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***: «**Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами**»

***Виконала:***

студентка групи ШІ-11

Труш Соломія Володимирівна

# **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:**

Розуміти що таке масиви, зокрема, одновимірні, двовимірні та динамічні масиви, а також структури даних і вкладені структури, знати що і розуміти алгоритми обробки. Навчитись використовувати засвоєні знання.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: **Масиви. Одновимірні та двовимірні масиви. Динамічні масиви.**
* Тема №2: **Структури. Вкладені структури.**
* Тема №3: **Алгоритми обробки та робота з масивами та структурами.**

1. Індивідуальний план опрацювання теорії:

* Тема №1: **Масиви. Одновимірні та двовимірні масиви. Динамічні масиви.**
  + Джерела Інформації
    - Відео 1. <https://www.youtube.com/watch?v=ENDaJi08jCU>
    - Відео 2. <https://www.youtube.com/watch?v=PocJ5jXv8No&ab_channel=TheCherno>
    - Стаття 1. <https://www.programiz.com/cpp-programming/arrays>
    - Стаття 2. <https://www.programiz.com/cpp-programming/multidimensional-arrays>
    - Лекційний матеріал з ВНС
    - [chat.openai.com](https://chat.openai.com/) .
  + Що опрацьовано:
    - Розумію, що таке масиви, основні принципи роботи з масивами
  + Статус: Ознайомлений
  + Початок опрацювання теми: 27.11
  + Звершення опрацювання теми: 30.11
* Тема №2: **Структури. Вкладені структури.**
  + Джерела Інформації:
    - Відео. <https://www.youtube.com/watch?v=vLnPwxZdW4Y>
    - Стаття 1. <https://acode.com.ua/urok-64-struktury/>
    - Стаття 2.<https://www.programiz.com/cpp-programming/structure>
    - Лекційний матеріал з ВНС
    - [chat.openai.com](https://chat.openai.com/) .
  + Що опрацьовано:
    - Розумію, що таке структури, для чого вони використовуються
  + Статус: Ознайомлений
  + Початок опрацювання теми: 03.12
  + Звершення опрацювання теми: 05.12
* Тема №3: **Алгоритми обробки та робота з масивами та структурами.**
  + Джерела Інформації:
    - Відео. <https://www.youtube.com/watch?v=xli_FI7CuzA>
    - Стаття 1. <https://www.programiz.com/dsa/bubble-sort>
    - Стаття 2. <https://www.programiz.com/dsa/sorting-algorithm>
    - Лекційний матеріал з ВНС
    - [chat.openai.com](https://chat.openai.com/) .
  + Що опрацьовано:
    - Ознайомлена з алгоритмами обробки та роботою з масивами та структурами
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.12
  + Звершення опрацювання теми: 03.12

# **Виконання роботи:**

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

Завдання **VNS Lab 4 - Task 1-N**

* **Варіант 11**
* Деталі завдання
* Сформувати одновимірний масив цілих чисел, використовуючи генератор

випадкових чисел.

* Роздрукувати отриманий масив.
* Знищити 5 останніх елементів масиву.
* Додати в початок масиву 3 елементи зі значенням M[I+1]+2.
* Роздрукувати отриманий масив.
* Важливі деталі для врахування в імплементації програми

Завдання **VNS Lab 5 - Task 1-N**

* **Варіант 11**
* Деталі завдання

Написати функцію, яка перевіряє чи по зростанню або спаданню

впорядкований зазначений рядок двовимірного масиву. Впорядкувати по

зростанню всі рядки двовимірного масиву, які не впорядковані по спаданню.

* Важливі деталі для врахування в імплементації програми

Завдання **Algotester Lab 2v1**

* Деталі завдання

У вас є дорога, яка виглядає як N чисел.

Після того як ви по ній пройдете - вашу втому можна визначити як різницю максимального та мінімального елементу.

Ви хочете мінімізувати втому, але все що ви можете зробити - викинути одне число з дороги, тобто забрати його з масиву.

В результаті цієї дії, яку мінімальну втому ви можете отримати в кінці дороги?

* Важливі деталі для врахування в імплементації програми

1≤ N ≤ 10^5

0≤ ri ≤ 10^5

Завдання **Algotester Lab 3v3**

* Деталі завдання

Зробити компресію стрічки, тобто якщо якась буква йде більше одного разу підряд у стрічці замінити її на букву + кількість входжень підряд.

* Важливі деталі для врахування в імплементації програми

1≤ S ≤ 10^5

Завдання **Class Practice Work**

* Деталі завдання

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

Визначте та реалізуйте рекурсивну функцію isPalindrome для рядків.

Визначте та реалізуйте перевантажену функцію isPalindrome для цілих чисел. Перетворити ціле число на рядок і використовуйте рядкову версію функції, щоб перевірити, чи це паліндром.

* Важливі деталі для врахування в імплементації програми

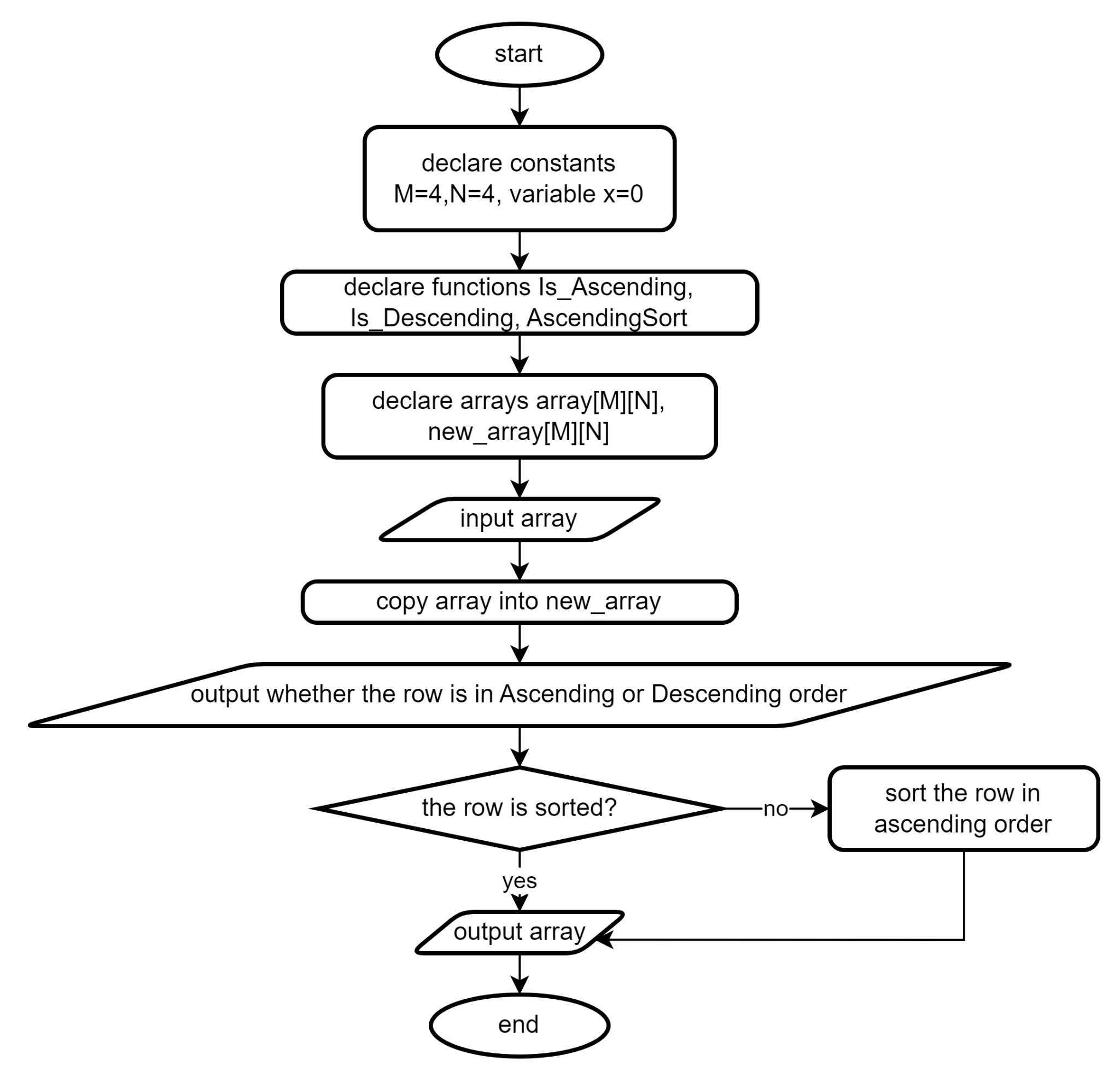
Завдання **Self Practice Work**

* Деталі завдання
* Важливі деталі для врахування в імплементації програми

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

Програма **VNS Lab 5 - Task 1-N**

* Блок-схема



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

Середовище сконфігуровано під час виконання Epic 1.

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

Завдання **VNS Lab 4 - Task 1-N** варіант 11

Посилання на файл програми у пул-запиті GitHub

#include <iostream>

#include <stdlib.h>

#include <ctime>

using namespace std;

int main ( ) {

int array [100];

int new\_array[100];

int count, new\_count, x;

cout << "Enter the size of array : ";

cin >> count;

new\_count = count;

srand(time(NULL));

for (int i=0; i<count; i++) {

array[i]=rand()%101;

cout << array[i] << " ";

}

for (int d=count; d>count-5; d--) {

new\_count--;

}

for (int j=0; j<new\_count; j++) {

new\_array[j]=array[j];

}

for (int b=new\_count; b>=0; b--) {

x=new\_array[b];

new\_array[b+3]=x;

}

for (int k=2; k>=0; k--) {

new\_array[k]=new\_array[k+1]+2;

}

cout << "\n";

for (int i=0; i<new\_count+3; i++) {

cout << new\_array[i] << " ";

}

}

Завдання **VNS Lab 5 - Task 1-N** варіант 11

Посилання на файл програми у пул-запиті GitHub

#include <iostream>

using namespace std;

const int M = 4;

const int N = 4;

int x = 0;

bool Is\_Ascending (int array[M][N], int mm) {

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

if (array[mm][i]>array[mm][i+1]) {

Is\_Ascending(array, mm+1);

return false;

}

}

return true;

}

bool Is\_Descending (int array[M][N], int mm) {

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

if (array[mm][i]<array[mm][i+1]) {

Is\_Descending(array, mm+1);

return false;

}

}

return true;

}

void AscendingSort (int new\_array[M][N], int m) {

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

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

if (new\_array[m][w] > new\_array[m][w+1]) {

x = new\_array[m][w];

new\_array[m][w] = new\_array[m][w+1];

new\_array[m][w+1] = x;

}

}

}

}

int main ( ) {

int array [M][N];

int new\_array [M][N];

for (int i=0; i<M; i++) {

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

cin >> array[i][j];

}

}

for (int k=0; k<M; k++) {

for (int l=0; l<N; l++) {

new\_array[k][l]=array[k][l];

}

}

for (int i=0; i<M; i++) {

cout << i+1 << " row - Ascending : " << Is\_Ascending(array, i) << " Descending : " << Is\_Descending(array, i) << endl;

}

for (int i=0; i<M; i++) {

if (!Is\_Ascending(array, i) && !Is\_Descending(array, i)) {

AscendingSort(new\_array, i);

}

}

for (int i=0; i<M; i++) {

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

cout << new\_array[i][j] << " ";

}

cout << "\n";

}

}

Завдання **Algotester Lab 2** варіант 1

Посилання на файл програми у пул-запиті GitHub

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int n, x;

void BubbleSort (int n, unsigned int rr[]) {

for (int j=0; j<n; j++) {

for (int w=0; w<n-1; w++) {

if (rr[w] > rr[w+1]) {

x = rr[w];

rr[w] = rr[w+1];

rr[w+1] = x;

}

}

}

}

int main () {

cin >> n;

unsigned int r[n];

unsigned int rr[n];

for (int i=0; i<n; i++) {

cin >> r[i];

}

for (int k=0; k<n; k++) {

rr[k]=r[k];

}

if (n==1) {

cout << 0;

return 0;

}

BubbleSort(n, rr);

int min\_f = rr[n-2] - rr[0];

int min\_l = rr[n-1] - rr[1];

int position = -1;

if (min\_f<min\_l) {

cout << min\_f;

} else {

cout << min\_l;

}

}

Завдання **Algotester Lab 3** варіант 3

Посилання на файл програми у пул-запиті GitHub

#include <iostream>

#include <string>

using namespace std;

int main () {

long int k = 1;

string line;

cin >> line;

string compressed\_line = "";

for (int l = 0; l < size(line)- 1; l++) {

if (size(line)==1) {

cout << line;

return 0;

}

if (line[l] == line[l+1]) {

k++;

}

else {

if (k==1) {

compressed\_line += line[l];

}

else {

compressed\_line += line[l];

compressed\_line += to\_string(k);

k=1;

}

}

}

if (k==1) {

compressed\_line += line.back();

}

else {

compressed\_line += line.back();

compressed\_line += to\_string(k);

}

cout << compressed\_line << endl;

return 0;

}

Завдання **Class Practice Work**

Посилання на файл програми у пул-запиті GitHub

#include <iostream>

using namespace std;

bool IsPalindrome (string word, int left, int right) {

if (word[left]==word[right]) {

left++;

right--;

IsPalindrome(word, left, right);

return true;

}

else {

return false;

}

}

int IsPalindrome (int number, int digit, int reversed) {

while (number > 0) {

int digit = number%10;

reversed = reversed\*10 + digit;

number /= 10;

IsPalindrome(number, digit, reversed);

return 1;

}

return 0;

}

int main () {

int left, right, number, original\_num, digit, reversed;

string word;

cin >> word;

cin >> number;

left = 0;

right = size(word)-1;

cout << IsPalindrome(word, left, right) << endl;

string s\_num = to\_string(number);

right = size(s\_num)-1;

cout << IsPalindrome (s\_num, left, right) << endl;

reversed = 0;

original\_num = number;

cout << IsPalindrome(number, digit, reversed) << endl;

}

Завдання **Self Practice Work**

#include <iostream>

using namespace std;

int main() {

unsigned int a,b,c;

cin >> a;

cin >> b;

cin >> c;

if (a+b>c) {

cout << "YES";

}

else {

cout << "NO";

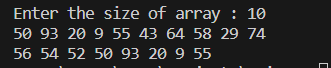
}

}

Посилання на файл програми у пул-запиті GitHub

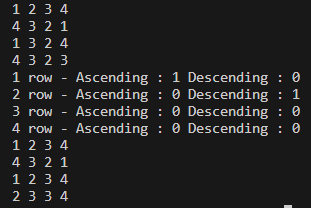
## **5. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання **VNS Lab 4 - Task 1-N** варіант 11



Планований час виконання - 2 години, фактичний - 2 год

Завдання **VNS Lab 5 - Task 1-N** варіант 11



Планований час виконання - 4 години, фактичний - 4 год

Завдання **Algotester Lab 2** варіант 1



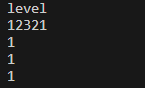
Планований час виконання - 4 години, фактичний - 4 год

Завдання **Algotester Lab 3** варіант 3



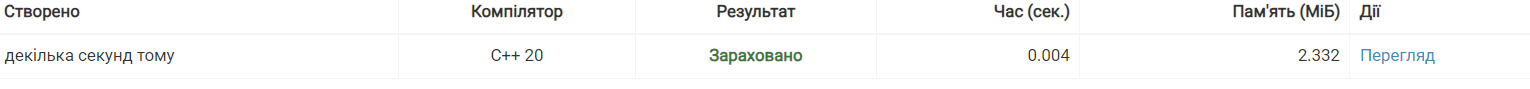
Планований час виконання - 3 години, фактичний - 4 год

Завдання **Class Practice Work**

****

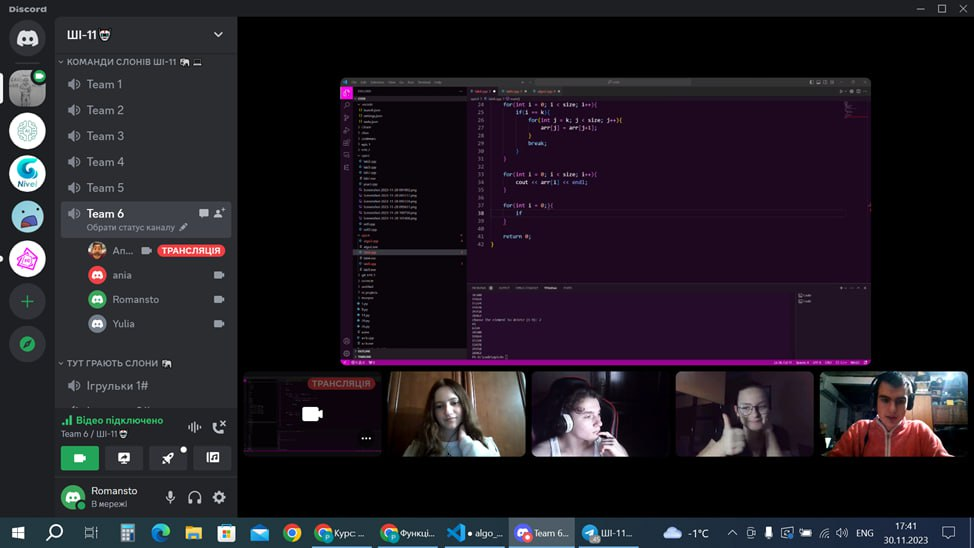
Планований час виконання - 2 години, фактичний - 1 год

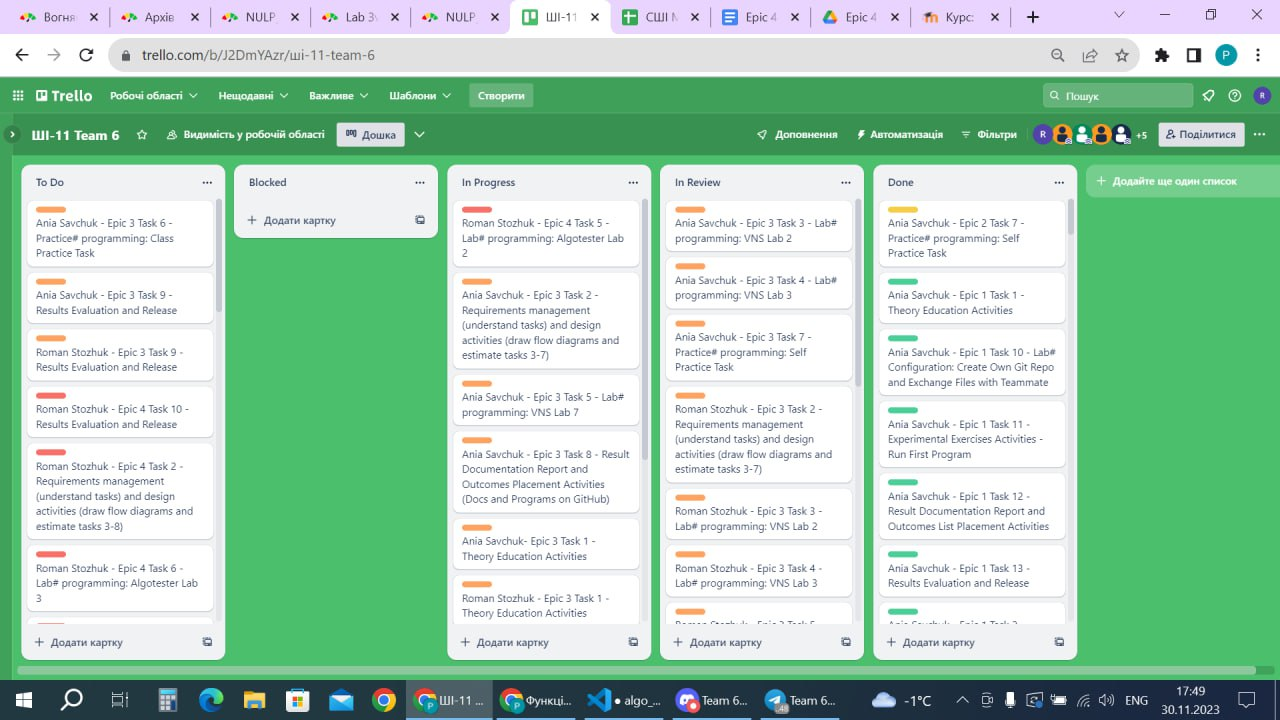
Завдання **Self Practice Work**

****

Планований час виконання - 30 хвилин, фактичний - 15 хвилин

## **6. Кооперація з командою:**





# **Висновки:**

Ознайомилась з масивами і принципами роботи з ними та їх обробки, структурами даних, вказівниками та посиланнями і алгоритмами обробки масивів та структур.