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:*** «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-12

Яцишин Ігор Васильович

# **Тема роботи:**

Ознайомлення то робота з одновимірними масивами в мові програмування C++. Принцип роботи з двовимірними масивами даних. Ознайомлення з типовими алгоритмами обробки масивів.

# **Мета роботи:**

Навчитися працювати з одновимірними та двовимірними масивами в мові C++. Вивчити основні алгоритми обробки масивів даних. Застосувати набуті знання на практиці.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Одновимірні масиви даних.
* Тема №2: Поняття двовимірний масив.
* Тема №3: Масиви як параметри функцій в мові C++.
* Тема №4: Стандартні операції обробки масивів.
* Тема №5: Базові алгоритми сортування масивів.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Одновимірні масиви даних.
  + Джерела Інформації
    - Стаття: C++ Arrays [C++ Arrays (With Examples)](https://www.programiz.com/cpp-programming/arrays)
  + Що опрацьовано:
    - Опрацьовано принцип роботи з одновимірними масивами, їх специфікацію та основні приклади використання
  + Статус: Ознайомлений
  + Початок опрацювання теми: 18.11.2023 11.20
  + Звершення опрацювання теми: 18.11.2023 12.00
* Тема №2: Поняття двовимірний масив.
  + Джерела Інформації:
    - Стаття: Two Dimensional Array in C++ [Two Dimensional Array in C++ | DigitalOcean](https://www.digitalocean.com/community/tutorials/two-dimensional-array-in-c-plus-plus)
    - Стаття: Two Dimensional Array [Two Dimensional Arrays in C++ | What are 2D Arrays? | Examples](https://www.toppr.com/guides/computer-science/programming-in-c-/structured-data-type/two-dimensional-array/)
  + Що опрацьовано:
    - Вивчено теоретичні основи роботи з 2D масивами в мові програмування С++
    - Опрацьовано приклади практичного застосування двовимірних масивів даних в програмному коді
  + Статус: Ознайомлений
  + Початок опрацювання теми: 18.11.2023 13.10
  + Звершення опрацювання теми: 18.11.2023 14.25
* Тема №3: Масиви як параметри функцій в мові C++.
  + Джерела Інформації:
    - Стаття: Масиви як параметри функцій в C++ [Масиви як параметри функцій](https://cherto4ka.xyz/2020/01/22/%D0%BC%D0%B0%D1%81%D0%B8%D0%B2%D0%B8-%D1%8F%D0%BA-%D0%BF%D0%B0%D1%80%D0%B0%D0%BC%D0%B5%D1%82%D1%80%D0%B8-%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D0%B9/)
  + Що опрацьовано:
    - Опрацьовано основні способи та правила передачі масивів як параметрів функцій
    - Розглянуто приклади подальшої роботи з масивами в тілі функцій та їх застосування
  + Статус: Ознайомлений
  + Початок опрацювання теми: 18.11.2023 14.30
  + Звершення опрацювання теми: 184.11.2023 15.50
* Тема №4: Стандартні операції обробки масивів.
  + Джерела Інформації:
    - Стаття: Стандартні операції обробки одновимірних масивів [Базові операції обробки одновимірних масивів](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/)
    - Відео: C++ Теорія ⦁ Урок 40 ⦁ Одновимірні масиви [C++ Теорія ⦁ Урок 40 ⦁ Одновимірні масиви](https://www.youtube.com/watch?v=ULdbOaMBPYc)
  + Що опрацьовано:
    - Опрацьовано та вивчено основні операції, що спрямовані на обробку масивів, розглянуто практичні приклади їх застосування
  + Статус: Ознайомлений
  + Початок опрацювання теми: 18.11.2023 16.00
  + Звершення опрацювання теми: 18.11.2023 16.50
* Тема №5: Базові алгоритми сортування масивів.
  + Джерела Інформації:
    - Відео: C++ Теорія ⦁ Урок 51 ⦁ Сортування вибіркою [C++ Теорія ⦁ Урок 51 ⦁ Сортування вибіркою](https://www.youtube.com/watch?v=uQxG9gBROog&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=51)
    - Відео: C++ Теорія ⦁ Урок 52 ⦁ Сортування бульбашкою [C++ Теорія ⦁ Урок 52 ⦁ Сортування бульбашкою](https://www.youtube.com/watch?v=maB87eyn7h8)
    - Відео: C++ Теорія ⦁ Урок 53 ⦁ Сортування вставками [C++ Теорія ⦁ Урок 53 ⦁ Сортування вставками](https://www.youtube.com/watch?v=YFLRN_Gmh4o)
    - Відео: C++ Теорія ⦁ Урок 55 ⦁ Швидке сортування [C++ Теорія ⦁ Урок 55 ⦁ Швидке сортування](https://www.youtube.com/watch?v=mSFZtI8ui4g)
  + Що опрацьовано:
    - Опрацьовано чотири найпоширеніших алгоритми сортування одновимірних масивів
    - Випробувано алгоритми сортування у програмному коді мовою програмування C++
  + Статус: Ознайомлений
  + Початок опрацювання теми: 19.11.2023 20.00
  + Звершення опрацювання теми: 19.11.2023 21.50

# **Виконання роботи:**

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

Завдання №1 VNS Lab 4

* Варіант завдання: 25
* Деталі завдання : реалізувати за допомогою статичного масиву однонаправлене кільце та виконати низку дій над його елементами
* Важливі деталі для врахування в імплементації програми:

Використання статичних масивів доволі великих розмірів для організації псевдозмінних меж масиву

Завдання №2 VNS Lab 5

* Варіант завдання: 25
* Деталі завдання: Відновити двовимірний масив з одновимірного, в який його записали по рядках
* Важливі деталі для врахування в імплементації програми

Використання власноруч написаної функції для відновлення масиву

Завдання №3 Algotester Lab 2

* Варіант завдання: 1
* Деталі завдання: <https://algotester.com/en/ContestProblem/DisplayWithEditor/134635>
* Важливі деталі для врахування в імплементації програми

Використання алгоритму сортування бульбашкою для сортування вхідного масиву даних

Завдання №4 Algotester Lab 3

* Варіант завдання: 3
* Деталі завдання: <https://algotester.com/en/ContestProblem/DisplayWithEditor/134639>
* Важливі деталі для врахування в імплементації програми

Використання стрічок та циклів для перевірки повторюваності елементів

Завдання №5 Class Practice Work

* Варіант завдання: немає
* Деталі завдання: Написати програму для перевірки стрічки та числа на паліндром
* Важливі деталі для врахування в імплементації програми

Використання рекурсивних функцій та перевантаження

Завдання №6 Self Practice Work

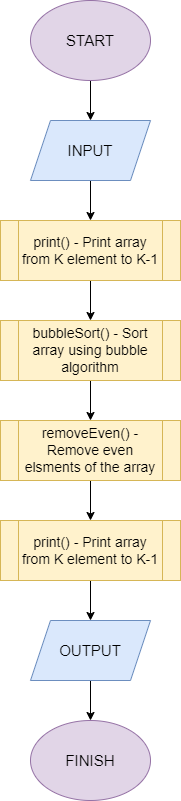
* Варіант завдання: немає
* Деталі завдання: [Коля, Вася і Теніс | Архів | Алготестер](https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/13)
* Важливі деталі для врахування в імплементації програми

Використання циклів for та масиву для обчислення рахунку гри

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

Програма №1 VNS Lab 4

* Блок-схема

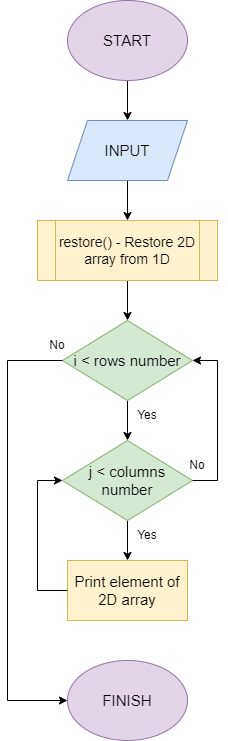


*Рисунок 1. Блок-схема до програми №1*

* Планований час на реалізацію: 80 хвилин

Програма №2 VNS Lab 5

* Блок-схема

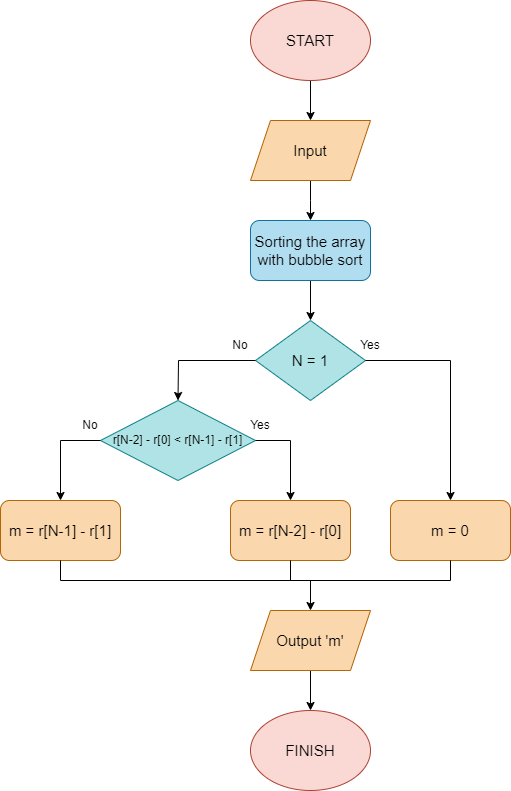


*Рисунок 2. Блок-схема до програми №2*

* Планований час на реалізацію: 90 хвилин

Програма №3 Algotester Lab 2

* Блок-схема

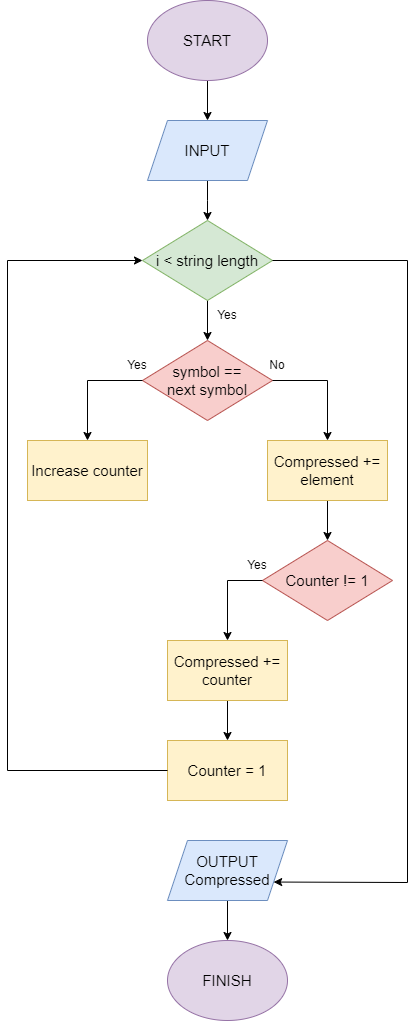


*Рисунок 3. Блок-схема до програми №3*

* Планований час на реалізацію: 80 хвилин

Програма №4 Algotester Lab 3

* Блок-схема

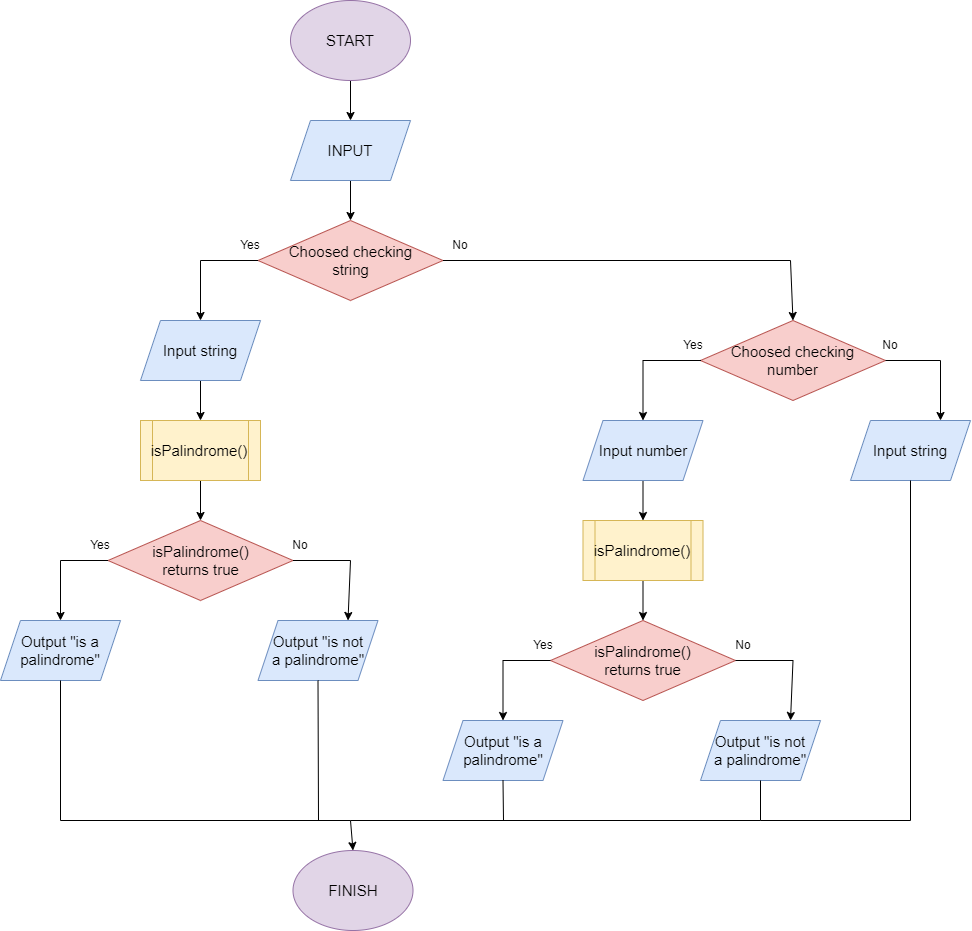


*Рисунок 4. Блок-схема до програми №4*

* Планований час на реалізацію: 60 хвилин

Програма №5 Class Practice Work

* Блок-схема



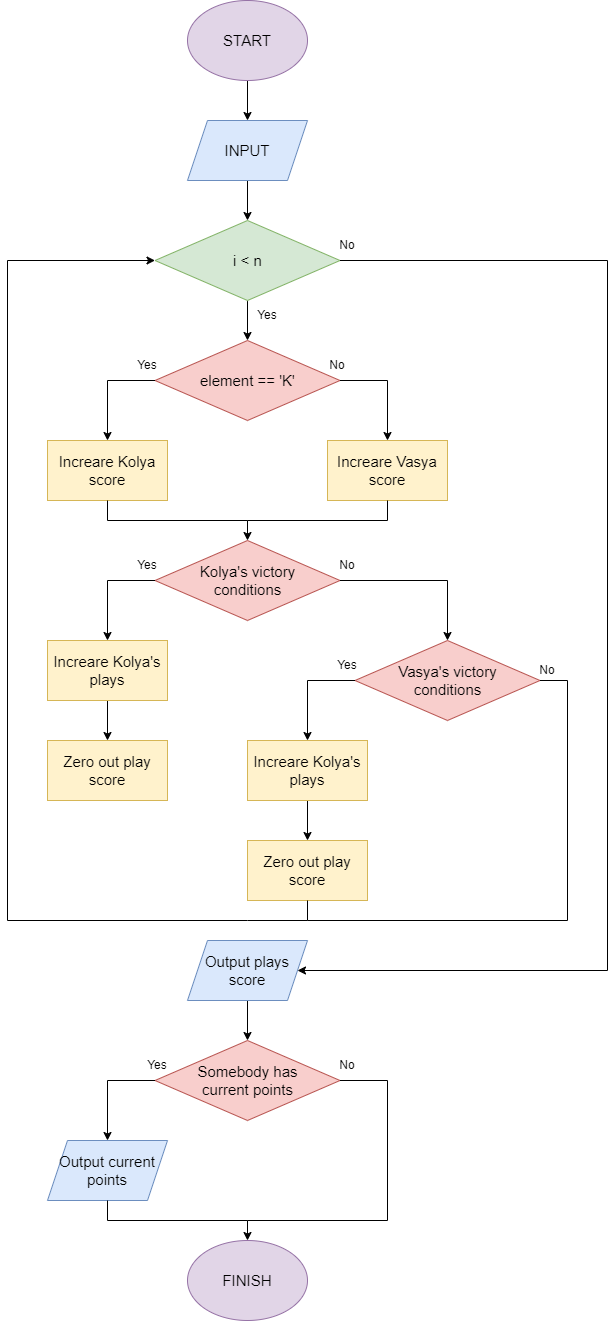
Output “wrong input”

*Рисунок 5. Блок-схема до програми №5*

* Планований час на реалізацію: 60 хвилин

Програма №6 Self Practice Work

* Блок-схема



*Рисунок 6. Блок-схема до програми №6*

* Планований час на реалізацію: 50 хвилин

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

Для виконання поставлених завдань додаткова конфігурація середовища не є необхідною

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

Завдання №1 VNS Lab 4

#include <iostream>

void print(int a[], int n, int k);

void bubbleSort(int a[], int size);

void removeEven(int a[], int &n);

int main() {

int n, k;

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

std::cin >> n;

std::cout << "Enter the value of 'K': ";

std::cin >> k;

int array[n];

std::cout << "Enter elements of the array: ";

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

std::cin >> array[i];

}

print(array, n, k);

bubbleSort(array, n);

removeEven(array, n);

print(array, n, k);

return 0;

}

void print(int a[], int n, int k) {

for(int i = k - 1; i < n; i++) {

std::cout << a[i] << " ";

}

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

std::cout << a[i] << " ";

}

std::cout << std::endl;

}

void bubbleSort(int a[], int n) {

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

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

if (a[j] < a[j+1]){

int temp = a[j];

a[j] = a[j+1];

a[j+1] = temp;

}

}

}

}

void removeEven(int a[], int &n) {

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

if(a[i] % 2 == 0) {

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

a[j] = a[j+1];

}

n--;

}

}

}

В програмі описано функції print(), bubbleSort() та removeEven(), які в якості одного з параметрів приймають масив та обробляють його згідно з варіантом завдання

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/500>

Завдання №2 VNS Lab 5

#include <iostream>

void restore(int \*array, int rows, int columns);

int main() {

int rows, columns;

std::cout << "Enter number of rows: " << std::endl;

std::cin >> rows;

std::cout << "Enter number of columns: " << std::endl;

std::cin >> columns;

int array[rows \* columns];

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

std::cout << "Enter element number " << i + 1 << ": ";

std::cin >> array[i];

}

restore(array, rows, columns);

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

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

std::cout << array[i \* columns + j] << " ";

}

std::cout << std::endl;

}

return 0;

}

void restore(int \*array, int rows, int columns) {

int row, column;

row = column = 0;

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

if(row == 0 || array[i] != array[i - column]) {

row++;

}

array[row \* columns + column];

if(column == columns - 1 || array[i+1] != array[i]) {

column++;

}

}

}

Програма містить в собі функцію restore(), яка відновлює двовимірний масив з одновимірного. Також в основному тілі програми початковий масив вводиться за допомогою циклу, а відновлений виводиться за допомогою двох циклів, один з яких є вкладеним в інший.

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/500>

Завдання №3 Algotester Lab 2

#include <iostream>

void bubbleSort(long arr[], long size);

int main() {

long N, m;

std::cin >> N;

long r[N];

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

std::cin >> r[i];

}

bubbleSort(r, N);

if (N == 1){

m = 0;

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

m = r[N - 2] - r[0];

}else {

m = r[N - 1] - r[1];

}

std::cout << m << std::endl;

return 0;

}

void bubbleSort(long arr[], long size) {

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

for(int j = 0; j < size - i; ++j){

if (arr[j] > arr[j+1]){

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

}

Програма отримує цілочисельне значення змінної N та кожнен елемент масиву r[] розміром N елементів. Шляхом “видалення” першого або останнього значеня масиву r розгалуження if мінімізує різницю найбільшого і найменшого елемента. Результат виводиться в консоль

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/500>

Завдання №4 Algotester Lab 3

#include <iostream>

#include <string>

int main() {

std::string s, sComp;

int counter;

std::cin >> s;

counter = 1;

for(int i = 0; i < s.length(); i++){

if(s[i] == s[i + 1]){

counter ++;

}else {

sComp += s[i];

if(counter != 1){

sComp += std::to\_string(counter);

}

counter = 1;

}

}

std::cout << sComp;

return 0;

}

Програма отримує стрічку з набором символів як вхідні дані. Після цього у тілі циклу for здійснюється перевірка на однаковість двох сусідніх символів рядку. У разі збігу збільшується лічильник. Після виконаних операцій виводиться компресований рядок

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/500>

Завдання №5 Class Practice Work

#include <iostream>

#include <string>

bool isPalindrome(std::string string);

bool isPalindrome(int number);

int main () {

std::string input1, operation;

int input2;

std::cout << "What do you want to check?" << std::endl;

std::cin >> operation;

if(operation == "string") {

std::cout << "Enter the string: ";

std::cin >> input1;

isPalindrome(input1);

if(isPalindrome(input1)) {

std::cout << input1 << " is a palindrome" << std::endl;

}else {

std::cout << input1 << " is not a palindrome" << std::endl;

}

}else if(operation == "number"){

std::cout << "Enter the number: ";

std::cin >> input2;

isPalindrome(input2);

if(isPalindrome(input2)) {

std::cout << input2 << " is a palindrome" << std::endl;

}else {

std::cout << input2 << " is not a palindrome" << std::endl;

}

}else {

std::cout << "Invalid input";

}

return 0;

}

bool isPalindrome(std::string string) {

int first = 0;

int last = string.length() - 1;

if(first < last){

if (string[first] != string[last]){

return false;

}else {

first++;

last--;

return isPalindrome(string.substr(first, last));

}

}

return true;

}

bool isPalindrome(int number) {

std::string str = std::to\_string(number);

return isPalindrome(str);

}

У програмі використовуються функція isPalindrome() для рядків та її перевантажена версія для цілих чисел. У тілі функції здійснюється рекурсивне порівняння першого та останнього символі стрічки чи числа. Результат виконання програми виводиться в консоль

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/500>

Завдання №6 Self Practice Work

#include <iostream>

int main() {

int n, K, V, s[] = {0, 0};

std::cin >> n;

K = V = 0;

char c[n];

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

std::cin >> c[i];

}

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

if(c[i] == 'K') {

s[0]++;

}else {

s[1]++;

}

if (s[0] >= 11 && s[0] - s[1] >= 2) {

K++;

s[0] = s[1] = 0;

}else if (s[1] >= 11 && s[1] - s[0] >= 2) {

V++;

s[0] = s[1] = 0;

}

}

std::cout << K << ":" << V << std::endl;

if (s[0] > 0 || s[1] > 0) {

std::cout << s[0] << ":" << s[1];

}

return 0;

}

У програмі використовуються цикли for для введення елементів масиву та для обчислення рахунку гри. Також використовується масив s[] для зберігання поточного рахунку гри. Результати гри по партіях та в поточному таймі виводяться на екран

Посилання на файл програми у пул-запиті GitHub

## <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/500>

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

Завдання №1 VNS Lab 4

INPUT:

Enter the array size: 7

Enter the value of 'K' : 3

Enter elements of the array: 1 2 3 4 5 6 7

OUTPUT:

3 4 5 6 7 1 2

3 1 7 5

Час затрачений на виконання завдання: 120 хвилин

Завдання №2 VNS Lab 5

INPUT:

Enter number of rows:

3

Enter number of columns:

3

Enter element number 1: 1

Enter element number 2: 2

Enter element number 3: 3

Enter element number 4: 4

Enter element number 5: 5

Enter element number 6: 6

Enter element number 7: 7

Enter element number 8: 8

Enter element number 9: 9

OUTPUT:

1 2 3

4 5 6

7 8 9

Час затрачений на виконання завдання: 60 хвилин

Завдання №3 Algotester Lab 2

INPUT:

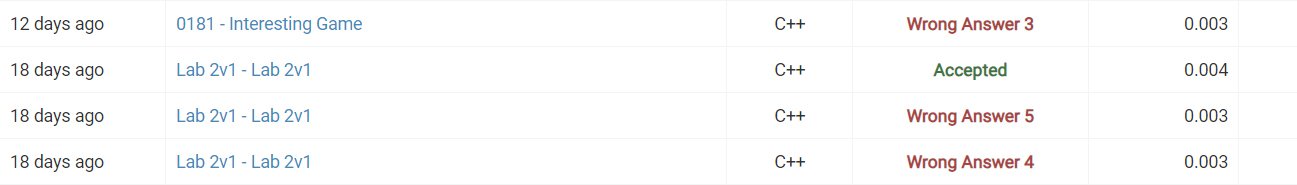
5

1 2 2 4 4

OUTPUT:

2

Час затрачений на виконання завдання: 80 хвилин



*Рисунок 7. Результат зарахування програми №3 на Алготестер*

Завдання №4 Algotester Lab 3

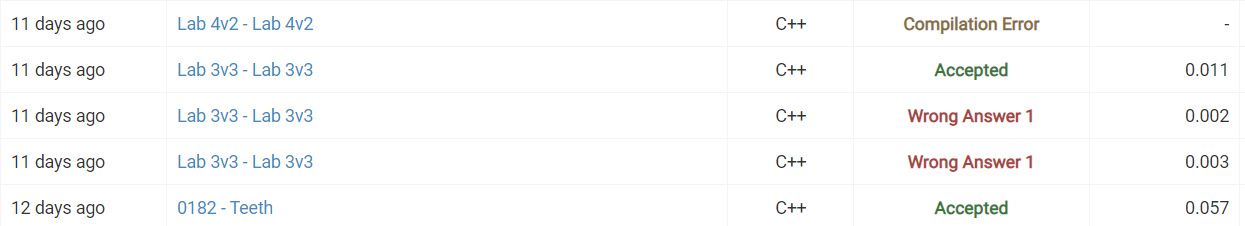
INPUT:

AAAAAAaaaaBBbCCCCCC

OUTPUT:

A6a4B2bC6

Час затрачений на виконання завдання: 45 хвилин



*Рисунок 8. Результат зарахування програми №4 на Алготестер*

Завдання №5 Class Practice Work

INPUT(EXAMPLE):

What do you want to check?

number

Enter the number: 1234

OUTPUT(EXAMPLE):

1234 is not a palindrome

Час затрачений на виконання завдання: 85 хвилин

Завдання №6 Self Practice Work

INPUT:

30

VVKVKKVVVVKVKKKKKVVKKVKKKKVVVV

OUTPUT

1:0

2:4

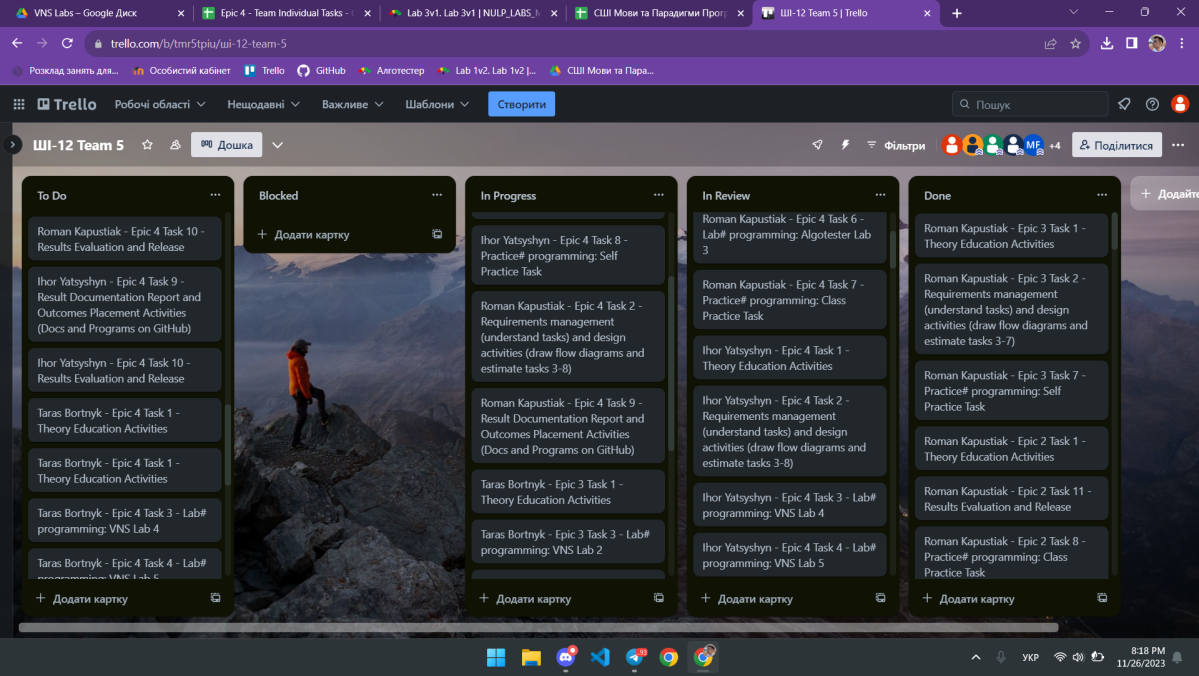
Час затрачений на виконання завдання: 90 хвилин

## **6. Кооперація з командою:**

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

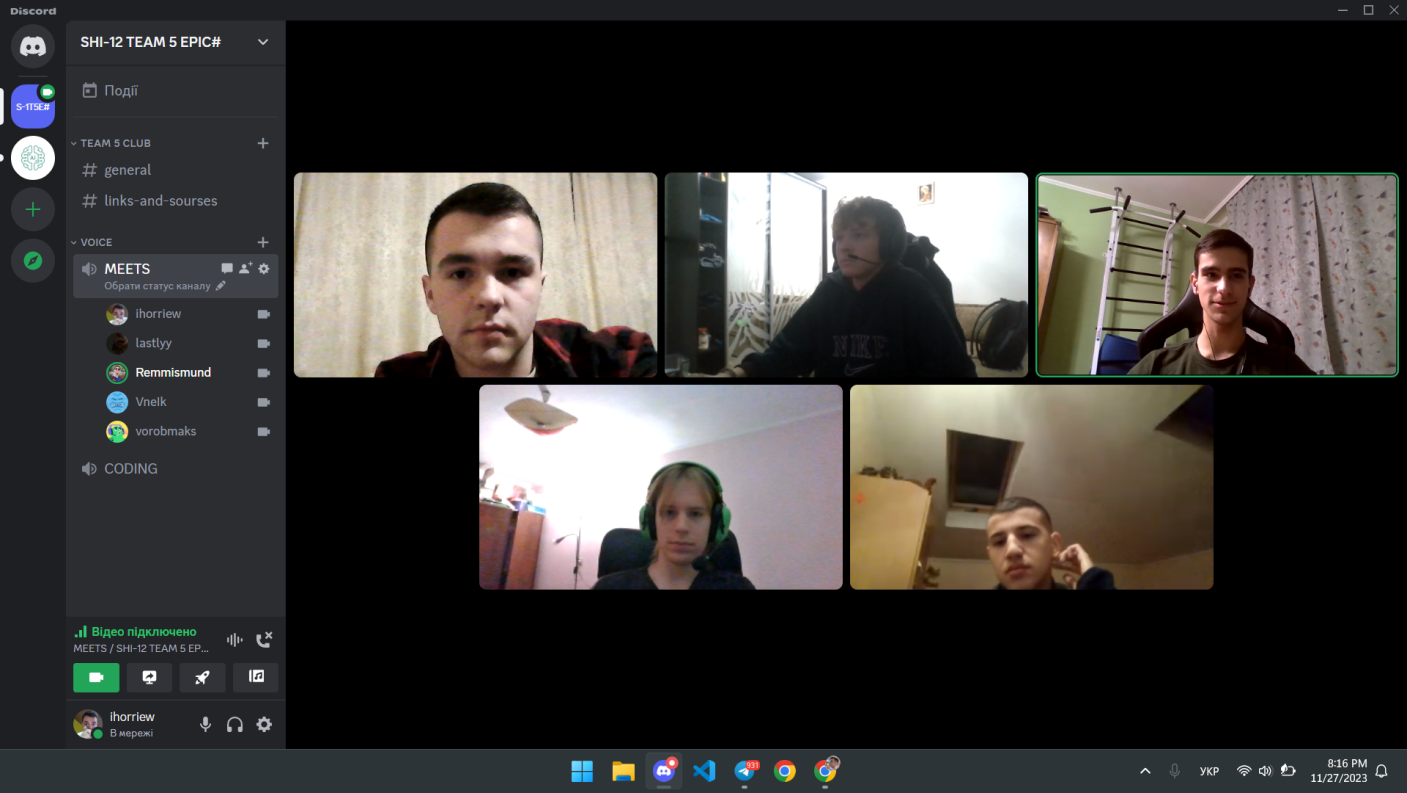


*Рисунок 9. Скріншот першої зустрічі*

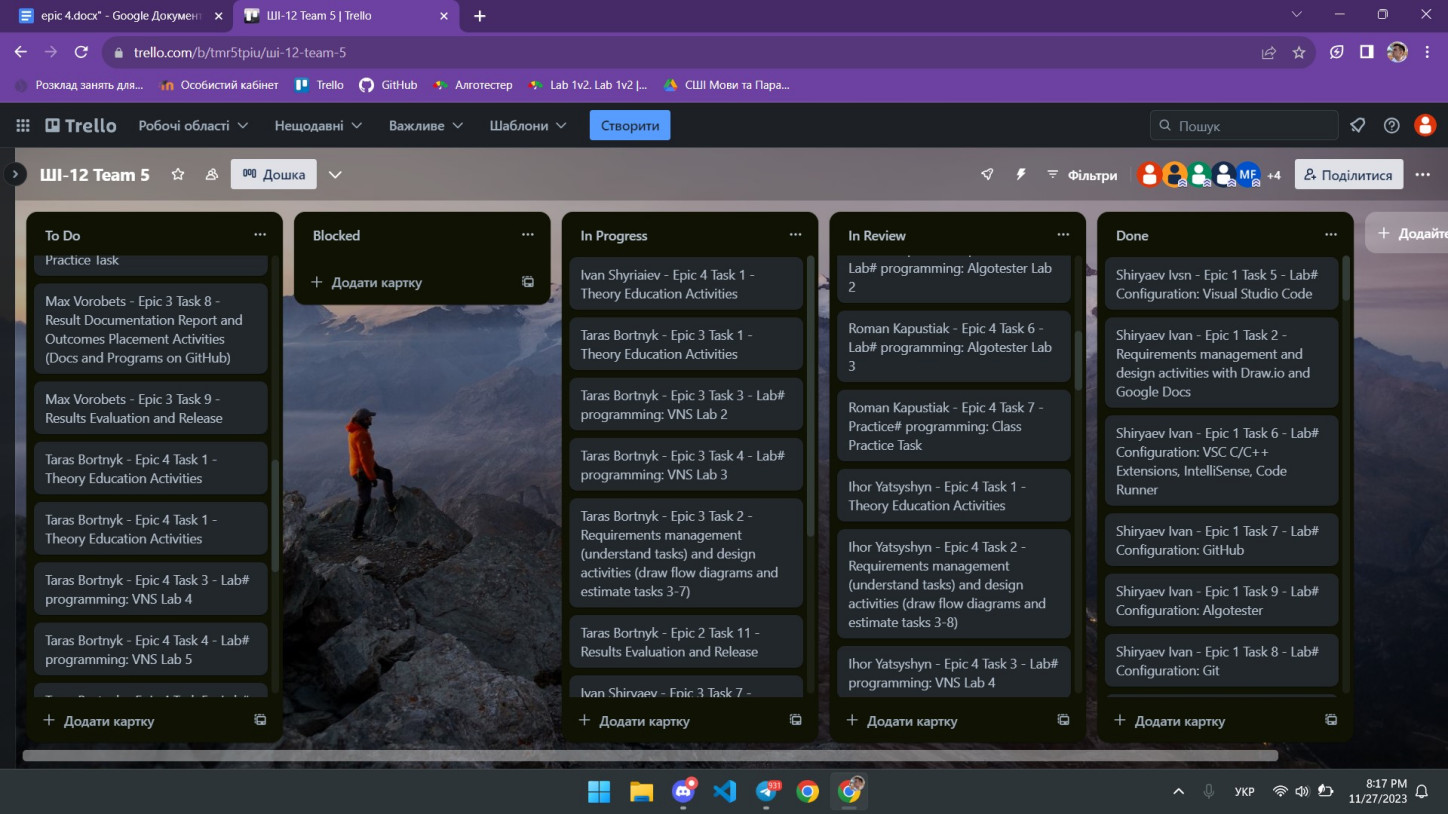
**

*Рисунок 10. Скріншот Trello на час першої зустрічі*

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

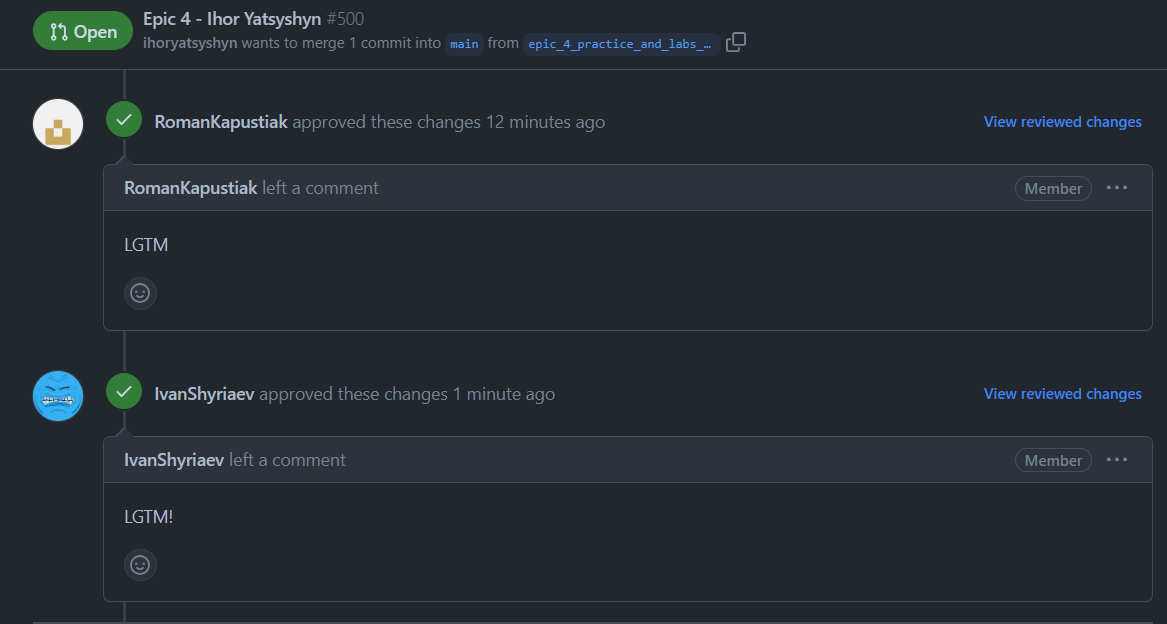


*Рисунок 11. Скріншот другої зустрічі*

**

*Рисунок 12. Скріншот Trello на час другої зустрічі*

* Скрін з 2-му коментарями від учасників команди на пул-реквесті з Рев’ю Роботи



*Рисунок 13. Скріншот коментарів від учасників команди до пул-реквесту*

# **Висновки:**

Під час опрацювання теоретичного матеріалу та роботи над завданнями розділу я навчився працювати з масивами в мові C++. Під час опрацювання матеріалу суттєвих труднощів не виникло. Весь пройдений матеріал закріплено практично завдяки виконанні лабораторних та практичних робіт а також самопрактиці. Також опрацьовано основні алгоритми сортування одновимірних масивів, що знадобилися і надалі знадобляться під час виконання поставлених завдань.