Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated  
  
  
  
**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

**Виконала:**

Студентка групи ШІ-13

Новосад Олена Петрівна

# **Тема роботи:**

Ознайомлення з концептом простих структур даних. Детальне ознайомлення із одновимірними та двомірними масивами, а також поняттям динамічного масиву. Опрацювання інформації щодо базових алгоритмів обробки масивів.

# **Мета роботи:**

Дослідити, що таке прості структури у мові С++, опрацювати теорію щодо одновимірних та двовимірних масивів, а також про поняття динамічного масиву та зручність його застосування. Після опрацювання попередньо згаданого матеріалу ознайомитися із алгоритмами обробки масивів. На практиці застосувати здобуті знання для кращого засвоєння інформації.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Прості структури даних.
* Тема №2: Одновимірні масиви. Динамічні масиви.
* Тема №3: Двовимірні масиви.
* Тема №4: Алгоритми обробки масивів.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Прості структури даних.
  + Джерела Інформації
    - <https://acode.com.ua/urok-64-struktury/>
  + Що опрацьовано:
    - Концепт простих структур даних та їх застосування.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 14.11.2023
  + Звершення опрацювання теми: 20.11.2023
* Тема №2: Одновимірні масиви. Динамічні масиви. Алгоритми обробки масивів
  + Джерела Інформації:
    - З. Я. Шпак «Програмування мовою С»
    - <https://acode.com.ua/urok-77-masyvy/>
    - <https://acode.com.ua/urok-78-fiksovani-masyvy/>
    - <https://acode.com.ua/urok-80-sortuvannya-masyviv-metodom-vyboru/>
    - <https://acode.com.ua/urok-89-dynamichne-vydilennya-pam-yati/>
    - <https://acode.com.ua/urok-99-vvedennya-v-std-vector/>
  + Що опрацьовано:
    - Що таке одновимірний масив, як його оголошувати та працювати з ним. Розуміння як працювати з індексами. Поняття динамічного масиву, та як його створювати. Алгоритми обробки масивів.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 07.11.2023
  + Звершення опрацювання теми: 25.11.2023
* Тема №3: Двовимірні масиви.
  + Джерела Інформації:
    - [https://acode.com.ua/urok-81-bagatovymirni-masyvy/](https://acode.com.ua/urok-81-bagatovymirni-masyvy/%20)
    - ВНС конспект лекцій
  + Що опрацьовано:
    - Що таке двовимірний масив та як працювати з його індексами.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 17.11.2023
  + Звершення опрацювання теми: 03.12.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 4

* Варіант 16
* Реалізувати з використанням масиву чергу (перший прийшов, перший пішов),

для чого організувати додавання, знищення елементів у масиві і друк масиву

після кожної операції.

* Важливі деталі для врахування в імплементації програми:

При виконанні роботи використовуються статичні масиви.

При зменшенні або збільшенні довжини масиву необхідно змінювати його

реальну довжину.

Завдання №2 VNS Lab 5

* Варіант 16
* Задано двовимірний масив N x N. Дозволяється довільно переставляти

елементи усередині будь-якого стовпця. Перевірити, чи можна виконавши

скінчену кількість перестановок у стовпцях, розташувати на бічній діагоналі

елементи так, щоб вони зростали.

Завдання №3 Algotester Lab 2

* Варіант 1
* У вас є дорога, яка виглядає як N чисел. Пiсля того як ви по нiй пройдете - вашу втому можна визначити як рiзницю максимального та мiнiмального елементу. Ви хочете мiнiмiзувати втому, але все що ви можете зробити - викинути одне число з дороги,

тобто забрати його з масиву. В результатi цiєї дiї, яку мiнiмальну втому ви можете отримати в кiнцi дороги?

Важливі деталі для врахування в імплементації програми:

Вхідні дані:

У першому рядку цiле число N - кiлькiсть чисел

У другому рядку масив r, який складається з N цiлих чисел

Вихідні дані:

Єдине цiле число m - мiнiмальна втома, яку можна отримати

Завдання №4 Algotester Lab 3

* Варіант 3
* Вам дана стрiчка s. Ваше завдання зробити компресiю стрiчки, тобто якщо якась буква йде бiльше одного разу пiдряд у стрiчцi замiнити її на букву + кiлькiсть входжень пiдряд.
* Важливі деталі для врахування в імплементації програми:

Великi та маленькi букви вважаються рiзними.

Завдання №5 Class Practice Work «Перевірка чи слово або число є паліндромом»

* Задача

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

Паліндром — це послідовність символів, яка однаково читається вперед і назад (наприклад, «radar», «level», «12321»).

* Вимоги:

1. Визначення функції:

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

2. Приклад визначення функції:

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

3. Перевантаження функцій:

Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.

*bool isPalindrome*(ціле число);

4. Рекурсія:

Рекурсивна функція для рядків перевірить символи в поточній початковій і кінцевій позиціях. Якщо вони збігаються, він буде рекурсивно перевіряти наступні позиції, поки початок не перевищить кінець, після чого рядок буде визначено як паліндром.

* Кроки реалізації:
* 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.

Завдання №6 Self Practice Work

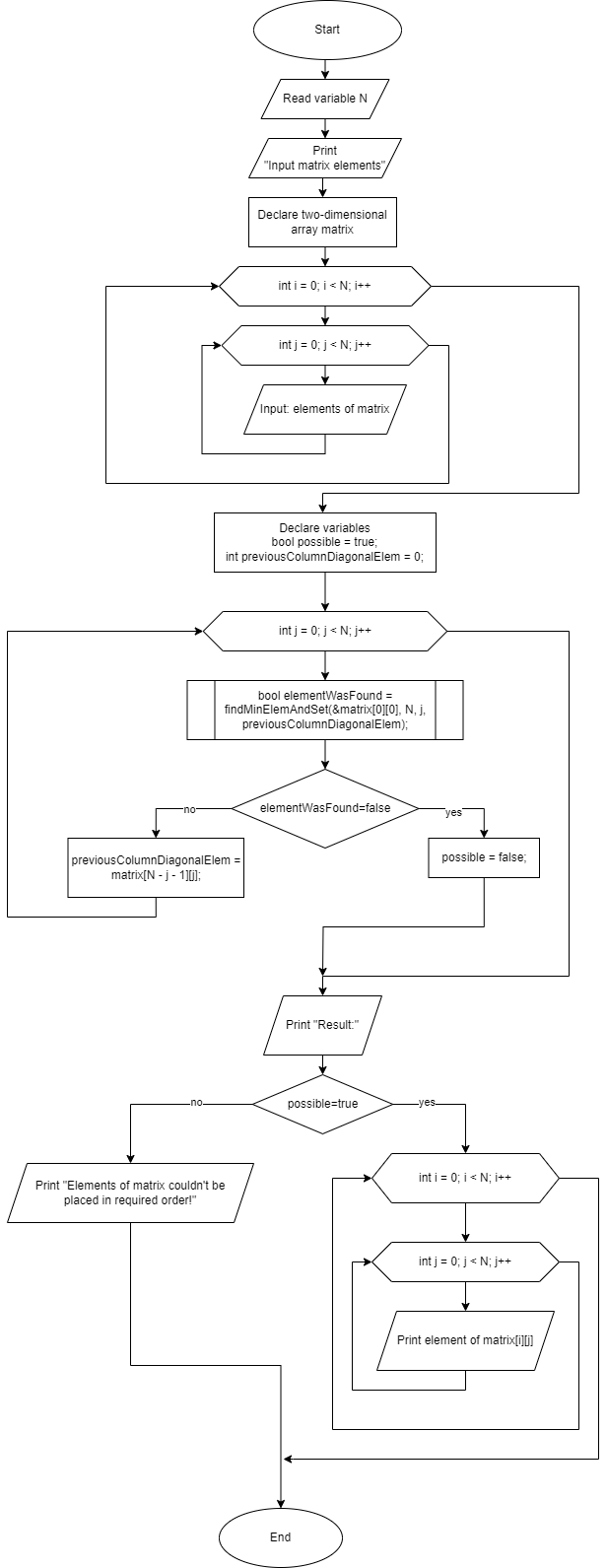
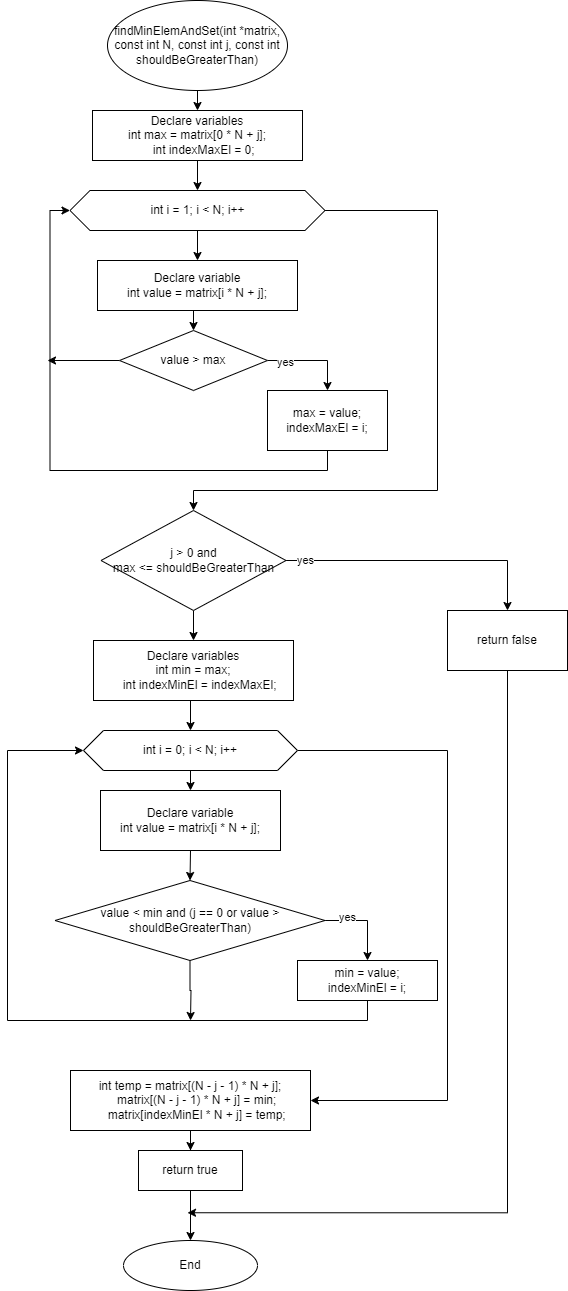
## 

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

Програма №1 VNS Lab 4

* Планований час на реалізацію: 1 день

Програма №2 VNS Lab 5

*  
* Планований час на реалізацію: 2 дні

Програма №3 Algotester Lab 2

* Планований час на реалізацію: 1 день

Програма №4 Algotester Lab 3

* Планований час на реалізацію: 1 день

Програма №5 Class Practice Work «Перевірка чи слово або число є паліндромом»

* Планований час на реалізацію: 1 день

Програма №6 Self Practice Work

* Планований час на реалізацію: 2 дні

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

Завдання №1 VNS Lab 4

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_olena_novosad/ai_13/olena_novosad/Epic%204/vns_lab_4_task_olena_novosad.cpp>

#include <iostream>

using namespace std;

struct FIFO {

    int arr[100000] = {0};

    int size = 0;

    int pop() {

        int elem = arr[0];

        int i = 1;

        while (i < size) {

            arr[i - 1] = arr[i];

            i++;

        }

        size--;

        return elem;

    }

    bool push(int newElem) {

        if (size < 100000) {

            arr[size] = newElem;

            size++;

            return true;

        }

        else {

            return false;

        }

    }

    void print() {

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

            cout << arr[i] << " ";

        }

        cout << endl;

    }

};

int main() {

    FIFO queue;

    queue.push(1);

    queue.print();

    queue.push(2);

    queue.print();

    queue.push(3);

    queue.print();

    queue.pop();

    queue.print();

    queue.pop();

    queue.print();

    queue.pop();

    queue.print();

    return 0;

}

Рисунок : Код до програми №1

Завдання №2 VNS Lab 5

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_olena_novosad/ai_13/olena_novosad/Epic%204/vns_lab_5_task_1_olena_novosad.cpp>

#include <iostream>

using namespace std;

bool findMinElemAndSet(int \*matrix, const int N, const int j, const int shouldBeGreaterThan)

{

    int max = matrix[0 \* N + j];

    int indexMaxEl = 0;

    for (int i = 1; i < N; i++)

    {

        int value = matrix[i \* N + j];

        if (value > max)

        {

            max = value;

            indexMaxEl = i;

        }

    }

    if (j > 0 && max <= shouldBeGreaterThan)

    {

        return false;

    }

    int min = max;

    int indexMinEl = indexMaxEl;

    for (int i = 0; i < N; i++)

    {

        int value = matrix[i \* N + j];

        if (value < min && (j == 0 || value > shouldBeGreaterThan))

        {

            min = value;

            indexMinEl = i;

        }

    }

    int temp = matrix[(N - j - 1) \* N + j];

    matrix[(N - j - 1) \* N + j] = min;

    matrix[indexMinEl \* N + j] = temp;

    return true;

}

int main()

{

    int N;

    cout << "Input N:\n";

    cin >> N;

    cout << endl;

    cout << "Input matrix " << N << "x " << N << ": \n";

    int matrix[N][N];

    for (int i = 0; i < N; i++)

    {

        for (int j = 0; j < N; j++)

        {

            cin >> matrix[i][j];

        }

    }

    bool possible = true;

    int previousColumnDiagonalElem = 0;

    for (int j = 0; j < N; j++)

    {

        bool elementWasFound = findMinElemAndSet(&matrix[0][0], N, j, previousColumnDiagonalElem);

        if (!elementWasFound)

        {

            possible = false;

            break;

        }

        else

        {

            previousColumnDiagonalElem = matrix[N - j - 1][j];

        }

    }

    cout << endl;

    cout << "Result:\n";

    if (possible)

    {

        for (int i = 0; i < N; i++)

        {

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

            {

                printf("%3d ", matrix[i][j]);

            }

            cout << "\n";

        }

    }

    else

    {

        cout << "Elements of matrix couldn't be placed in required order!";

    }

    return 0;

}

Рисунок : Код до програми №2

Завдання №3 Algotester Lab 2

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_olena_novosad/ai_13/olena_novosad/Epic%204/algotester_lab_2_task_1_olena_novosad.cpp>

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int m = 0;

int main()

{

    int N;

    cin >> N;

    if (N <= 0)

    {

        cout << m;

        return 0;

    }

    vector<int> vect;

    int n = 0;

    while (n < N)

    {

        int elem;

        cin >> elem;

        vect.push\_back(elem);

        n++;

    }

    if (N == 1)

    {

        cout << m;

        return 0;

    }

    sort(vect.begin(), vect.end());

    int min = vect.at(0);

    int pmin = vect.at(1);

    int max = vect.at(N - 1);

    int pmax = vect.at(N - 2);

    int deltamax = max - pmax;

    int deltamin = pmin - min;

    if (deltamax > deltamin)

    {

        m = pmax - min;

    }

    else

    {

        m = max - pmin;

    }

    cout << m << endl;

}

Рисунок : Код до програми №3

Завдання №4 Algotester Lab 3

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_olena_novosad/ai_13/olena_novosad/Epic%204/algotester_lab_3_task_1_olena_novosad.cpp>

#include <iostream>

#include <string>

using namespace std;

string compression(string S) {

    string result;

    bool compressed = true;

    int j = 1;

    char currentSymb;

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

        if (compressed) {

            result += S[i];

            currentSymb = S[i];

            compressed = false;

        }

        else {

            if (currentSymb == S[i]) {

                j++;

            }

            else {

                compressed = true;

                if (j != 1) {

                    result += to\_string(j);

                }

                j = 1;

                i--;

            }

        }

    }

    if (j != 1) {

        result += to\_string(j);

    }

    return result;

}

int main() {

    string S;

    cin >> S;

    cout << compression(S);

    return 0;

}

Рисунок :Код до програми №4

Завдання №5 Class Practice Work «Перевірка чи слово або число є паліндромом»

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_olena_novosad/ai_13/olena_novosad/Epic%204/practice_work_team_tasks_olena_novosad.cpp>

#include <iostream>

#include <string>

using namespace std;

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

{

    int length = str.length();

    if (str[start] == str[end])

    {

        int new\_start = start + 1;

        int new\_end = end - 1;

        if (new\_start == new\_end || new\_end < new\_start)

        {

            return true;

        }

        else

        {

            return isPalindrome(str, new\_start, new\_end);

        }

    }

    else

    {

        return false;

    }

}

bool isPalindrome(int number)

{

    string str = to\_string(number);

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

}

int main()

{

    string palindrom;

    cin >> palindrom;

    cout << (isPalindrome(palindrom, 0, palindrom.length() - 1) ? "yes" : "no") << endl;

    int palindrom\_n;

    cin >> palindrom\_n;

    cout << (isPalindrome(palindrom\_n) ? "yes" : "no");

    return 0;

}

Підпис та № до блоку з скріншотами до конфігурації

Рисунок :Код до програми №5

Завдання №6 Self Practice Work

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_olena_novosad/ai_13/olena_novosad/Epic%204/practice_work_self_algotester_tasks_olena_novosad.cpp>

#include <iostream>

#include <vector>

using namespace std;

vector<long int> v;

int findIndexOfNextOddElem(int fromIndex) {

    if (fromIndex < v.size()) {

        for (int j = fromIndex; j < v.size(); j++) {

            if (v[j] % 2 == 1) {

                return j;

            }

        }

    }

    return -1;

}

int findIndexOfNextEvenElem(int fromIndex) {

    if (fromIndex < v.size()) {

        for (int j = fromIndex; j < v.size(); j++) {

            if (v[j] % 2 == 0) {

                return j;

            }

        }

    }

    return -1;

}

void swapElems(int index1, int index2) {

    if (index1 < 0 || index2 < 0) {

        return;

    }

    long int temp = v[index1];

    v[index1] = v[index2];

    v[index2] = temp;

}

int main() {

    int n;

    cin >> n;

    for (int i = 0; i < n; i++) {

        long int elem;

        cin >> elem;

        v.push\_back(elem);

    }

    int oddElemsCount = 0;

    int evenElemsCount = 0;

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

        if (v.at(i) % 2 == 0) {

            evenElemsCount++;

        }

        else {

            oddElemsCount++;

        }

    }

    if (abs(oddElemsCount - evenElemsCount) > 1) {

        cout << -1;

        return 0;

    }

    bool needEven = evenElemsCount > oddElemsCount;

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

        bool currentIsEven = (v.at(i) % 2) == 0;

        if (needEven) {

            if (!currentIsEven) {

                int index2 = findIndexOfNextEvenElem(i + 1);

                swapElems(i, index2);

            }

        }

        else {

            if (currentIsEven) {

                int index2 = findIndexOfNextOddElem(i + 1);

                swapElems(i, index2);

            }

        }

        needEven = !needEven;

    }

    for (int i = 0; i < n; i++) {

        cout << v[i] << " ";

    }

    return 0;

}

Рисунок :Код до програми №6

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

Завдання №1 VNS Lab 4

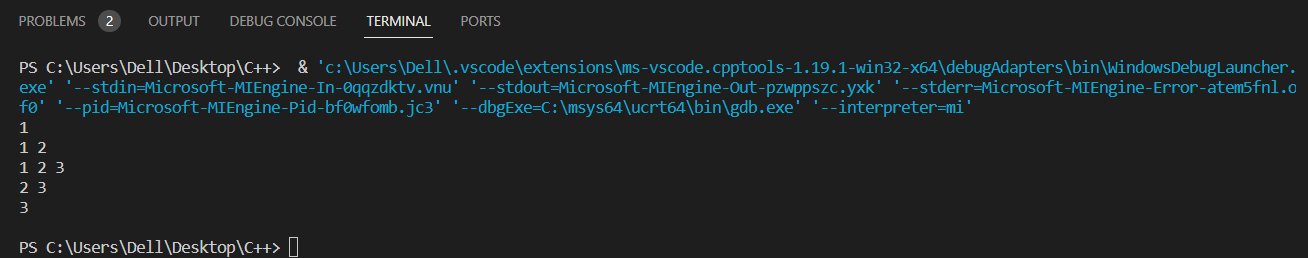


Рисунок : Тестування програми №1

Завдання №2 VNS Lab 5

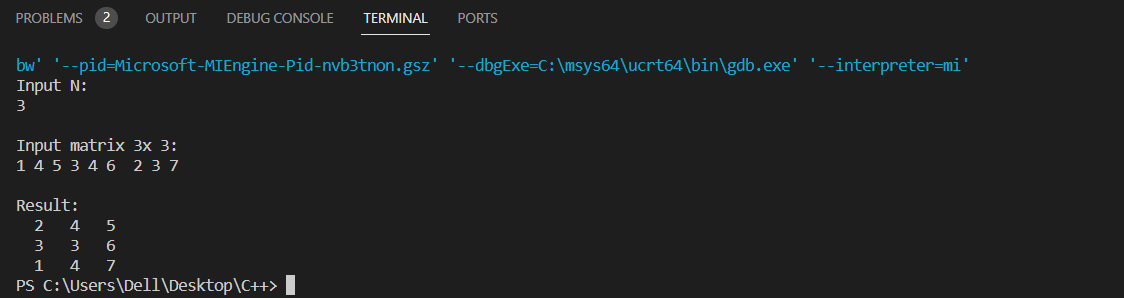


Рисунок : Тестування програми №2

Завдання №3 Algotester Lab 2

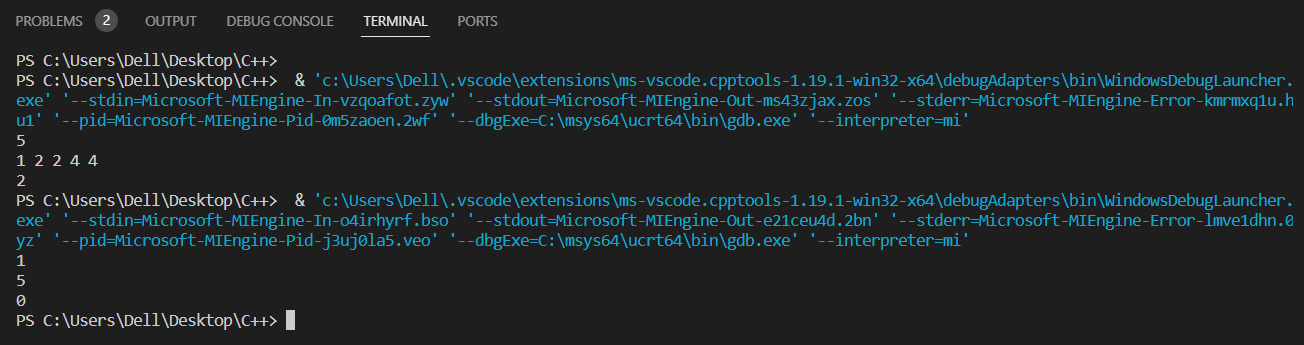


Рисунок :Тестування програми №3

Завдання №4 Algotester Lab 3

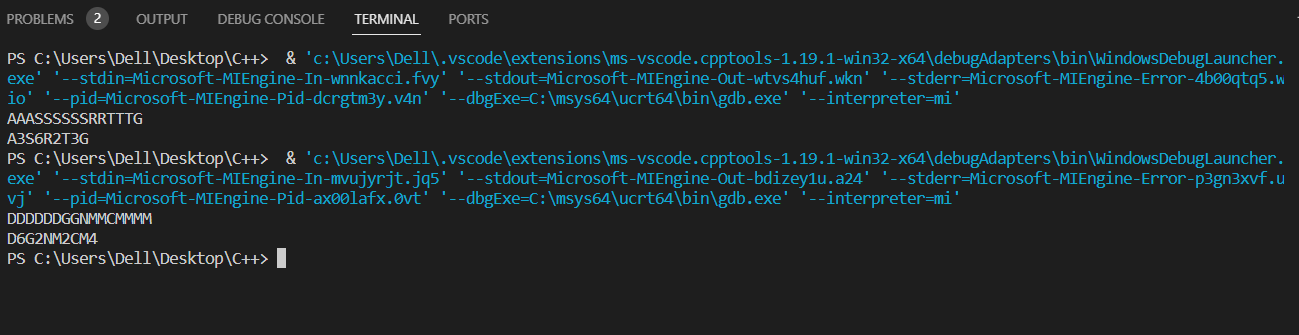


Рисунок :Тестування програми №4

Завдання №5 Class Practice Work «Перевірка чи слово або число є паліндромом»

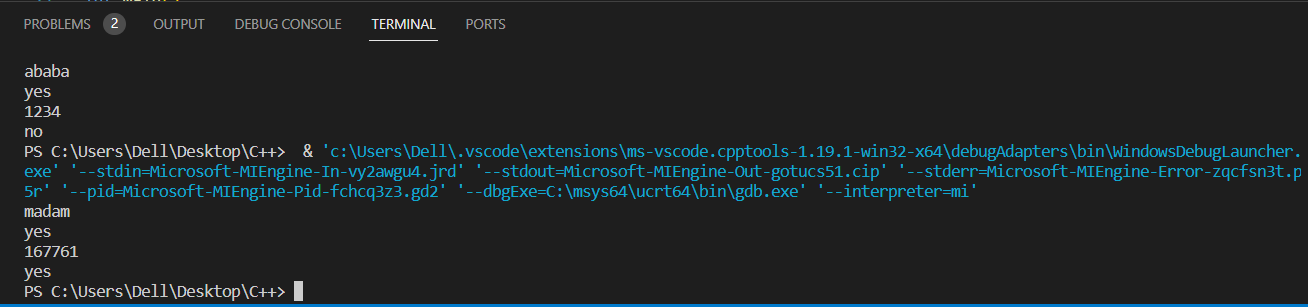


Рисунок :Тестування програми №5

Завдання №6 Self Practice Work

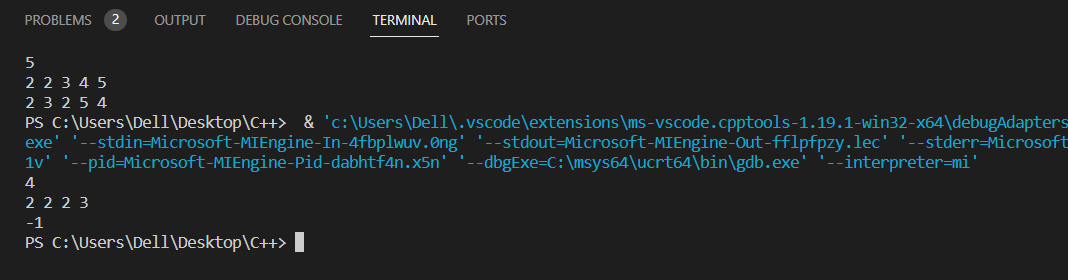
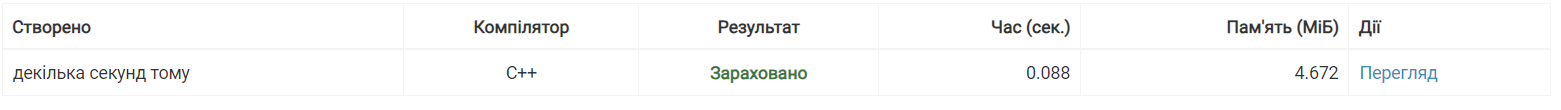


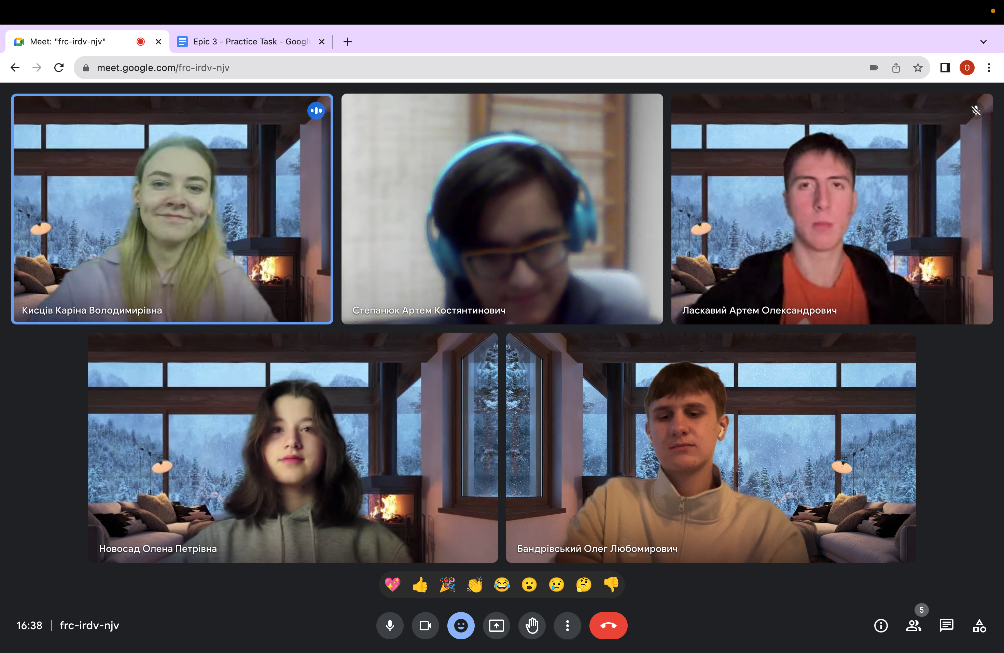
Рисунок : Тестування програми №6



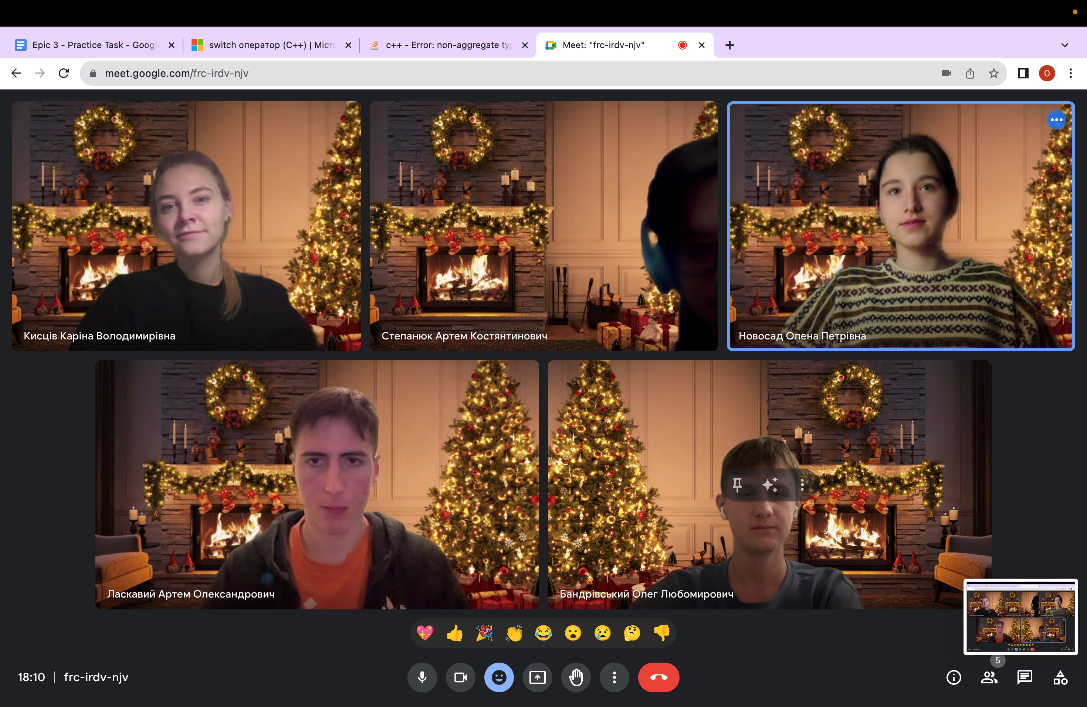
Час затрачений на виконання завдання: 2 дні.

## **6. Кооперація з командою:**

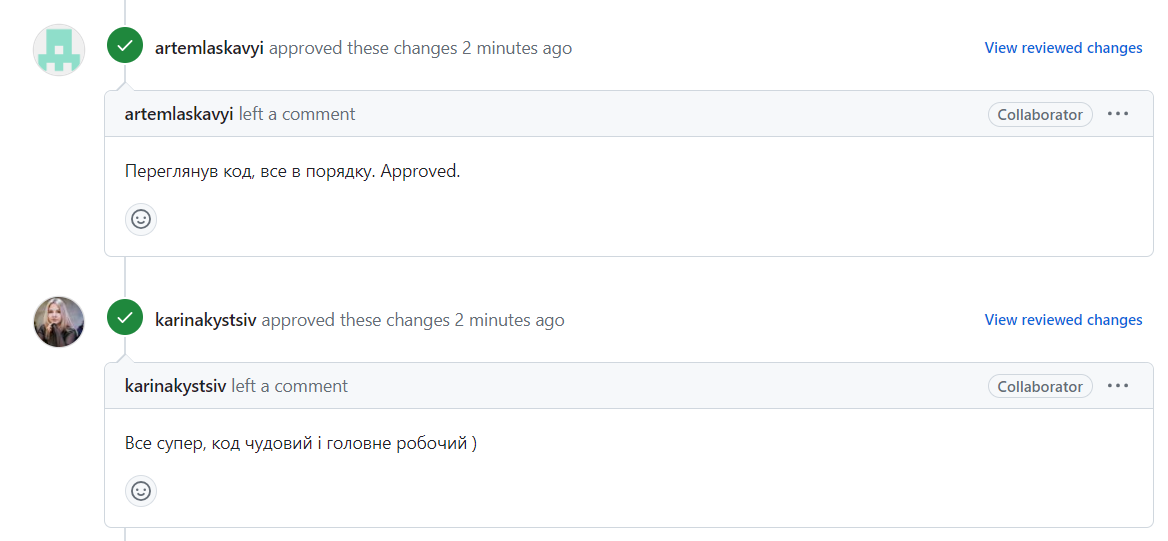
* Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло



Скрін з 2-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:**

Під час виконання практичних завдань та роботи з теорією я детальніше ознайомилася із одновимірними та двовимірними масивами даних. Також дізналася про особливості застосування динамічного масиву та його практичності у виконанні певних завдань. Розглянула інформацію про алгоритми сортування масивів. Ознайомилася із поняттям простих структур даних та застосувала у виконанні задач.