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

Попович Олег Васильович

# **Тема роботи:**

Прості структури даних у С++. Ознайомлення з одновимірними та двовимірними масивами. Знайомство з базовими алгоритми обробки даних

# **Мета роботи:**

Ознайомитися з одновимірними та двовимірними у С++. Опрацювати та застосувати алгоритми обробки для масивів.

Виконати задані лабораторні роботи.

Виконати задане практичне завдання.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Масиви
* Тема №2: Багатовимірні масиви
* Тема №3: Алгоритми роботи з масивами на мові C++

1. Індивідуальний план опрацювання теорії:

* Тема №1: Масиви.
  + Джерела Інформації
* “The C++ Programming Language” by Bjarne Stroustrup
* “C++ For Dummies” by Stephen R. Davis
  + Що опрацьовано:
* **Поняття масиву**
* **Створення та ініціалізація масивів.** .
* **Доступ до елементів масиву**
  + Статус: Ознайомлений
  + Початок опрацювання теми: 05.10.2023
  + Звершення опрацювання теми: 12.10.2023
* Тема №2: Багатовимірні масиви.
  + Джерела Інформації:
* “The C++ Programming Language” by Bjarne Stroustrup
* “C++ For Dummies” by Stephen R. Davis
  + Що опрацьовано:
* **Поняття багатовимірного масиву.**
* **Створення та ініціалізація багатовимірних масивів.**
* **Доступ до елементів багатовимірного масиву.**
* **Алгоритми обходу багатовимірних масивів.**
  + Статус: Ознайомлений
  + Початок опрацювання теми: 05.10.2023
  + Звершення опрацювання теми: 12.10.2023
* Тема №3: Рекурсія
  + Джерела Інформації:
* “The C++ Programming Language” by Bjarne Stroustrup
* “C++ For Dummies” by Stephen R. Davis
  + Що опрацьовано:
* Алгоритми сортування масивів.
* Алгоритми пошуку елементів у масиві.
* Алгоритми обчислення суми або середнього значення елементів масиву.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 05.10.2023
  + Звершення опрацювання теми: 12.10.2023

# **Виконання роботи:**

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

Завдання **№1 VNS Lab 4 Task 1**

* Реалізувати з використанням масиву стек (перший прийшов, останній пішов),

для чого організувати додавання, знищення елементів з масиву і друк масиву

після кожної операції.

Завдання **№2  VNS Lab 5 Task 1.**

* У двовимірному масиві знайти середнє арифметичне першого стовпця й

кількість елементів у кожному з наступних стовпців, що перевищують

середнє арифметичне попереднього стовпця.

Завдання **№3  Algotester Lab 3v3.**

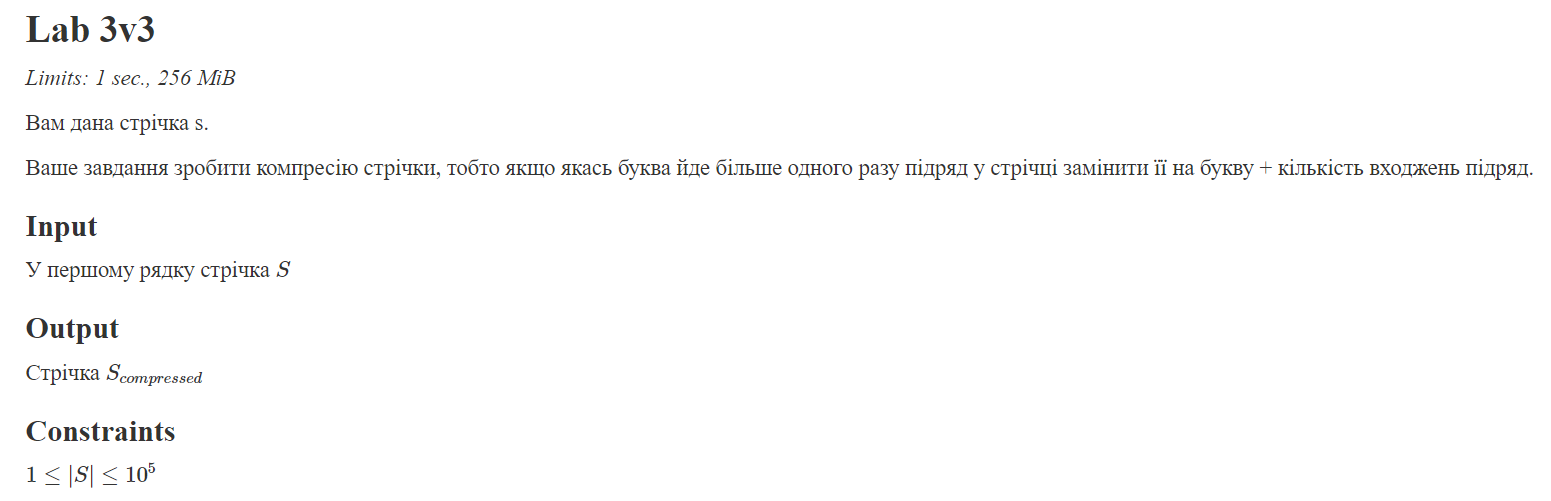


Рисунок 1 Завдання до Algotester Lab 3

Завдання **№4  Practice Work**

* 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:

* Визначення функції: Реалізуйте рекурсивну функцію isPalindrome, яка перевіряє, чи заданий рядок є паліндромом.
* Приклад визначення функції: bool isPalindrome(const string& str, int start, int end);
* Перевантаження функцій: Перевантажте функцію isPalindrome для роботи з цілими значеннями. bool isPalindrome(ціле число);
* 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.

Завдання **№5 Self Practice Work**

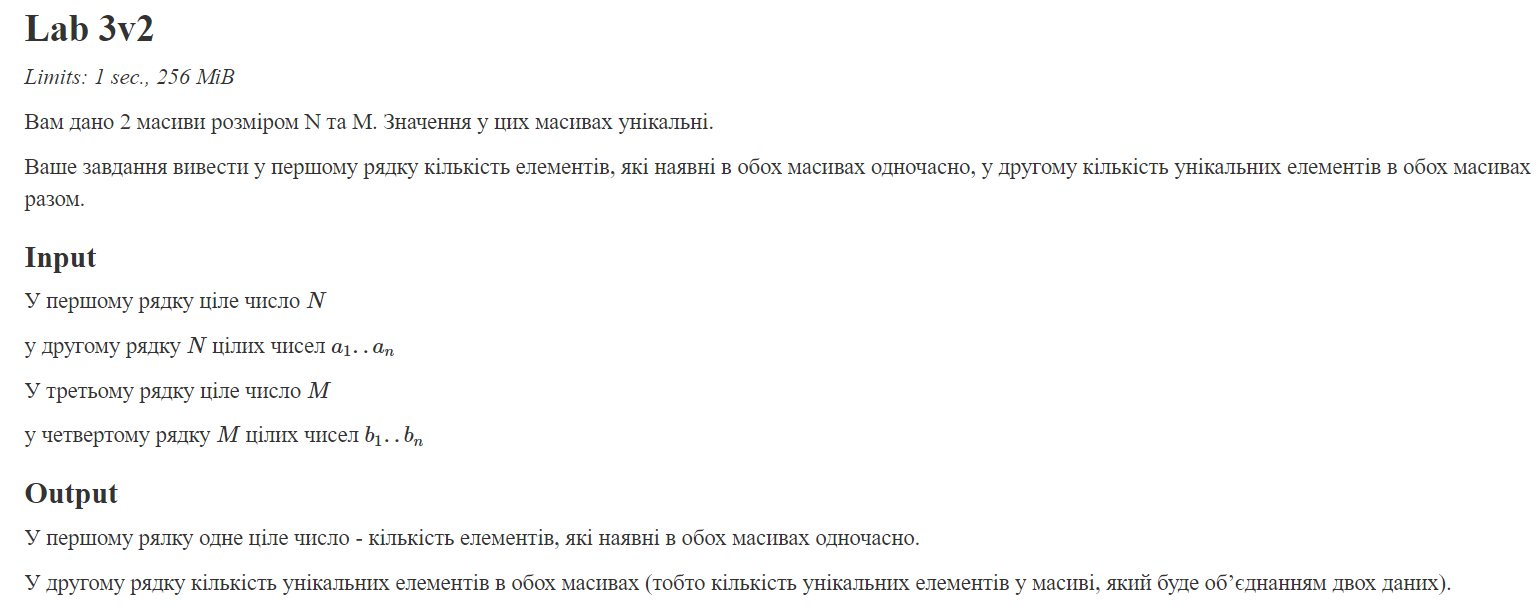


Рисунок 2 Завдання до Self Practice

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

Програма **№1 VNS Lab 4 Task 1**

* Планований час на реалізацію: **5 хвилини**.

Програма **№2 VNS Lab 5 Task 1.**

* Блок-схема

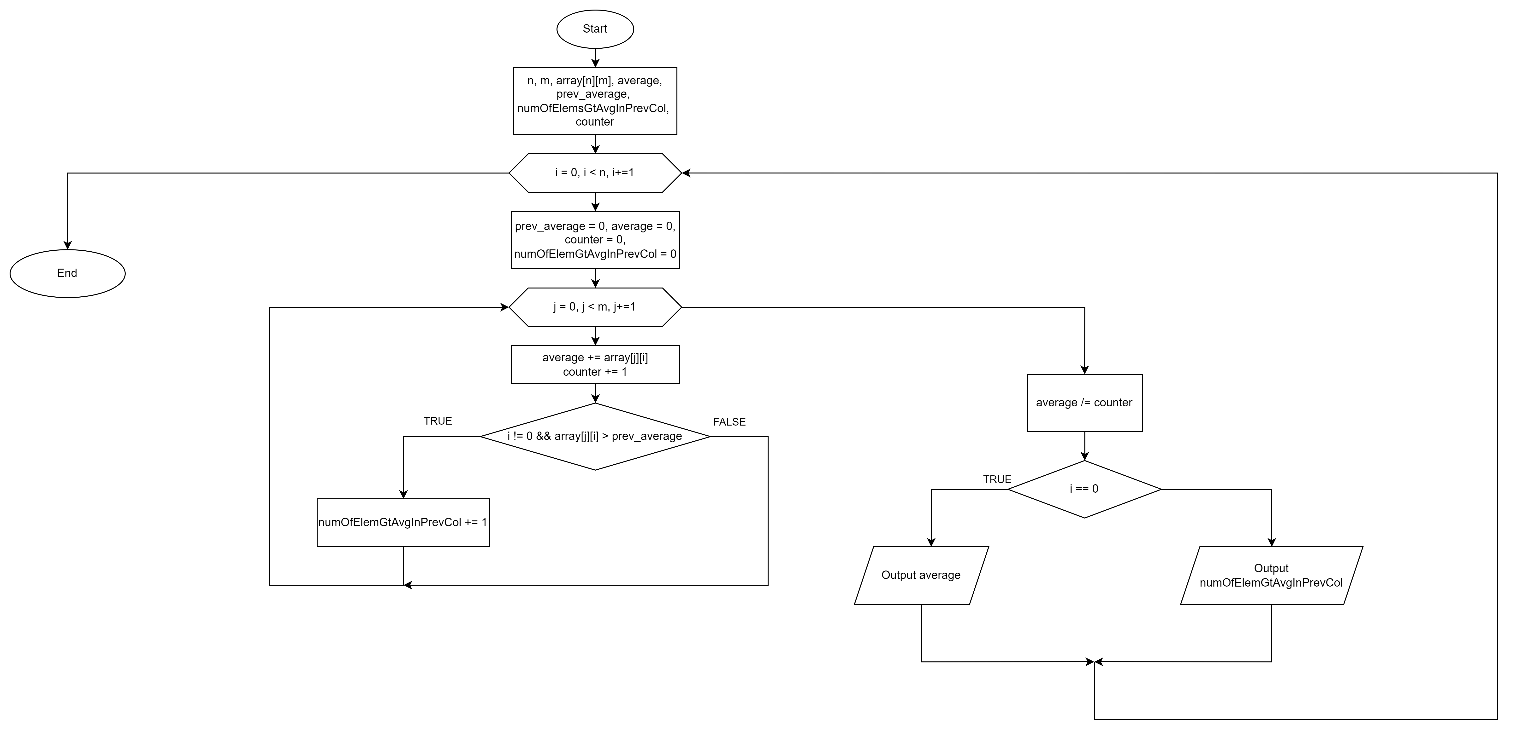


Рисунок 3 Блок схема до програми №2

* Планований час на реалізацію: **10 хвилини**.

Програма **№3 Algotester Lab 4 Task 1.**

* Блок-схема

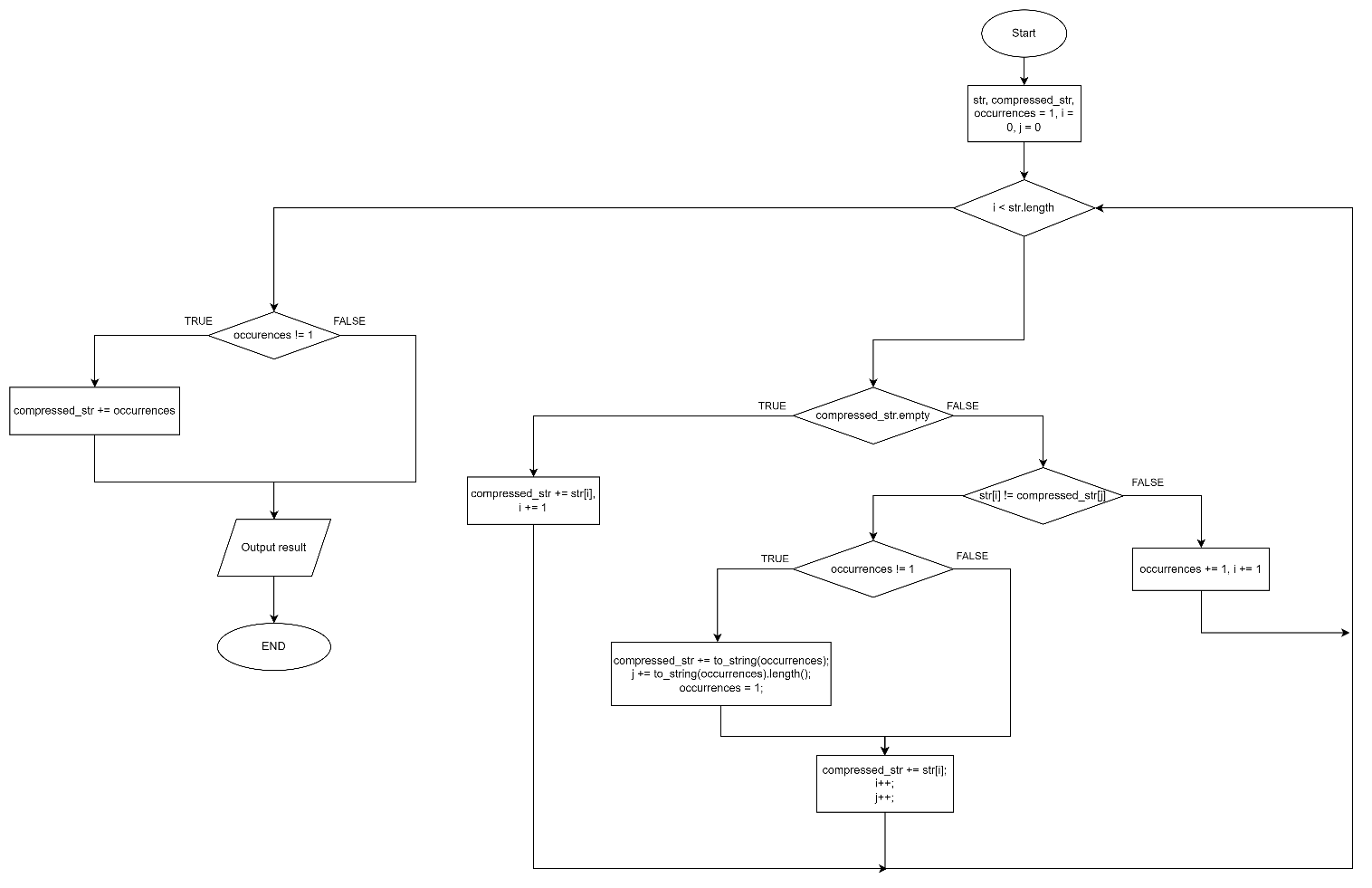


Рисунок 4 Блок схема до програми №3

* Планований час на реалізацію: **20 хвилин**.

Програма **№4 Practice Work.**

* Планований час на реалізацію: **10 хвилин**.

Програма **№6 Self Practice Work.**

* Планований час на реалізацію: **25 хвилин**.

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

Завдання **№1 VNS Lab 4 Task 1**

Деталі по програмі + Вставка з кодом з підписами до вставки

//  
// Created by olehio-p on 11/29/2023.  
//  
#include <algorithm>  
#include <stdexcept>  
#include <string>  
#include <iostream>  
  
  
class Stack{  
 private:  
 int capacity;  
 int\* array;  
 int end;  
 public:  
 void push(int elem);  
 int pop();  
 void printStack();  
 explicit Stack(int \_capacity = 100) {  
 end = 0;  
 capacity = \_capacity;  
 array = new int[capacity];  
 }  
 ~Stack() {  
 delete[] array;  
 }  
};  
  
void Stack::push(int elem) {  
 if(end >= capacity) {  
 int\* new\_array = new int[capacity + (int)(capacity \* 0.75)];  
 std::copy(array, array+end, new\_array);  
 capacity += (int)(capacity \* 0.75);  
 delete[] array;  
 array = new\_array;  
 }  
 array[end++] = elem;  
}  
  
int Stack::pop() {  
 if(end > 0) {  
 return array[end--];  
 } else {  
 throw std::out\_of\_range("The Stack is empty");  
 }  
}  
  
void Stack::printStack() {  
 std::string str = "Stack{";  
 if(end == 0) {  
 std::cout << "Stack{}" << std::endl;  
 return;  
 }  
 for(int i = 0; i < end; i++) {  
 str += std::to\_string(array[i]);  
 if(i != end-1) {  
 str += ", ";  
 } else {  
 str += "}";  
 }  
 }  
 std::cout << str << std::endl;  
}  
  
int main(int argc, char const \*argv[])  
{  
 Stack stack = \*new Stack(10);  
 stack.push(10);  
 stack.push(9);  
 stack.printStack();  
 stack.pop();  
 stack.printStack();  
 stack.pop();  
 stack.printStack();  
 stack.push(10);  
 stack.push(9);  
 stack.push(8);  
 stack.push(7);  
 stack.push(6);  
 stack.push(5);  
 stack.push(4);  
 stack.push(3);  
 stack.push(2);  
 stack.push(1);  
 stack.push(0);  
 stack.printStack();  
 return 0;  
}

Рисунок 5 Код до програми №1

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_oleh_popovych/ai_11/oleh_popovych/epic_4/vns_lab_4_task_1_variant_22_oleh_popovych.cpp>

Завдання **№2  VNS Lab 5 Task 1**

Деталі по програмі + Вставка з кодом з підписами до вставки

//  
// Created by olehio-p on 11/29/2023.  
//  
#include <iostream>  
  
  
int main(int argc, char const \*argv[])  
{  
 const int n = 3, m = 3;  
 int array[n][m] = {{1, 2, 3},  
 {4, 5, 6},  
 {7, 8, 9}};  
  
 int average;  
 int prev\_average;  
 int numOfElemsGtAvgInPrevCol;  
 int counter;  
 for(int i = 0; i < n; i++) {  
 prev\_average = average;  
 average = 0;  
 counter = 0;  
 numOfElemsGtAvgInPrevCol = 0;  
 for(int j = 0; j < m; j++) {  
 average += array[j][i];  
 counter++;  
 if(i != 0 && array[j][i] > prev\_average) {  
 numOfElemsGtAvgInPrevCol++;  
 }  
 }  
 average /= counter;  
 if(i == 0) {  
 std::cout << "The average of the first column: " << average << std::endl;  
 } else {  
 std::cout << "Number of elems greater than average in previous column: "  
 << numOfElemsGtAvgInPrevCol << std::endl;  
 }  
 }  
 return 0;  
}

Рисунок 6 Код до програми №2

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_oleh_popovych/ai_11/oleh_popovych/epic_4/vns_lab_5_task_1_variant_22_oleh_popovych.cpp>

Завдання **№3  Algotester Lab 3 Task 1**

Деталі по програмі + Вставка з кодом з підписами до вставки

//  
// Created by olehio-p on 11/29/2023.  
//  
#include <iostream>  
#include <string>  
  
  
using namespace std;  
int main(int argc, char const \*argv[])  
{  
 string str, compressed\_str;  
 cin >> str;  
  
 int occurrences = 1;  
 int i = 0, j = 0;  
 while(i < str.length()) {  
 if(compressed\_str.empty()) {  
 compressed\_str += str[i];  
 i++;  
 } else {  
 if(str[i] != compressed\_str[j]) {  
 if(occurrences != 1) {  
 compressed\_str += to\_string(occurrences);  
 j += to\_string(occurrences).length();  
 occurrences = 1;  
 continue;  
 }  
 compressed\_str += str[i];  
 i++;  
 j++;  
 } else {  
 occurrences++;  
 i++;  
 }  
 }  
 }  
 if(occurrences != 1)  
 compressed\_str += to\_string(occurrences);  
 cout << compressed\_str << endl;  
 return 0;  
}

Рисунок 7 Код до програми №3

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_oleh_popovych/ai_11/oleh_popovych/epic_4/algotester_lab_3_variant_3_oleh_popovych.cpp>

Завдання **№4  Practice Work**

Деталі по програмі + Вставка з кодом з підписами до вставки

//  
// Created by olehio-p on 11/2/2023.  
//  
#include <iostream>  
#include <string>  
  
bool isPalindrome(const std::string&, int, int);  
bool isPalindrome(int);  
bool is\_number(const std::string&);  
  
int main(int argc, char const \*argv[])  
{  
 std::string word;  
 bool is\_palindrome;  
  
 std::cout << "Enter a word or number to check if it's a palindrome: ";  
 std::cin >> word;  
  
 if(is\_number(word))  
 is\_palindrome = isPalindrome(stoi(word));  
 else  
 is\_palindrome = isPalindrome(word, 0, word.length()-1);  
  
 std::string answer = word + (is\_palindrome ? " is " : " isn't ") + "palindrome";  
 std::cout << answer << std::endl;  
 return 0;  
}  
  
bool isPalindrome(const std::string& str, int start, int end)  
{  
 if(start < end)  
 return tolower(str[start]) == tolower(str[end])  
 && isPalindrome(str, ++start, --end);  
 else  
 return true;  
}  
  
bool isPalindrome(int number)  
{  
 std::string str = std::to\_string(number);  
 return isPalindrome(str, 0, str.length()-1);  
}  
  
bool is\_number(const std::string& str)  
{  
 std::string::const\_iterator it = str.begin();  
 while(it != str.end() && std::isdigit(\*it))  
 ++it;  
 return !str.empty() && it == str.end();  
}

Рисунок 8 Код до програми №4

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_oleh_popovych/ai_11/oleh_popovych/epic_4/practice_work_task_1_oleh_popovych.cpp>

Завдання **№5  Self Practice Work**

Деталі по програмі + Вставка з кодом з підписами до вставки

//  
// Created by olehio-p on 11/29/2023.  
//  
#include <iostream>  
  
  
using namespace std;  
int main(int argc, char const \*argv[])  
{  
 int n, m;  
 cin >> n;  
 int a[n];  
  
 for(int i = 0; i < n; i++) {  
 cin >> a[i];  
 }  
 cin >> m;  
  
 int b[m];  
 for(int i = 0; i < m; i++) {  
 cin >> b[i];  
 }  
  
  
 int intersection = 0;  
 for(int i = 0; i < n; i++) {  
 for(int j = 0; j < m; j++) {  
 if(a[i] == b[j]) {  
 intersection++;  
 break;  
 }  
 }  
 }  
  
 int i = 0, j = 0, unique = 0;  
 while(i < n && j < m) {  
 if(a[i] < b[j]) {  
 unique++;  
 i++;  
 } else if(a[i] > b[j]) {  
 unique++;  
 j++;  
 } else {  
 unique++;  
 i++;  
 j++;  
 }  
 }  
 while(i < n) {  
 unique++;  
 i++;  
 }  
 while(j < m) {  
 unique++;  
 j++;  
 }  
 cout << intersection << endl;  
 cout << unique << endl;  
 return 0;  
}

Рисунок 9 Код до програми № 5

Посилання на файл програми у пул-запиті GitHub

## <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_oleh_popovych/ai_11/oleh_popovych/epic_4/self_practice_work_algotester_task_1_oleh_popovych.cpp>

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

Завдання **№1 VNS Lab 4 Task 1**

Деталі виконання і тестуванню програми

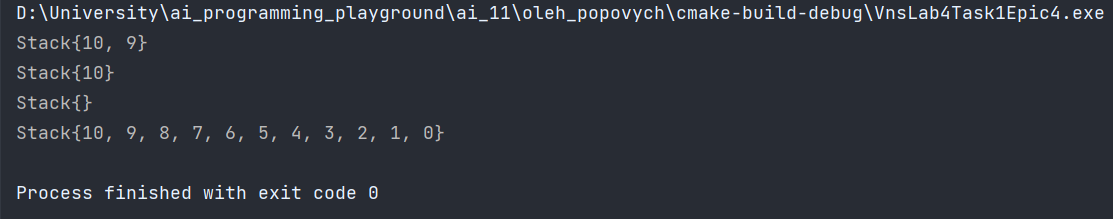


Рисунок 10 Результат виконання програми №1

Час затрачений на виконання завдання: **5 хвилин**.

Завдання **№2  VNS Lab 5 Task 1.**

Деталі виконання і тестуванню програми

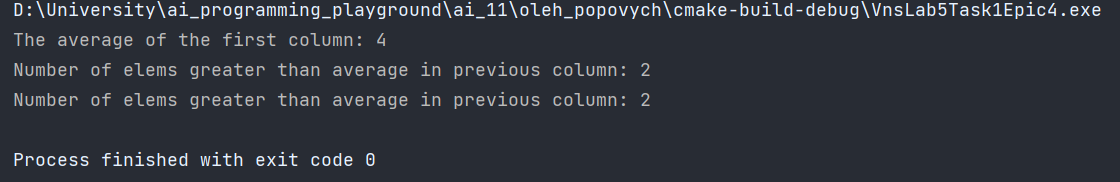


Рисунок 11 Результат виконання програми №2

Час затрачений на виконання завдання: **10 хвилини.**

Завдання **№3  Algotester Lab 3 Task 1**

Деталі виконання і тестуванню програми

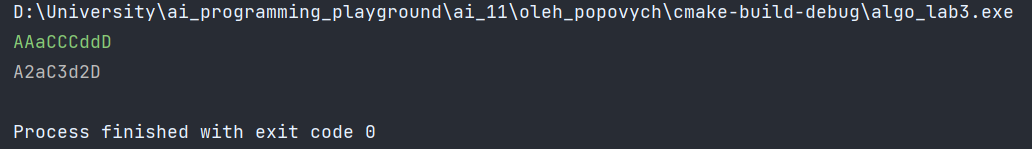
 

Рисунок 12 Результат виконання програми №3

Час затрачений на виконання завдання: **15 хвилин.**

Завдання **№4  Practice Work.**

Деталі виконання і тестуванню програми

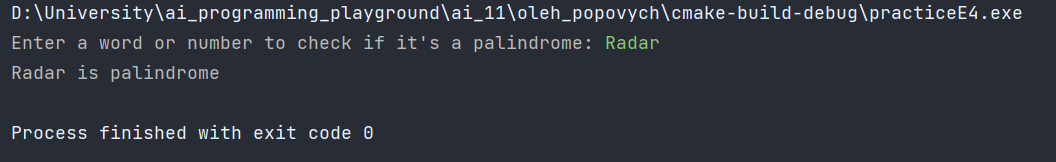


Рисунок 13 Результат виконання програми №4

Час затрачений на виконання завдання: **15 хвилин.**

Завдання **№5  Self Practice Work.**

Деталі виконання і тестуванню програми



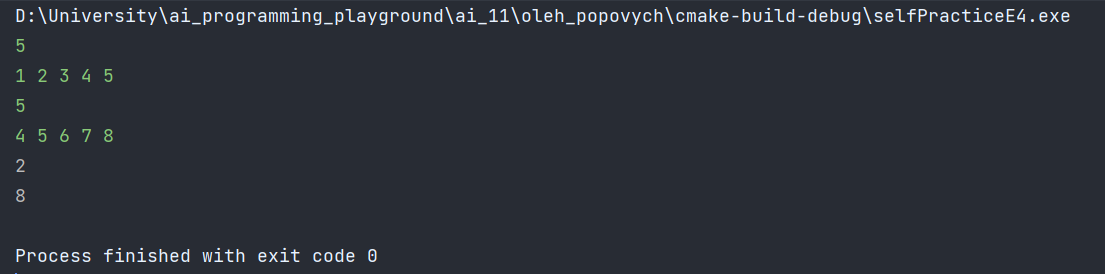
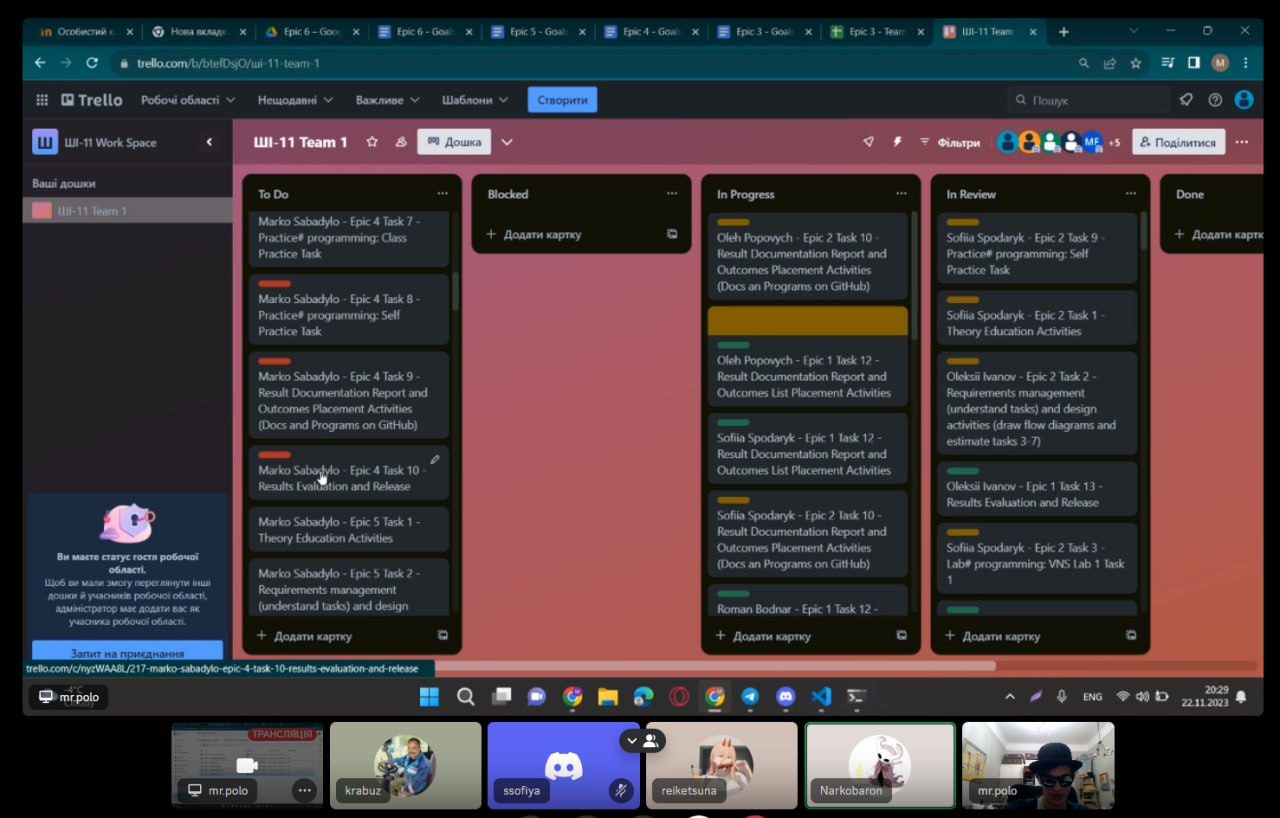


Рисунок 14 Результат виконання програми №5

Час затрачений на виконання завдання: **25 хвилин.**

## **6. Кооперація з командою:**

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



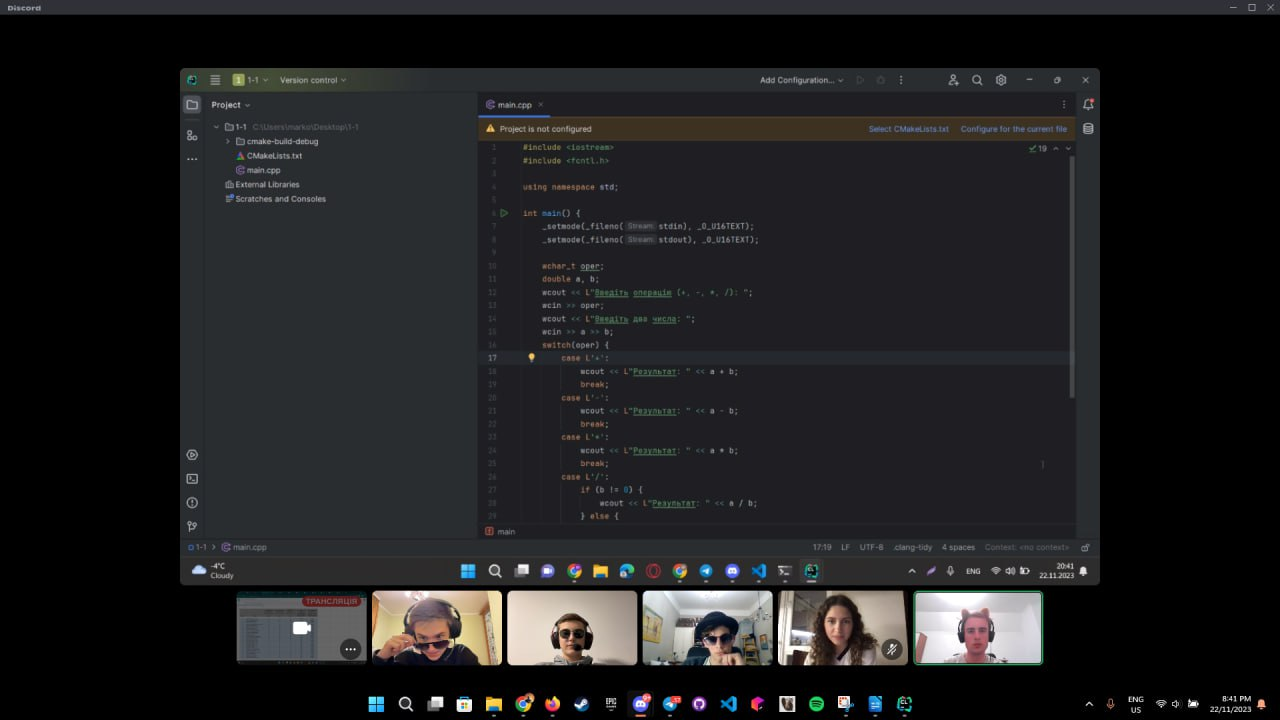


Рисунок 14 Cкріни зустрічі по обговоренню задач Епіку 4

# **Висновки:**

Ознайомився з масивами у С++. Під час виконання лабораторних задач застосовував їх та працював з алгоритмами обробки масивів.

Виконав задані лабораторні роботи.

Виконав задане практичне завдання.