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.»

***Виконав:***

студент групи ШІ-11

Іванов Олексій Олександрович

# **Тема роботи:**

Опрацювання даних в одновимірних і двовимірних масивах на мові програмування С++

# **Мета роботи:**

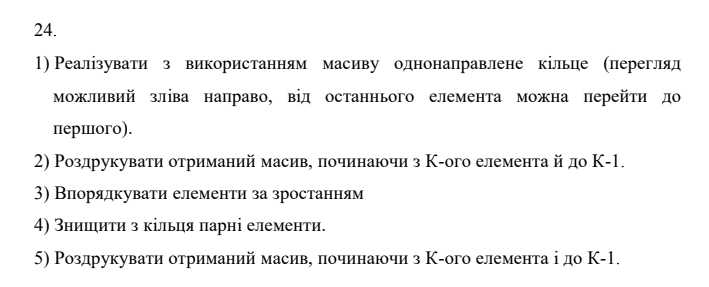
* Закріпити знання про одновимірні та двовимірні масиви на мові програмування С++
* Ознайомитися з алгоритмами обробки даних у масивах
* Навчитися розробляти програми для обробки даних у масивах

# **Виконання роботи:**

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

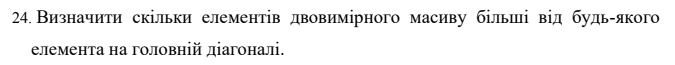
Завдання № 1 VNS Lab 4

* Варіант завдання — 24
* Деталі завдання

Рисунок 1: VNS Lab 4

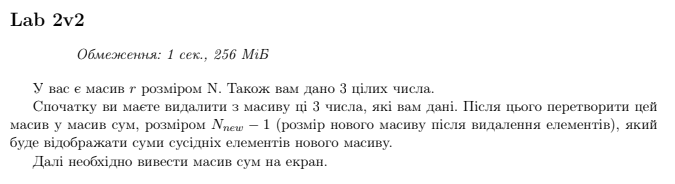
Завдання № 2 VNS Lab 5

* Варіант завдання — 24
* Деталі завдання

Рисунок 2: VNS Lab 5

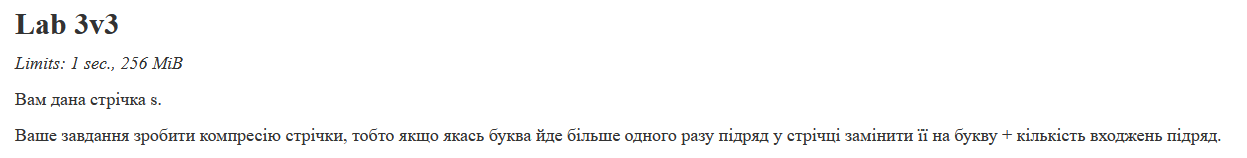
Завдання № 3 Algotester Lab 2

* Варіант завдання - 2
* Деталі завдання

Рисунок 3: Algotester Lab 2

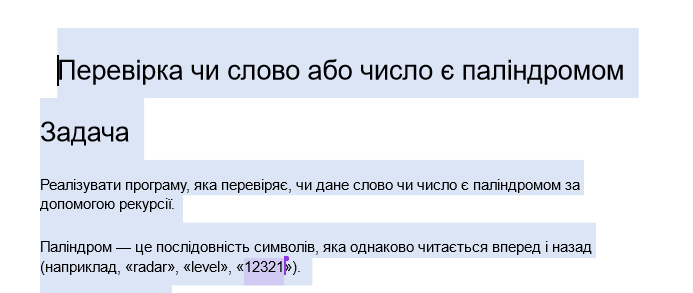
Завдання № 4 Algotester Lab 3

* Варіант завдання - 3
* Деталі завдання

Рисунок 4: Algotester Lab 3

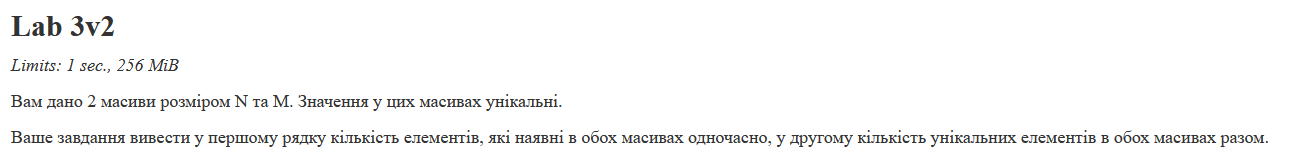
Завдання № 5 Class Practice Work

* Деталі завдання

Рисунок 5: Class Practice Work

Завдання № 6 Self Practice Work

* Деталі завдання

Рисунок 6: Self Practice Work

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

Програма № 1 Кільце

* Планований час на реалізацію — 20 хвилин
* Важливі деталі для врахування в імплементації

1) Використання С-масиву

Програма № 2 Недіагональні елементи, більші за діагональ

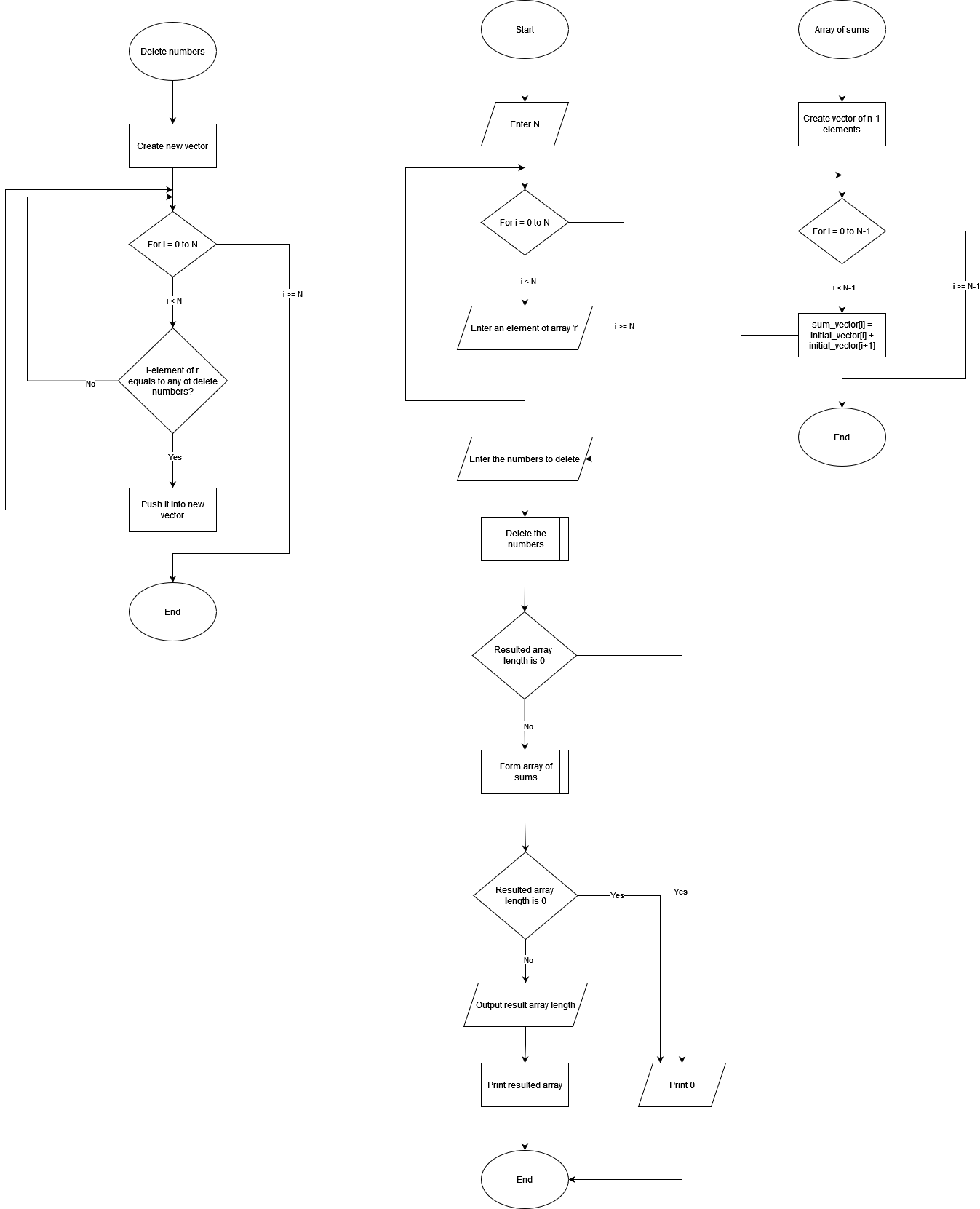
* Планований час на реалізацію — 20 хвилин
* Важливі деталі для врахування в імплементації

1) Використати двовимірний С-масив

2) Використати властивості діагоналі матриці (к-сть елементів діагоналі рівна меншому з кількості рядків або стовпців)

Програма № 3 Чистий масив сум

* Блок-схема

Рисунок 7: Блок-схема до програми № 3 "Чистий масив сум"

* Планований час на реалізацію — 30 хвилин
* Важливі деталі для врахування в імплементації

1) Необхідно перевіряти масиви на нульову кількість елементів

Програма № 4 Компресія стрічки

* Планований час на реалізацію — 10 хвилин
* Важливі деталі для врахування в імплементації

1) Коли нічого не треба стискати (себто є тільки одна послідовна буква, то ми не пишемо TOKEN1, а просто виводимо TOKEN)

Програма № 5 Паліндром

* Планований час на реалізацію — 10 хвилин
* Важливі деталі для врахування в імплементації

1) Зробити перегрузку функції для чисел і стрічок

Програма № 6 Пересічення масивів

* Планований час на реалізацію — 15 хвилин
* Важливі деталі для врахування в імплементації

1) Вивести кількість однакових елементів

2) Вивести кількість різних елементів

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

Завдання № 1 [VNS Lab 4](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/590/files" \l "diff-5fbf1366d34da7194643f971574304b52fc80b1f3be0c76433cefe2455505f74)

#include <iostream>  
#include <algorithm>  
using namespace std;  
  
void print\_from\_k(int\* ring, int size, int k);  
  
// This procedure MUTATES the state of ring and its size  
void ring\_without\_even(int\*\* ring, int\* size);   
  
int main() {  
 int n, k;  
 cout << "Enter size of array: ";  
 cin >> n;  
  
 int\* ring = new int[n];  
 cout << "Enter ring: ";  
 for (int i = 0; i < n; ++i) cin >> ring[i];  
  
 cout << "Enter position: ";  
 cin >> k;  
  
 print\_from\_k(ring, n, k);  
 sort(ring, ring + n);  
 ring\_without\_even(&ring, &n);  
 print\_from\_k(ring, n, k);  
  
 delete[] ring;  
 return 0;  
}  
  
void print\_from\_k(int\* ring, int size, int k){  
 if (k < 0 || k >= size) {  
 cout << "Error: k is out of range" << endl;  
 return;  
 }  
  
 for (int i = k; i < size; ++i) cout << ring[i] << " ";  
 for (int i = 0; i < k; ++i) cout << ring[i] << " ";  
 cout << endl;  
}  
  
// This procedure MUTATES the state of ring and its size  
void ring\_without\_even(int\*\* ring, int\* size) {  
 int new\_size = 0;  
 for (int i = 0; i < \*size; ++i) if ((\*ring)[i] % 2 == 1) ++new\_size;  
  
 int\* new\_ring = new int[new\_size];  
 for (int i = 0, j = 0; i < \*size; ++i)  
 if ((\*ring)[i] % 2 == 1)  
 new\_ring[j++] = (\*ring)[i];  
  
 delete[] \*ring;  
 \*ring = new\_ring;  
 \*size = new\_size;  
}

Завдання № 2 [VNS Lab](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/590/files" \l "diff-472b13b9f495401a012d0a0830d980f3c7dc0e3697330298dda6bc8f51cc6c1e) 5

#include <iostream>  
using namespace std;  
  
int main() {  
 int m, n;  
  
 cout << "Enter matrix size: ";  
 cin >> m >> n;  
  
 int \*\*arr = new int \*[m];  
 for (int i = 0; i < m; i++) arr[i] = new int[n];  
  
 cout << "Enter matrix elements: ";  
 for (int i = 0; i < m; i++)  
 for (int j = 0; j < n; j++)  
 cin >> arr[i][j];  
  
 int lesser = min(m, n);  
 int \*diag = new int[lesser];  
 for (int i = 0; i < lesser; ++i) diag[i] = arr[i][i];  
  
 int min\_diag = diag[0];  
 for (int i = 1; i < lesser; ++i)  
 if (diag[i] < min\_diag) min\_diag = diag[i];  
  
 int bigger\_than\_diag = 0;  
 for (int i = 0; i < m; i++)  
 for (int j = 0; j < n; j++) {  
 if (i == j) continue;  
 if (arr[i][j] > min\_diag) bigger\_than\_diag++;  
 }  
  
 cout << "Number of elements bigger than diagonal: " << bigger\_than\_diag << endl;  
  
 for (int i = 0; i < m; i++) delete[] arr[i];  
 delete[] arr;  
 delete[] diag;  
  
 return 0;  
}

Завдання № 3 [Algotester Lab 2](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/590/files" \l "diff-0e2381e6b66d588fd7193ebf5866263fe103b18e654572f46f9dc705d604b082)

#include <iostream>  
#include <algorithm>  
#include <vector>  
using namespace std;  
  
int main() {  
 int n;  
 cin >> n;  
  
 int\* r = new int[n];  
 vector<int> numsToDelete(3);  
  
 for (int i = 0; i < n; i++)  
 cin >> r[i];  
  
 for (int& numToDelete: numsToDelete)  
 cin >> numToDelete;  
  
 vector<int> filteredArray;  
  
 // Iterate over the array r and check if each element is present in numsToDelete.  
 for (int i = 0; i < n; i++)  
 if (find(numsToDelete.begin(), numsToDelete.end(), r[i])  
 == numsToDelete.end())  
 filteredArray.push\_back(r[i]);  
  
 if (filteredArray.empty()) {  
 cout << 0;  
 return 0;  
 }  
  
 vector<int> arrayOfSums(filteredArray.size() - 1);  
 cout << arrayOfSums.size() << "\n";  
 if (arrayOfSums.empty()) return 0;  
  
 for (int i = 0; i < arrayOfSums.size(); ++i) {  
 arrayOfSums[i] = filteredArray[i] + filteredArray[i + 1];  
 cout << arrayOfSums[i] << " ";  
 }  
 return 0;  
}

Завдання № 4 [Algotester Lab](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/590/files" \l "diff-8d09ef2c536df4e24ec8ffb18cfe6779882902bf3e02d665f880fd78cd79ad7e) 3

#include <iostream>  
#include <string>  
using namespace std;  
  
int main() {  
 string input, result;  
 getline(cin, input);  
  
 char c;  
 int count = 1;  
 for (int i = 0; i < input.size(); ++i) {  
 c = input[i];  
 if (i + 1 < input.size() && input[i+1] == c) ++count;  
 else {  
 result += c;  
 if (count > 1) result += to\_string(count);  
 count = 1;  
 }  
 }  
  
 cout << result;  
 return 0;  
}

Завдання № 5 [Class Practice Work](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/590/files" \l "diff-7fbc0800795bfa0bcd5b5f79a4cf0c0bfdcbc56b1dc406418a752b02a3febe17)

#include <iostream>  
using namespace std;  
  
bool isPalindrome(const string&str, const size\_t start, const size\_t end) {  
 if (start >= end) return true;  
 return str[start] == str[end] && isPalindrome(str, start + 1, end - 1);  
}  
  
bool isPalindrome(const string& str) {  
 return isPalindrome(str, 0, str.length() - 1);  
}  
  
bool isPalindrome(const long long& num) {  
 return isPalindrome(to\_string(num));  
}  
  
int main() {  
 string str;  
 cout << "Enter a string: ";  
 cin >> str;  
  
 cout << str << " is " << (isPalindrome(str) ? "" : "not ") << "a palindrome." << endl;  
  
 return 0;  
}

Завдання № 6 [S](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/590/files" \l "diff-5c62c84b1e5edc053eb95129e1b1138f48d76038233049b78a2b8e9cdfccc3b5)elf Practice Work

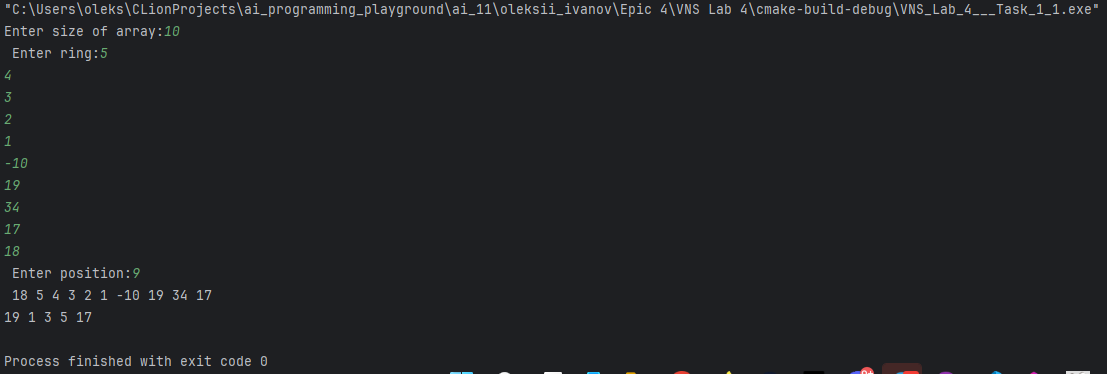
#include <iostream>  
using std::cin, std::cout;  
  
// Lab 3v2  
  
int main() {  
 int n, m;  
 int\* arr1, \*arr2;  
  
 cin >> n;  
 arr1 = new int[n];  
 for (int i = 0; i < n; ++i) cin >> arr1[i];  
  
 cin >> m;  
 arr2 = new int[m];  
 for (int i = 0; i < m; ++i) cin >> arr2[i];  
  
 int unique\_count = 0, common\_count = 0;  
  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < m; ++j) {  
 if (arr1[i] == arr2[j]) {  
 common\_count++;  
 break;  
 }  
 }  
 }  
  
 unique\_count = n + m - common\_count;  
  
 cout << common\_count << '\n' << unique\_count << '\n';  
  
 delete[] arr1;  
 delete[] arr2;  
 return 0;  
}

## [**Посилання на pull-request**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/590)

## **4. Результати виконання завдань, тестування та фактично витрачений час:**

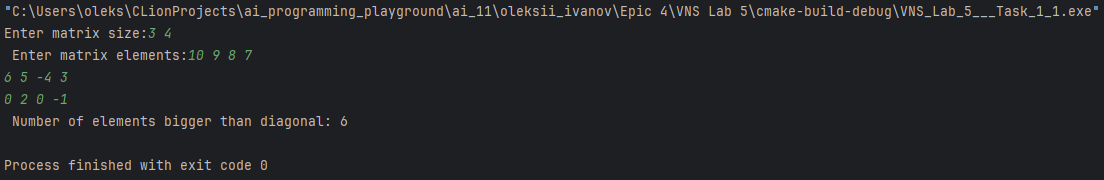
Завдання № 1 VNS Lab 4

Час, витрачений на виконання завдання — 20 хвилин

Рисунок 8: VNS Lab 4 Execution Results

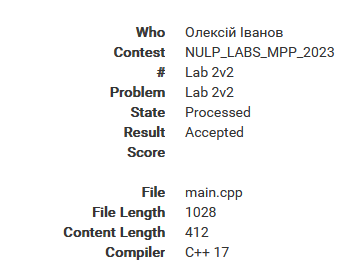
Завдання № 2 VNS Lab 5

Час, витрачений на виконання завдання — 15 хвилин

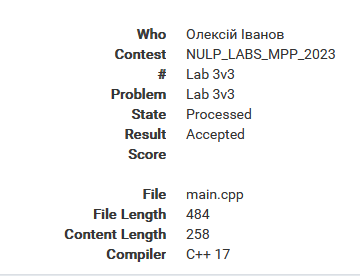
Рисунок 9: VNS Lab 5 Execution Results

Завдання № 3 Algotester Lab 2

Час, витрачений на виконання завдання — 15 хвилин

Рисунок 10: Algotester Lab 2 Execution Results

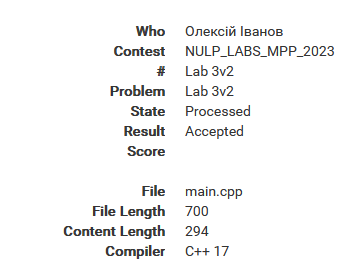
Завдання № 4 Algotester Lab 3

Рисунок 11: Algotester Lab 2 Execution Results

Час, витрачений на виконання завдання — 10 хвилин

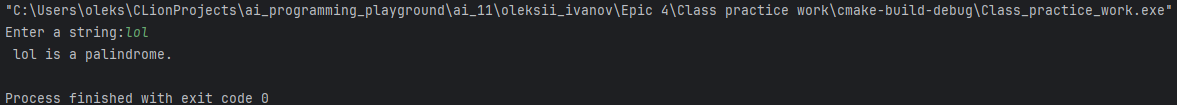
Завдання № 5 Self practice work

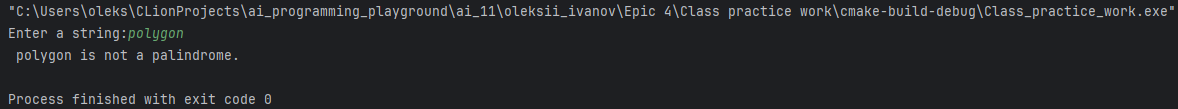
Час, витрачений на виконання завдання — 30 хвилин

Рисунок 12: Self practice work Execution Results

Завдання № 6 Class practice work

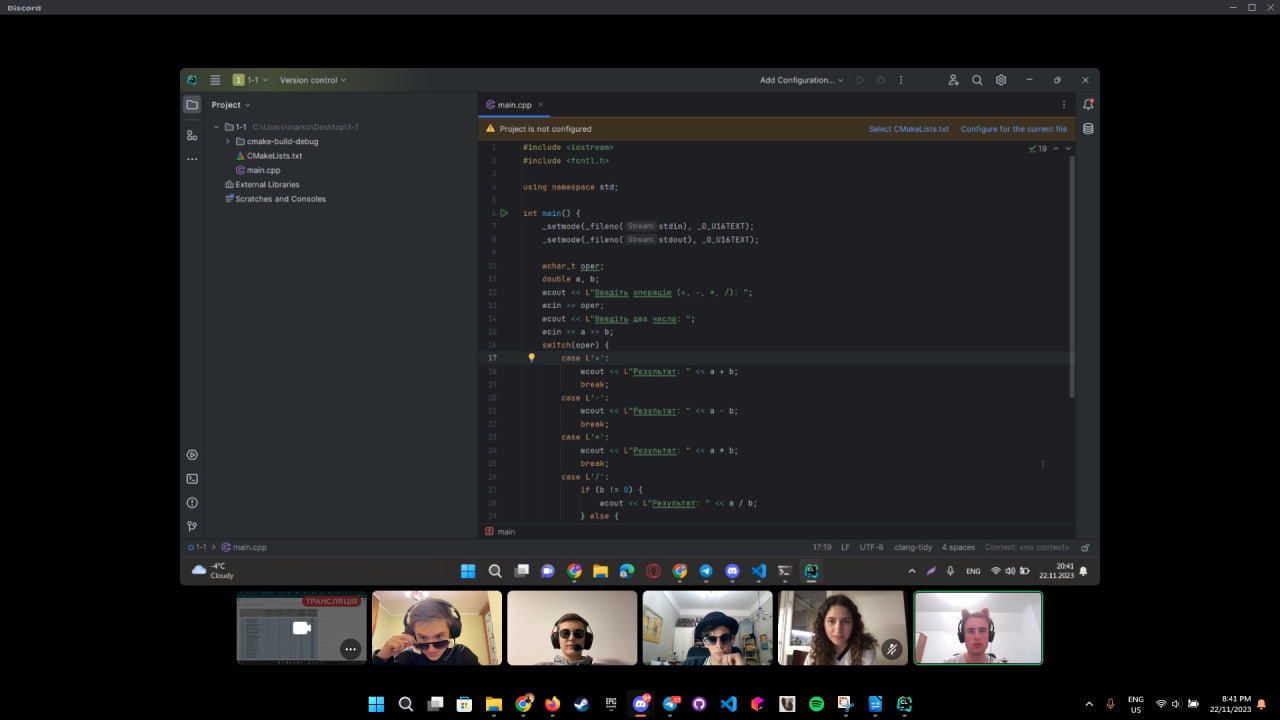
Час, витрачений на виконання завдання — 10 хвилин

Рисунок 13: Self practice work Execution Results for positive output

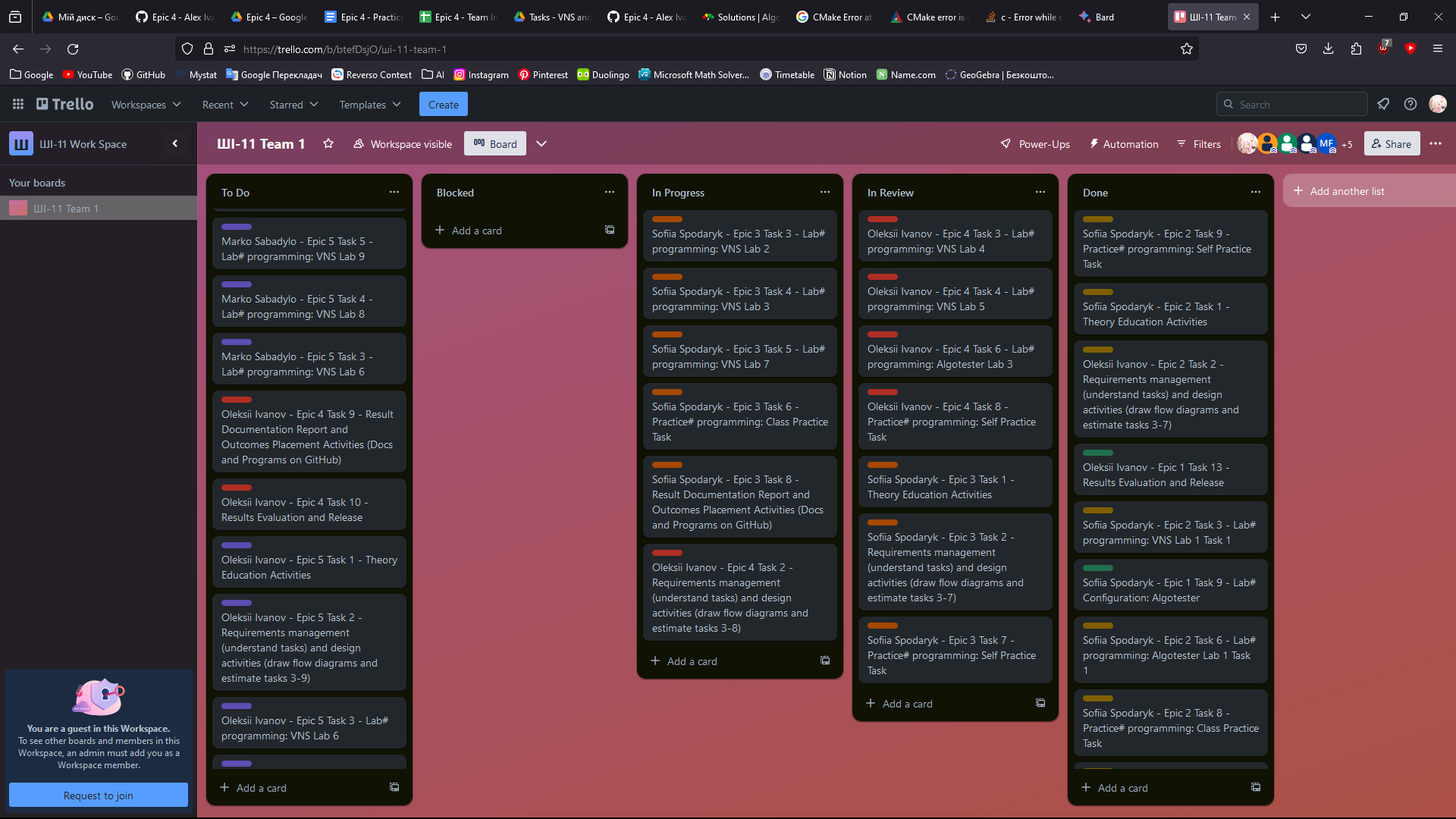
Рисунок 14: Self practice work Execution Results for negative output

## **5. Кооперація з командою:**

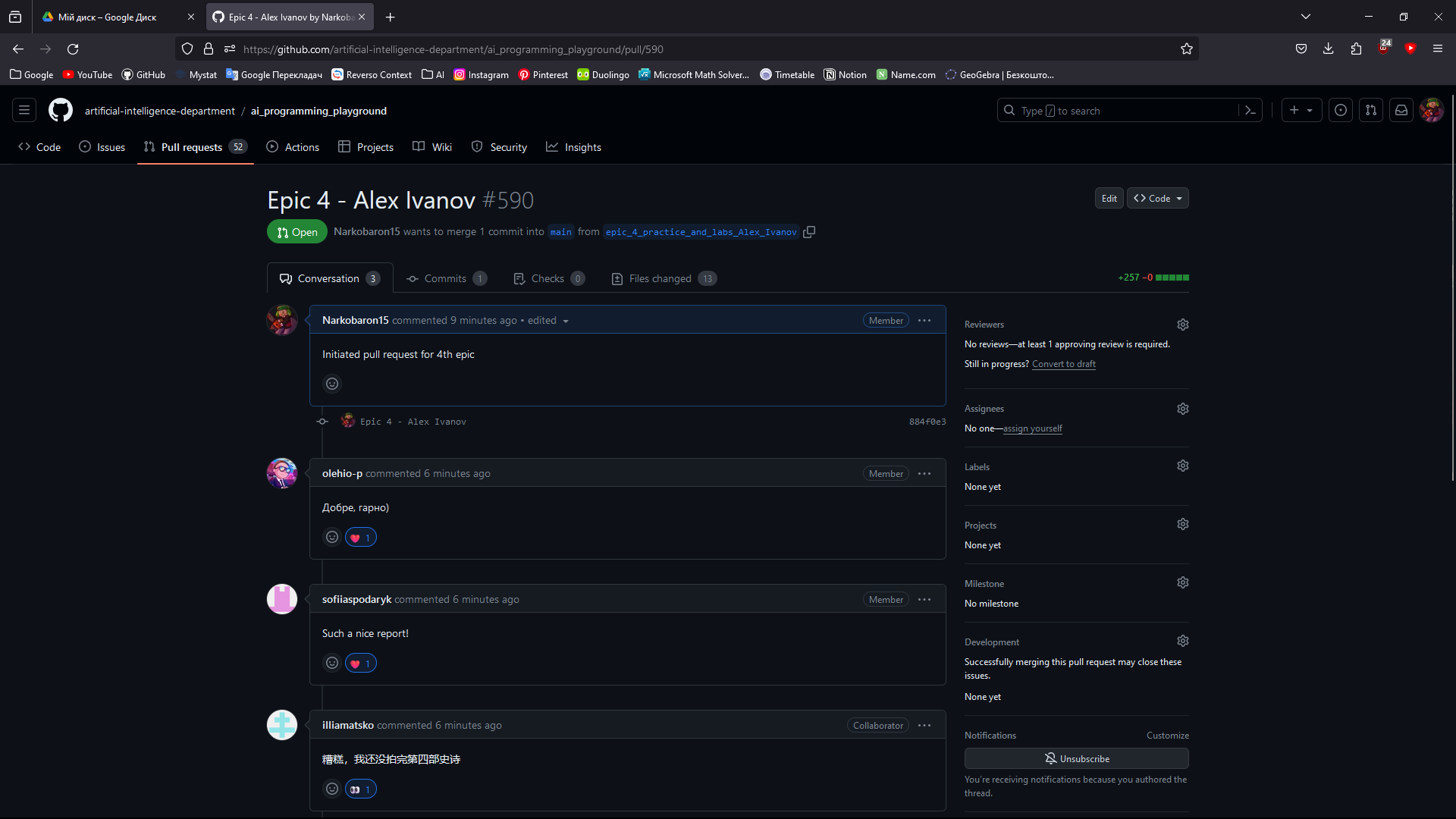
* Скрін із зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло

Рисунок 15: Скріншот командної роботи над проектом

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

Рисунок 16: Скріншот прогресу по трелло

# **Висновки:**

Рисунок 17: Скріншот коментарів до пул-реквесту

Під час даної лабораторної роботи було розглянуто наступні питання:

* **Одновимірні масиви:**
  + Поняття одновимірного масиву.
  + Види одновимірних масивів.
  + Ініціалізація одновимірних масивів.
  + Доступ до елементів одновимірних масивів.
  + Обробка одновимірних масивів.
* **Двовимірні масиви:**
  + Поняття двовимірного масиву.
  + Види двовимірних масивів.
  + Ініціалізація двовимірних масивів.
  + Доступ до елементів двовимірних масивів.
  + Обробка двовимірних масивів.

У процесі виконання лабораторної роботи були отримані наступні висновки:

* Одновимірні масиви - це структури даних, які дозволяють зберігати впорядковану послідовність елементів одного типу.
* Двовимірні масиви - це структури даних, які дозволяють зберігати впорядковану послідовність елементів одного типу, розташованих у вигляді таблиці.

Виконання лабораторної роботи дозволило закріпити теоретичні знання з розділу «Прості структури даних. Одновимірні масиви. Двовимірні Масиви. Алгоритми обробки.» та отримати практичні навички роботи з одновимірними та двовимірними масивами.

Особливо важливими є наступні навички:

* Ініціалізація одновимірних та двовимірних масивів.
* Доступ до елементів одновимірних та двовимірних масивів.
* Обробка одновимірних та двовимірних масивів.
* Поняття динамічного масиву та двовимірного зубчатого масиву.

Ці навички необхідні для подальшого вивчення програмування та розробки більш комплексних програм.