Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 4**

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

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

***Виконала:***

студентка групи ШІ-13

Жмуд Анастасія

# **Тема роботи:**

Робота з одновимірними масивами. Вказівники. Функції і масиви.

# **Мета роботи:**

Одержання навичок обробки одновимірних масивів. Використання вказівників для роботи з масивами. Організувати обробку масивів з використанням функцій, навчитися передавати масиви як параметри функцій.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Одновимірні масиви
* Тема №2: Вказівники
* Тема №3: Обробка масивів за допомогою функцій

1. Індивідуальний план опрацювання теорії:

* Тема №1: Одновимірні масиви
  + Джерела Інформації
    - <https://cherto4ka.xyz/2020/01/16/massive_begin/>
    - <https://acode.com.ua/urok-77-masyvy/>
    - http://cpp.dp.ua/obrobka-odnovymirnyh-masyviv/
  + Що опрацьовано:
    - Поняття одновимірного масиву
    - Ввід та вивід масивів
    - Робота з елементами одновимірного масиву
  + Статус: Ознайомлена
  + Початок опрацювання теми: 22.11.23р
  + Звершення опрацювання теми: 23.11.23
* Тема №2: Вказівники
  + Джерела Інформації:
    - https://acode.com.ua/urok-84-vkazivnyky/
    - https://victana.lviv.ua/knyhy/konspekty-lektsii/138-alhorytmizatsiia-ta-prohramuvannia-chastyna-1/658-rozdil-7-vkazivnyky-ta-posylannia-2017-r
    - <https://acode.com.ua/urok-97-vkazivnyky-na-vkazivnyky/>
    - https://studfile.net/preview/3740813/page:3/
  + Що опрацьовано:
    - Вказівники. Операції з вказівниками
    - Роз іменування вказівників
  + Статус: Ознайомлений
  + Початок опрацювання теми: 24.11.23
  + Звершення опрацювання теми: 26.11.23
* Тема №3: Обробка масивів за допомогою функцій
  + Джерела Інформації:
    - 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/
  + Що опрацьовано:
    - Робота функції з масивом
    - Різноманітні функції для роботи з одновимірними масивами
  + Статус: Ознайомлений
  + Початок опрацювання теми: 19.11.23
  + Звершення опрацювання теми: 20.11.23

**Виконання роботи:**

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

**Завдання №1 Опрацювання теорії**

* Деталі завдання:

Опрацювання різноманітних матеріалів, аналіз та дослідження відео, статей, книг на задані теми. Організація та структурування отриманих даних для можливості ефективніше засвоювати отримані знання на практиці. Вивчення найважливіших моментів.

**Завдання №2 Перегляд вимог та проектування**

* Деталі завдання

Введення завдань в Trello, перегляд задач, та створення математичної моделі. Проектування блок-схем в Draw.io.

**Завдання №3 VNS Lab 4**

* Варіант завдання №23
* Деталі завдання:

Реалізувати з використанням масиву двонаправлене кільце, роздрукувати отриманий масив, починаючи з К-ого елемента і до К-1, Знищити з кільця всі елементи, що співпадають з його максимальним значенням. Роздрукувати отриманий масив, починаючи з К-ого елемента.

* Важливі деталі для врахування в імплементації програми:

Кожен елемент масиву повинен містити вказівники на попередній та наступний елемент

Завдання №4 **VNS Lab 5**

* Варіант завдання №23
* Деталі завдання:

Задано одновимірний масив, який складається з N цілих чисел. Сформувати на його основі двовимірний масив N x N так, щоб сума елементів у першому стовпці дорівнювала першому елементу одновимірного масиву, сума елементів у другому стовпці повинна дорівнювати другому елементу одновимірного масиву й т.ін. Нулі не використовувати.

**Завдання №5 Algotester Lab 2**

* Варіант завдання №3
* Деталі завдання:

Вам дано масив цiлих чисел розмiром N, на першiй та останнiй клiтинцi розмiщено по дрону. Вони одночасно взлiтають. На початку кожного ходу швидкiсть дрону стає рiвною значенню клiтинки, у якiй вiн знаходиться. Вони це роблять до моменту, коли трапиться одна з зазначених подiй:

Якшо 2 дрони опиняються в однiй клiтинцi - ви виводите Collision.

Якщо лiвий дрон опиниться справа вiд правого - це Miss

У випадку якщо вони зупиняться один навпроти одного, тобто у клiтинках ai та ai+1 -виведiть Stopped

* Важливі деталі для врахування в імплементації програми:

Необхідно перевіряти чи не сталася одна з цих ситуацій до вильоту дронів.

**Завдання №6 Algotester Lab 3**

* Варіант завдання №2
* Деталі завдання:

Вам дано 2 масиви розмiром N та M. Значення у цих масивах унiкальнi.

Завдання вивести у першому рядку кiлькiсть елементiв, якi наявнi в обох масивах одночасно, у другому кiлькiсть унiкальних елементiв в обох масивах разом.

**Завдання №7 Class Practice Task**

* Деталі завдання:

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

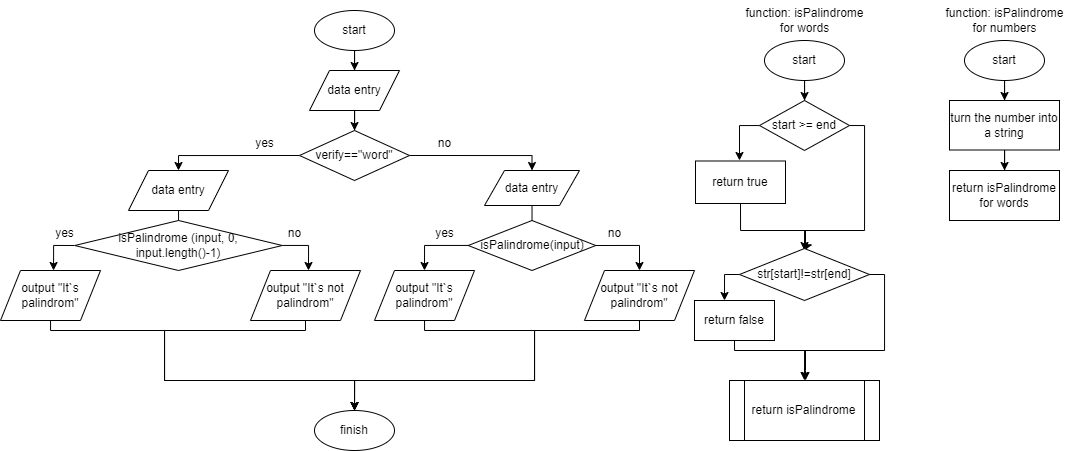
* Важливі деталі для врахування в імплементації програми:

Реалізувати рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.

Використання перевантаження функцій та рекурсії.

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

**Програма №1 Class Practice Task**

* Блок-схема :
* Планований час на реалізацію: 20 хв
* Важливі деталі для врахування в імплементації:

Реалізувати рекурсивну функцію isPalindrome, яка перевіряє, чи заданий рядок є паліндромом. Використання перевантаження функцій та рекурсії.

**Програма №2 Algotester Lab 3**

* Планований час на реалізацію: 30 хв

**Програма №3 Algotester Lab 2**

* Планований час на реалізацію: 2 години
* Важливі деталі для врахування в імплементації:

Необхідно перевіряти чи не сталася одна з зазначених ситуацій до вильоту дронів.

**Програма №4 VNS Lab 5**

* Планований час на реалізацію: 1 година

**Програма №5 VNS Lab 4**

* Планований час на реалізацію: 5 годин
* Важливі деталі для врахування в імплементації:

Кожен елемент масиву повинен містити вказівники на попередній та наступний елемент.

## 

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

**Завдання №1 Class Practice Task**

#include <iostream>

#include <string>

#include <algorithm>

using namespace std;

bool isPalindrome(const string& str, int start, int end) {

    if (start >= end) {

        return true;

    }

    if (str[start] != str[end]) {

        return false;

    }

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

}

bool isPalindrome(int number) {

    string strNum = to\_string(number);

    return isPalindrome(strNum, 0, strNum.length() - 1);

}

int main() {

    string verify;

    cout << "What do you want to verify, word or number? ";

    cin >> verify;

    if(verify=="word"){

        string input;

        cout<<"Enter a word: ";

        cin>>input;

         if (isPalindrome(input, 0, input.length() - 1)) {

        cout << "It's palindrom.\n";

    } else {

        cout << "It's not palindrom.\n";

    }

    }

    else{

        int input;

        cout<<"Enter a number:  ";

        cin>>input;

         if (isPalindrome(input)) {

        cout << "It's palindrom.\n";

    } else {

        cout << "It's not palindrom.\n";

    }

    }

    return 0;

}

Першим етапом роботи є підключення бібліотек «iostream», «string» та «algorithm», що містять в собі використовувані в коді програми функцій. Далі, оголошуємо функцію isPalindrome

Підпис та № до блоку з кодом програми в якій за допомогою першої умови перевіряє крокоммо чи ще не досягли кінця слова, а в другій – чи рівні символи (чи є слово паліндромом). Функція повертає власне значення, яке буде остаточним тільки коли слово закінчиться.

Наступним кроком перевантажуємо цю ж функцію для чисел і в ний перетворюємо число на рядок та повертаємо значення функції для рядків для цього числа.

В функції main() спочатку визначаємо чи буде елемент для перевірки словом чи числом – після, за допомогою умов викликаємо функцію isPalindrom, та після того як вона поверне значення виводимо, чи слово(число) є паліндромом чи ні.

**Завдання №2 Algotester Lab 3**

#include <iostream>

using namespace std;

int main(){

    int n, m, k=0, k1=0;

    cin>>n;

    int array1[n];

    for(int i=0; i<n; i++){

        cin>>array1[i];

    }

    cin>>m;

    int array2[m];

    for(int j=0; j<m; j++){

        cin>>array2[j];

    }

    for(int i=0; i<n; i++){

        for(int j=0; j<m; j++){

            if(array1[i]==array2[j]){

                k++;

            }

        }

    }

   cout<<k<<endl<<((n+m)-k);

}

Спочатку під’єднуємо бібліотеку «iostream», далі, в функції main() оголошуємо усі необхідні змінні та вводимо їх. Потім в оголошуємо цикл в циклі (for) та рахуємо скільки є спільних елементів у масивах. Після, виводимо це значення та віднімаємо від загальної кількості кількість спільних й отримуємо число від’ємних елементів. Його теж виводимо.

**Завдання №3 Algotester Lab 2**

#include <iostream>

using namespace std;

int main(){

    int N, first=0;

    cin>>N;

    int second = N-1;

    int\* array=new int[N];

    for ( int i=0; i<N; i++){

        cin>>array[i];

    }

    while(true){

        if(first == (second - 1)){

            cout<<++first<<" "<<++second<<endl;

            cout<<"Stopped";

            break;

        }

        if( first == second){

            cout<<++first<<" "<<++second<<endl;

            cout<<"Collision";

            break;

        }

        if(first>second){

            cout<<++first<<" "<<++second<<endl;

            cout<<"Miss";

            break;

        }

        first += array[first];

        second -= array[second];

    }

    delete[] array;

    return 0;

}

Преш за все підключаємо бібліотеку “iostream” для роботи з усім необхідними нам функціями. Після, в функції main() вводимо усі необхідні нам змінні та оголошуємо цикл, який виконується поки не буде виконана одна з умов. В ньому, для початку перевіряємо чи не виконуються усі згадані в умові задачі, ситуації, після переміщуємо дрони на задані значення.

**Завдання №4 VNS Lab 5**

#include <iostream>

using namespace std;

int function(int n, int array[]){

    int array1 [n][n];

    int sum=0;

    for( int j=0; j<n; j++){

        sum=0;

        for(int i=0; i<n; i++){

            if(i==n-1){

                array1[i][j]=array[j]-sum;

            } else{

            array1[i][j]=rand();

            if(array1[i][j]==0){

                --i;

            }

            sum+=array1[i][j];

            }

        }

    }

    for( int i=0; i<n; i++){

        cout<<endl;

        for(int j=0; j<n; j++){

            cout<<array1[i][j]<<"   ";

        }

    }

}

int main() {

    int n;

    cout<<"Enter the number of array numbers:  ";

    cin>>n;

    int array[n];

    cout<<"Enter the array:  ";

    for(int i=0; i<n; i++){

        cin>>array[i];

    }

    function(n, array);

}

Для початку під’єднуємо бібліотеку «iostream». Далі йде оголошення функції “function” яка формує двовимірний масив за всіма заданими умовами. В ній, ми оголошуємо 2 цикли, один з яких є вкладеним. Спочатку проходимось по стовпцях і заповнюємо їх, а потім по рядках. Після, виводимо отриманий масив. В функції main() вводимо одновимірний масив та викликаємо “function”.

**Завдання №5 VNS Lab 4**

#include <iostream>

using namespace std;

struct Node{

    int data;

    struct Node\* prev;

    struct Node\* next;

};

void output1(struct Node\* element) {

    for (int i = 0; i < 7; i++)

    {

        cout<<element->data<<" ";

        element = element->prev;

    }

    cout<<endl;

}

void del(struct Node\* array[5], int value) {

    int k=0, n=0;

    struct Node\* arr1[5];

    for(int i=0; i<5; i++){

    if(array[i]->data==value){

            delete array[i];

            array[i] = nullptr;

            k++;

    }

        if(array[i]!=NULL){

        arr1[n]=array[i];

        cout<<array[i]->data<<" ";

        array[i] = array[i]->next;

        n++;}

        if(i==4) break;

   }

   for(int i=0; i<2; i++){

    cout<<arr1[i]->data<<" ";

   }

}

int main(){

    int max=0;

    cout<<"Enter array:  ";

    struct Node\* array[5];

    for(int i=0; i<5; i++){

        int k=0;

        array[i]=NULL;

        array[i]=(struct Node\*)malloc(sizeof(struct Node));

        cin>>k;

        array[i]->data=k;

        if(array[i]->data>max){

            max=array[i]->data;

        }

    }

    struct Node\* head=array[0];

    for(int i=0; i<5; i++){

        if(i==0){

            array[i]->prev=array[4];

            array[i]->next=array[i+1];

        } else if(i==4){

            array[i]->prev=array[i-1];

            array[i]->next=array[0];

        } else{

            array[i]->prev=array[i-1];

            array[i]->next=array[i+1];

        }

    }

    int k=0;

    output1(array[4]);

    del(array, max);

    cout<<endl;

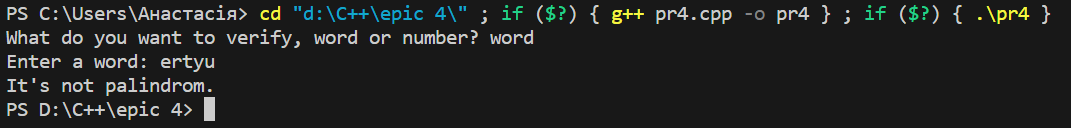
}

Для початку роботи потрібно під’єднати бібліотеку «iostream». Наступним етапом є оголошення структури Node, яка містить дані, та посилання. Після, в функції output1 виводимо кільце. Функція del видаляє елементи кільця, що мають таке ж значення, як максимальний елемент та виводить кільце в заданому порядку та кількості елементів.

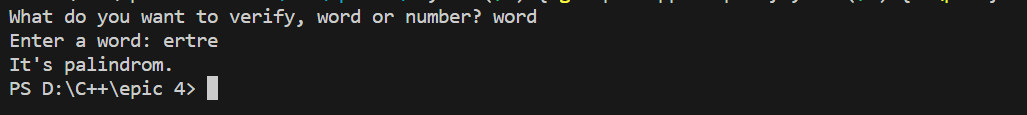
В функції main() вводимо масив структур та формуємо двонаправлене кільце. Після, по черзі викликаємо згадані раніше функції, спочатку – output1, потім – del.

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

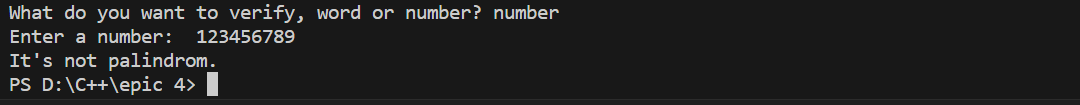
**Завдання № 1 Class Practice Task**

****

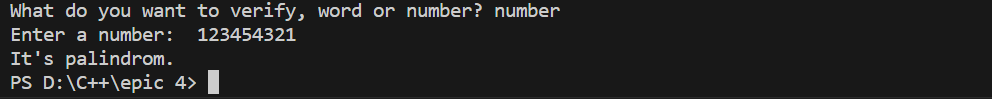
Тест №1



Тест №2



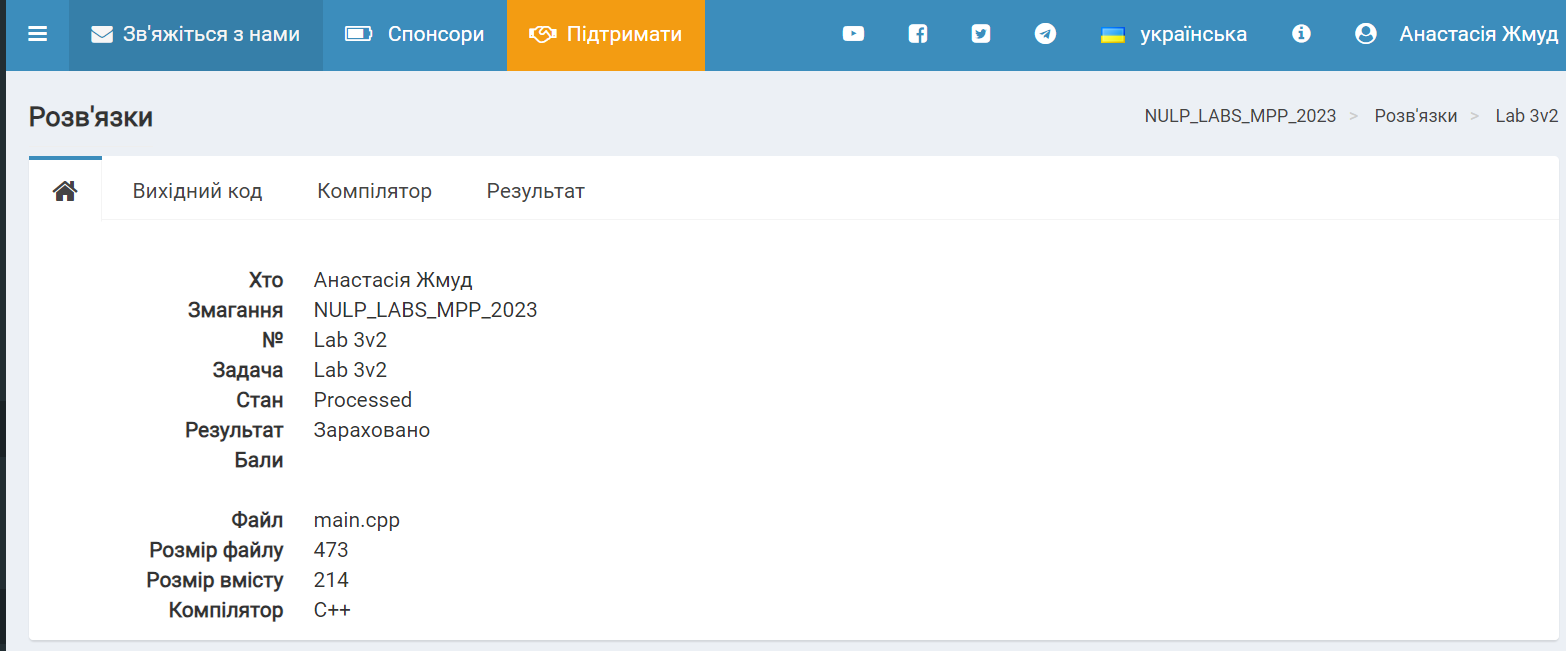
Тест №3



Тест №4

Час затрачений на виконання завдання: 1 година

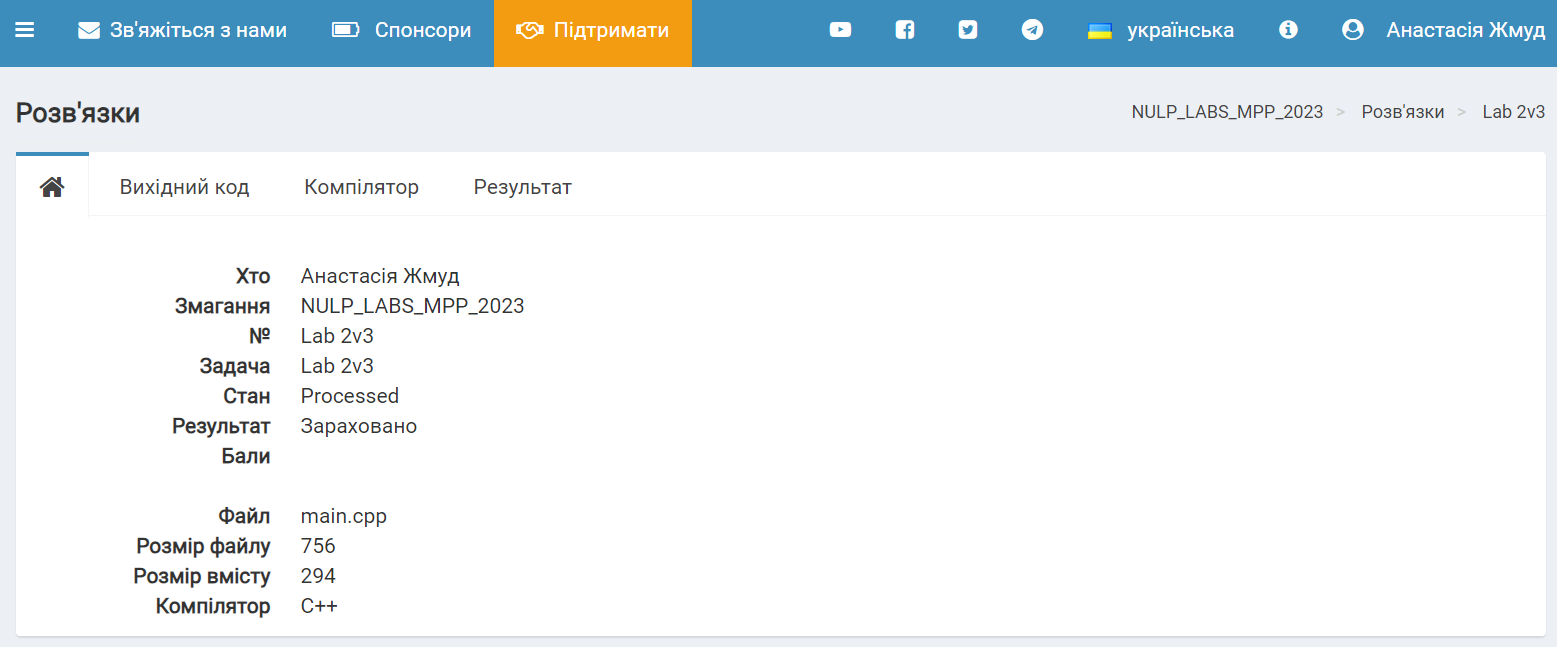
**Завдання №2 Algotester Lab 3**

****

Зарахування в алготестері

Час затрачений на виконання завдання: 30 хв

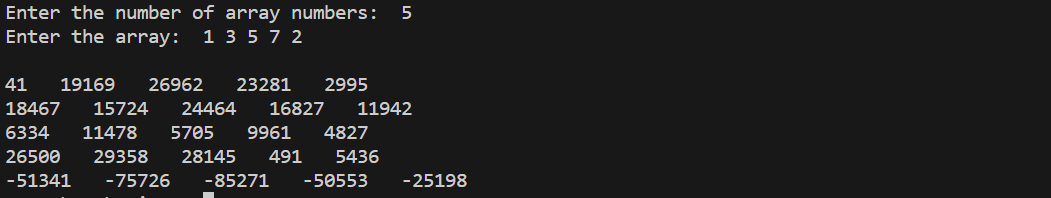
**Завдання №3 Algotester Lab 2**



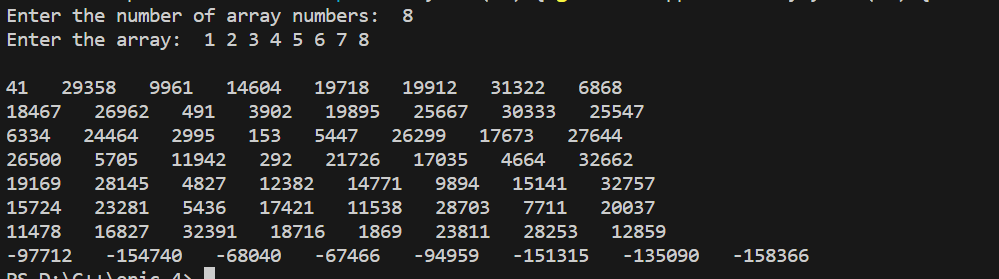
Зарахування в алготестері

Час затрачений на виконання завдання: 2 години

**Завдання №4 VNS Lab 5**



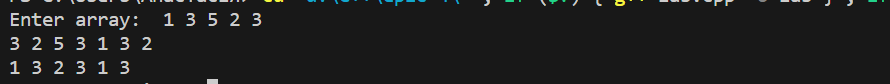
Тест №1



Тест №2

Час затрачений на виконання завдання: 30 хв.

**Завдання №5 VNS Lab 4**



Тест №1

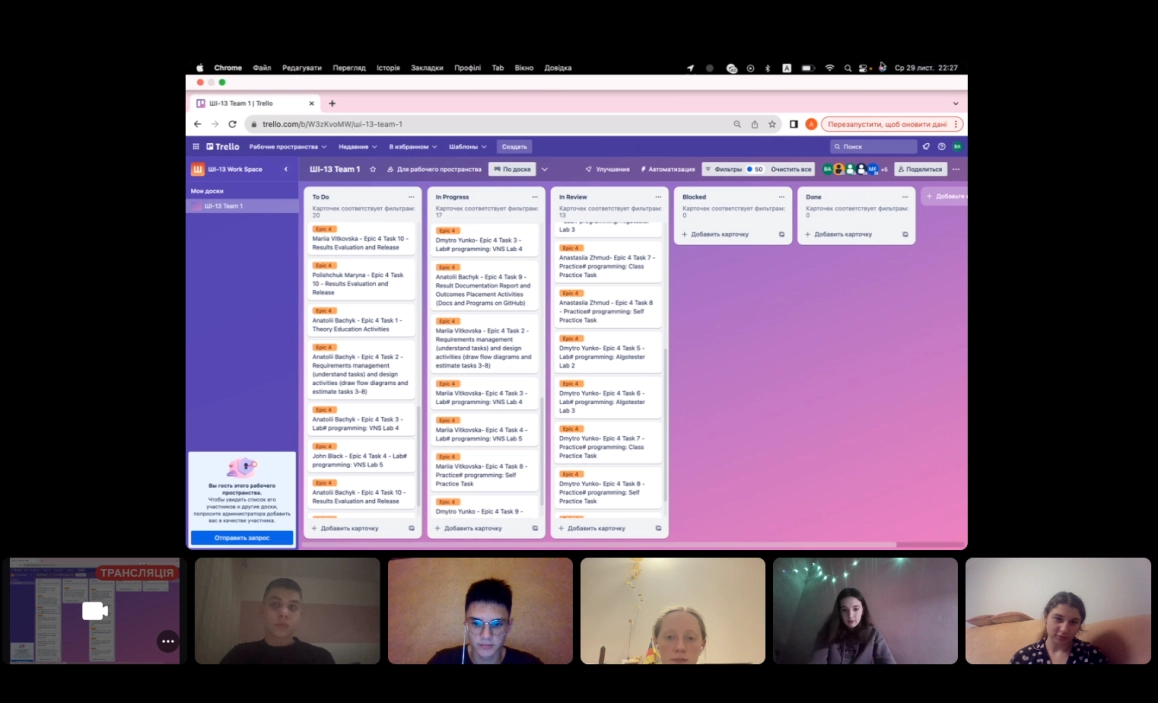


Тест №2

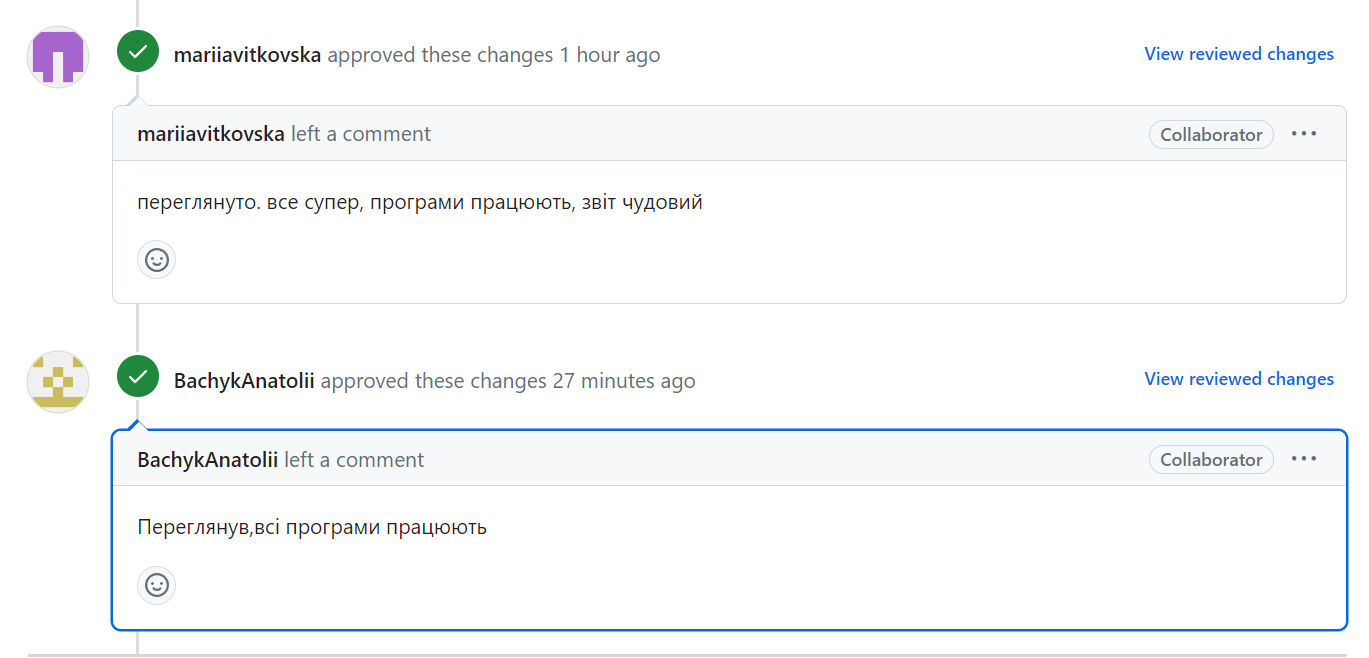
Час затрачений на виконання завдання: 6 годин

## **6. Кооперація з командою:**

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



* Скрін з 2-му коментарями від учасників команди на пул реквесті з Ревю Роботи:



# **Висновки:**

В ході роботи над цим епіком я успішно оволоділа навичками обробки одновимірних масивів та використання вказівників для маніпулювання ними. Мета була досягнута завдяки використанню функцій для ефективної обробки масивів і передачі масивів як параметрів у функції. Теоретичні відомості, які я отримала під час цієї роботи, включають основи одновимірних масивів, роботу з вказівниками та методи обробки масивів за допомогою функцій. Окрім всього, я отримала практичні навички роботи з масивами, що є важливим елементом в програмуванні.