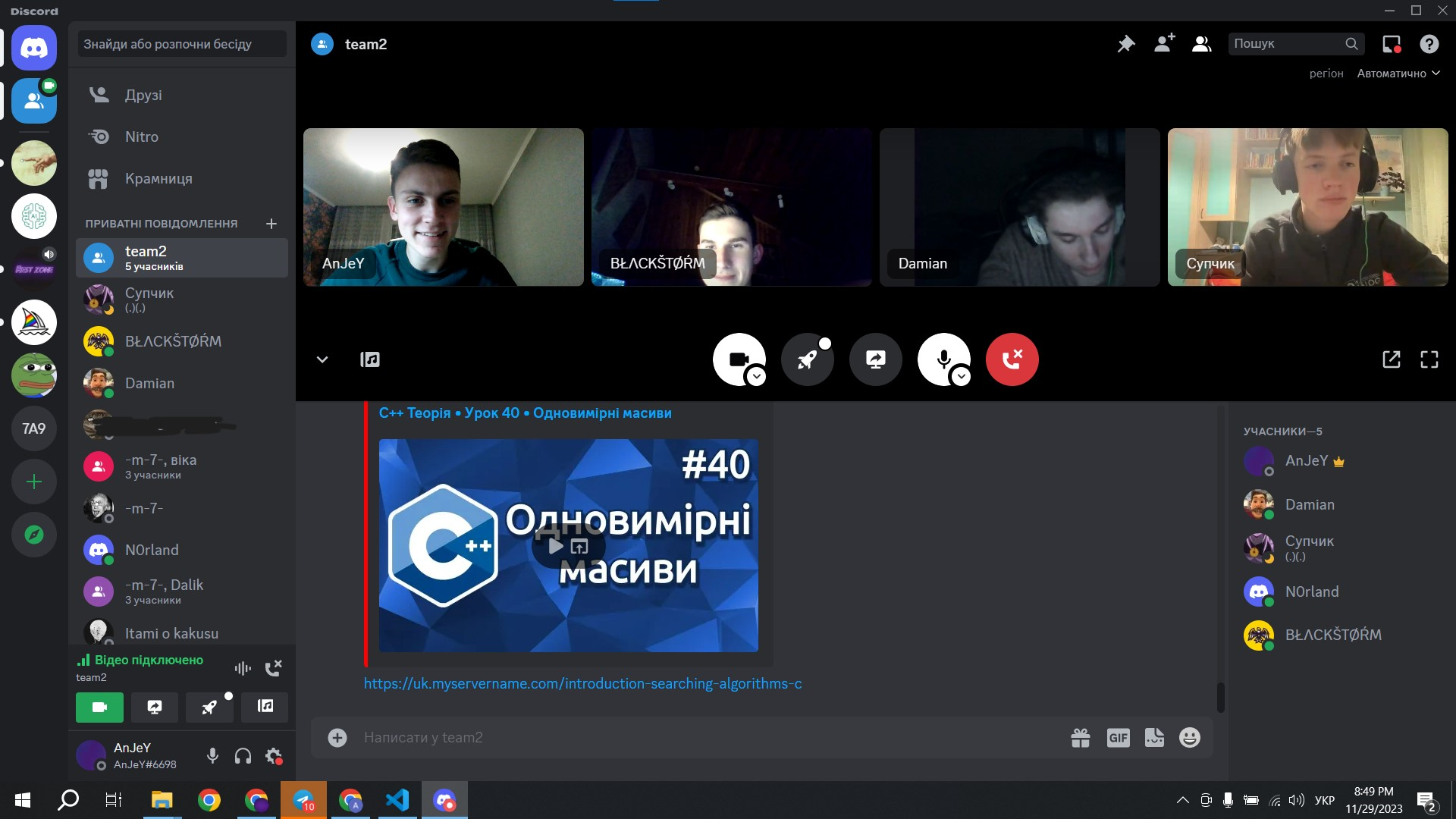


*Зображення №33. Зустріч з командою*

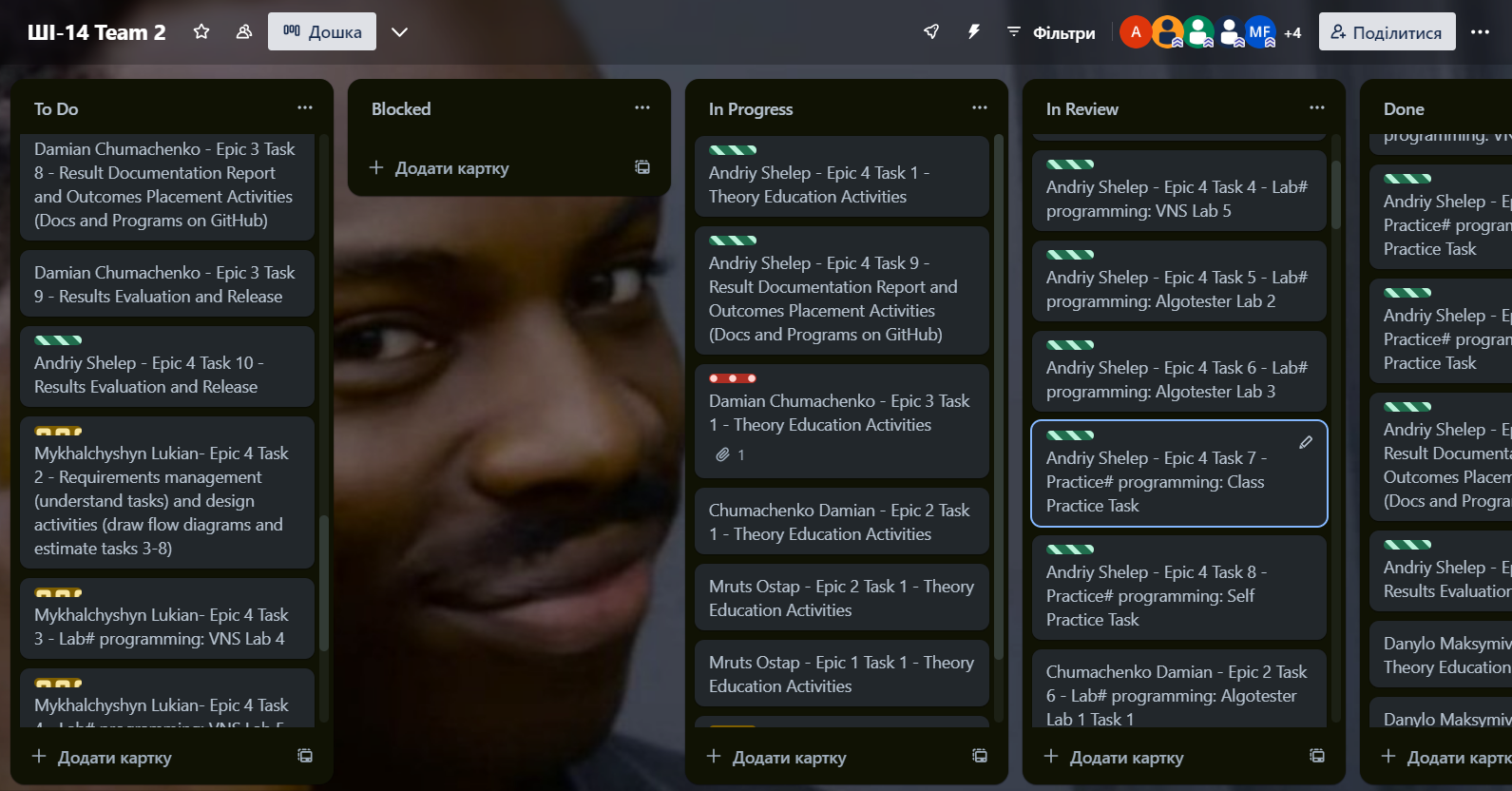


*Зображення 34. Trello*

* Скрін з 2-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло

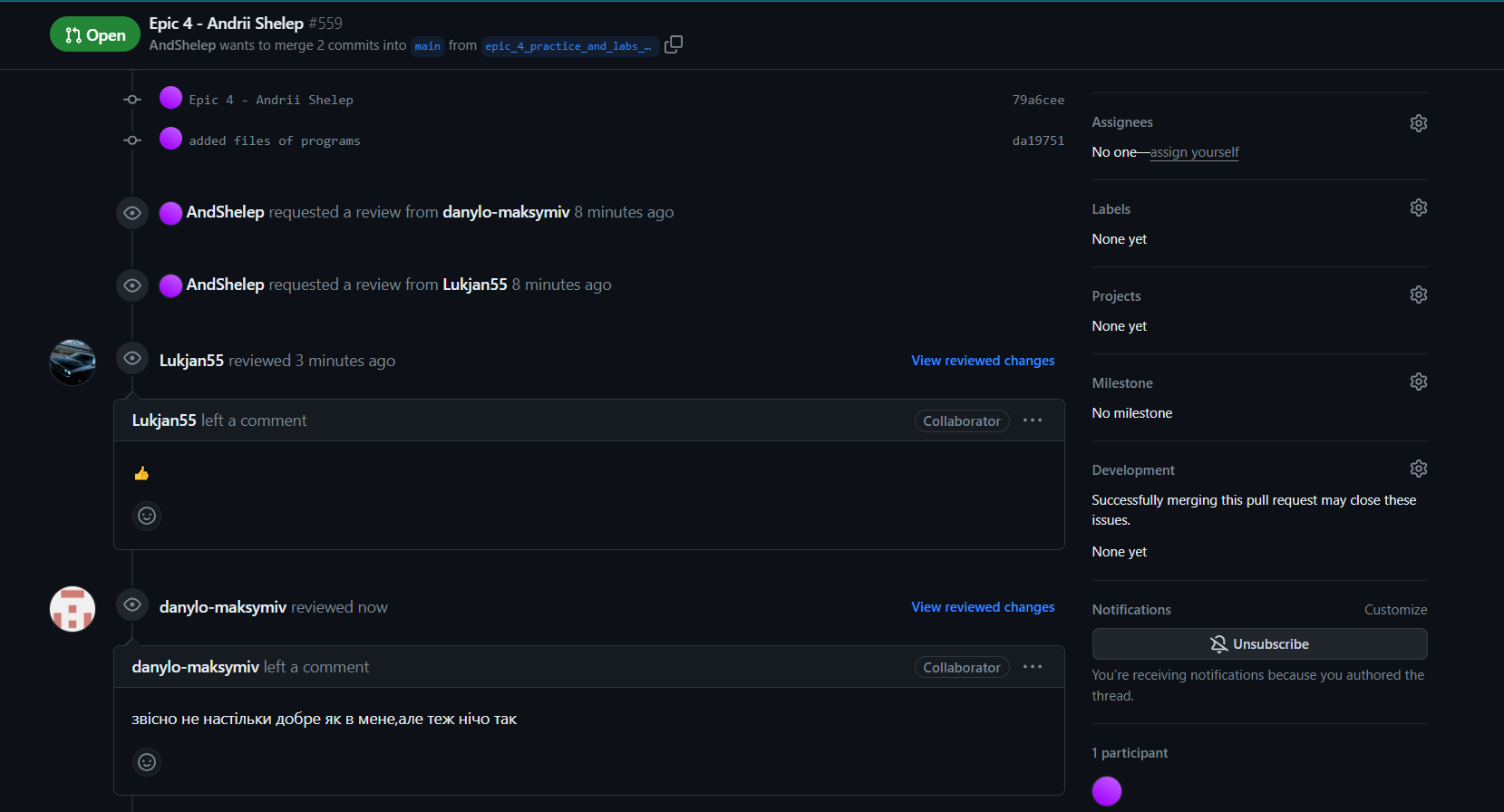


*Зображення №35. Зустріч з командою*



*Зображення 36. Trello*

* Скрін з коментарями від учасників команди на пул реквесті з Ревю Роботи



*Зображення 37**. Пул-реквест*

# **Висновки:**

Ознайомився з масивами у С++. Під час виконання лабораторних задач застосовував їх та працював з алгоритмами обробки масивів (наприклад, сортування).