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-14

Кисіль Сергій Дмитрович

# **Тема роботи:**

# Ознайомлення з одновимірними та багатовимірними масивами, ініціалізація одновимірних та багатовимірних масивів, принципи доступу до елементів масиву, вказівники та посилання, адресна арифметика, поняття про динамічні масиви, конструкції std::array та std::vector

# **Мета роботи:**

Ознайомитись з різними видами масивів, отримати поняття про багатовимірні масиви. Ознайомитесь з вказівниками та посиланнями(що реалізовані на їх основі), зрозуміти принцип роботи та випадки застосування вказівників та посилань, отримати поняття про адресну арифметику і її застосування для індексації масивів, ознайомитись з динамічними масивами та методом їх ініціалізації, отримати поняття про модифіковані масиви std::array та std::vector

Завдання:

* Epic 4 Task 1 - Theory Education Activities
* Epic 4 Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-8)
* Epic 4 Task 3 - Lab# programming: VNS Lab 4
* Epic 4 Task 4 - Lab# programming: VNS Lab 5
* Epic 4 Task 5 - Lab# programming: Algotester Lab 2
* Epic 4 Task 6 - Lab# programming: Algotester Lab 3
* Epic 4 Task 7 - Practice# programming: Class Practice Task
* Epic 4 Task 8 - Practice# programming: Self Practice Task
* Epic 4 Task 9 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)
* Epic 4 Task 10 - Results Evaluation and Release

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

⦁ Тема №1: Одновимірні та багатовимірні статичні масиви

⦁ Тема №2: Вказівники

⦁ Тема №3: Посилання

⦁ Тема №4: Адресна арифметика, доступ до елементів масива через вказівники

⦁ Тема №5: Динамічні одновимірні масиви, початкове уявлення про динамічні багатовимірні масиви

⦁ Тема №6: Конструкції std::array та std::vector

1. Індивідуальний план опрацювання теорії:

Тема №1: Одновимірні та багатовимірні статичні масиви

* + Джерела Інформації:

<https://acode.com.ua/urok-77-masyvy/>

<https://acode.com.ua/urok-78-fiksovani-masyvy/>

<https://acode.com.ua/urok-81-bagatovymirni-masyvy/>

<https://acode.com.ua/urok-90-dynamichni-masyvy/>

Team meats

Одногрупники та однопотоківці

* + Що опрацьовано:

Ініціалізація одновимірних масивів

Ініціалізація багатовимірних масивів

* + Статус: Ознайомлений/ Ознайомлений частково / Не ознайомлений
  + Початок опрацювання теми: 01.11.2023
  + Завершення опрацювання теми: 29.11.2023

Тема №2: Вказівники

* + Джерела Інформації:

Team meats

<https://acode.com.ua/urok-84-vkazivnyky/>

<https://acode.com.ua/urok-85-nulovi-vkazivnyky/>

<https://acode.com.ua/urok-91-vkazivnyky-i-const/>

<https://acode.com.ua/urok-110-vkazivnyky-na-funktsiyi/>

(також кожну тему опрацьовував на <https://vns.lpnu.ua/pluginfile.php?file=%2F1116827%2Fmod_resource%2Fcontent%2F1%2Fkonspekt_lekcyi_ukr_1.pdf> )

* + Що опрацьовано:

Поняття про адреси змінних в пам’яті

Поняття про застосування поінтерів для збереження адрес

Використання поінтерів в масивах та функціях

* + Статус: Ознайомлений/ Ознайомлений частково / Не ознайомлений
  + Початок опрацювання теми: 01.11.2023
  + Завершення опрацювання теми: 01.11.2023

Тема №3: Посилання

* + Джерела Інформації:

Team meats

Одногрупники

<https://acode.com.ua/urok-92-posylannya/>

<https://acode.com.ua/urok-93-posylannya-i-const/>

* + Що опрацьовано:

Ознайомлення з посиланнями(референсами) та їх застосуванням як псевдонімів змінних

Поняття про принцип реалізації референсів через поінтери

Статус: Ознайомлений/ Ознайомлений частково / Не ознайомлений

* + Початок опрацювання теми: 01.11.2023
  + Завершення опрацювання теми: 29.11.2023

Тема №4: Адресна арифметика, доступ до елементів масива через вказівники

* + Джерела Інформації:

Одногрупники

<https://acode.com.ua/urok-94-operator-dostupu-do-chleniv-cherez-vkazivnyk/>

<https://acode.com.ua/urok-86-vkazivnyky-i-masyvy/>

* + Що опрацьовано:

Зв’язок поняття поінтера з масивами

Доступ до елементів масива через вказівники

Статус: Ознайомлений/ Ознайомлений частково / Не ознайомлений

* + Початок опрацювання теми: 01.11.2023
  + Звершення опрацювання теми: 29.11.2023

Тема №5: Динамічні одновимірні масиви, початкове уявлення про динамічні багатовимірні масиви

* + Джерела Інформації:

<https://acode.com.ua/urok-90-dynamichni-masyvy/>

* + Що опрацьовано:

Ініціалізація динамічного одновимірного масиву

* + Статус: Ознайомлений/ Ознайомлений частково / Не ознайомлений
  + Початок опрацювання теми: 01.11.2023
  + Звершення опрацювання теми: 29.11.2023

Тема №6: Конструкції std::array та std::vector

* + Джерела Інформації:

<https://acode.com.ua/urok-98-vvedennya-v-std-array/>

<https://acode.com.ua/urok-99-vvedennya-v-std-vector/>

* + Що опрацьовано:

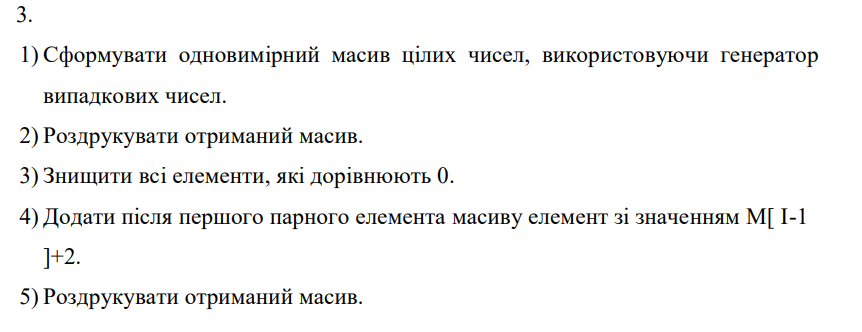
Std::array як модифікація статичного масиву

Std::vector як модифікація динамічного масиву

* + Статус: Ознайомлений/ Ознайомлений частково / Не ознайомлений
  + Початок опрацювання теми: 01.11.2023
  + Звершення опрацювання теми: 29.11.2023

# **Виконання роботи:**

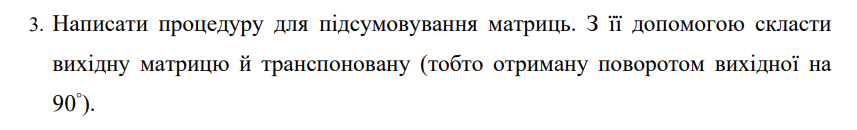
## **1. Опрацювання завдання та вимог до програм та середовища:**

* Завдання №1 Лаб# Програмування: VNS Lab4 (Завдання 1)
* Варіант завдання: 3
* Деталі завдання:

Малюнок(1) - умова завдання 1.

Завдання №2 Лаб# Програмування: VNS Lab5 (Завдання 1)

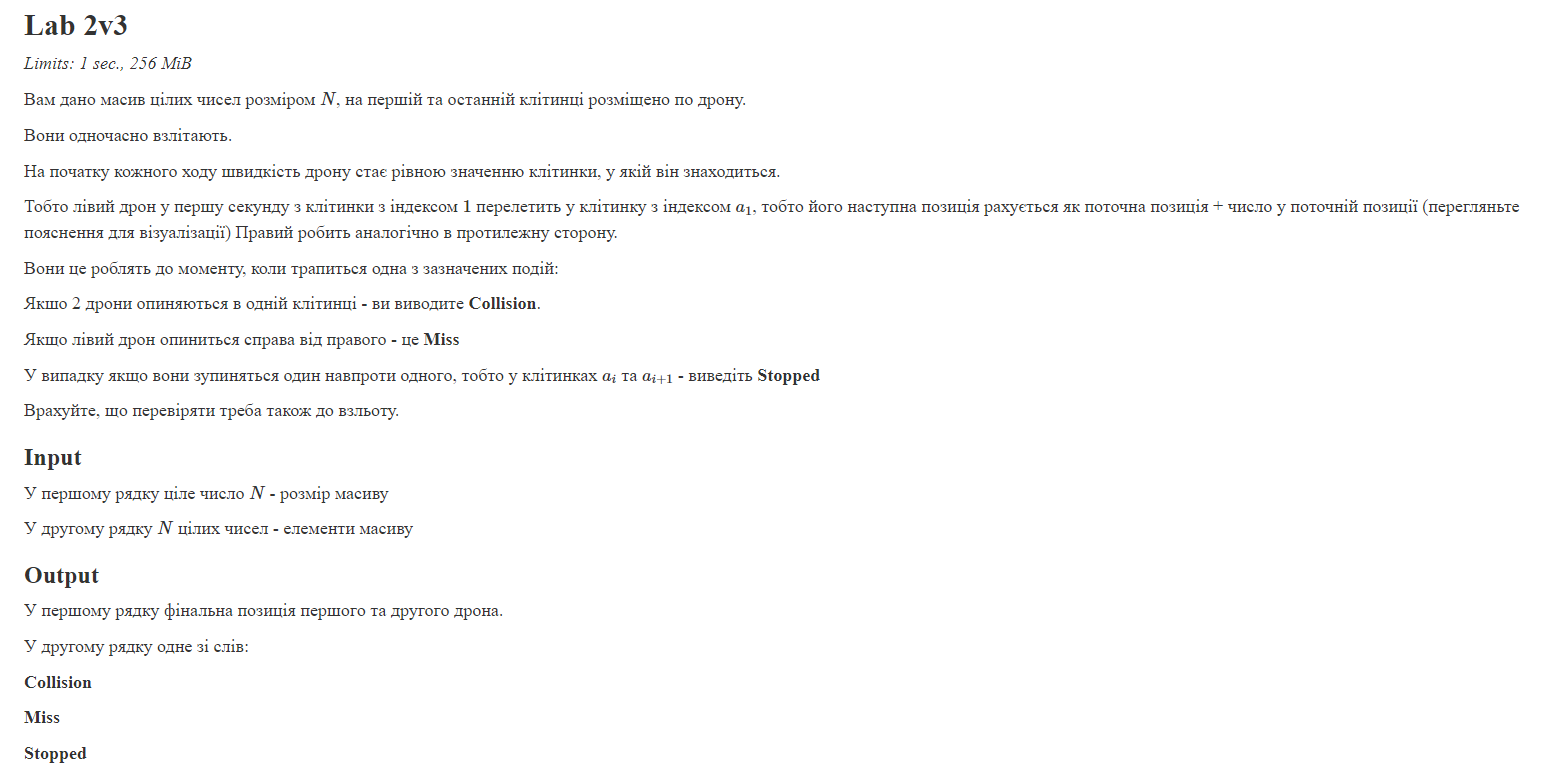
Варіант завдання: 3

* Деталі завдання:

Малюнок(2) - умова завдання 2.

Завдання №3 Лаб# Програмування: Algotester Lab 2 (Завдання 1)

Варіант завдання: 3

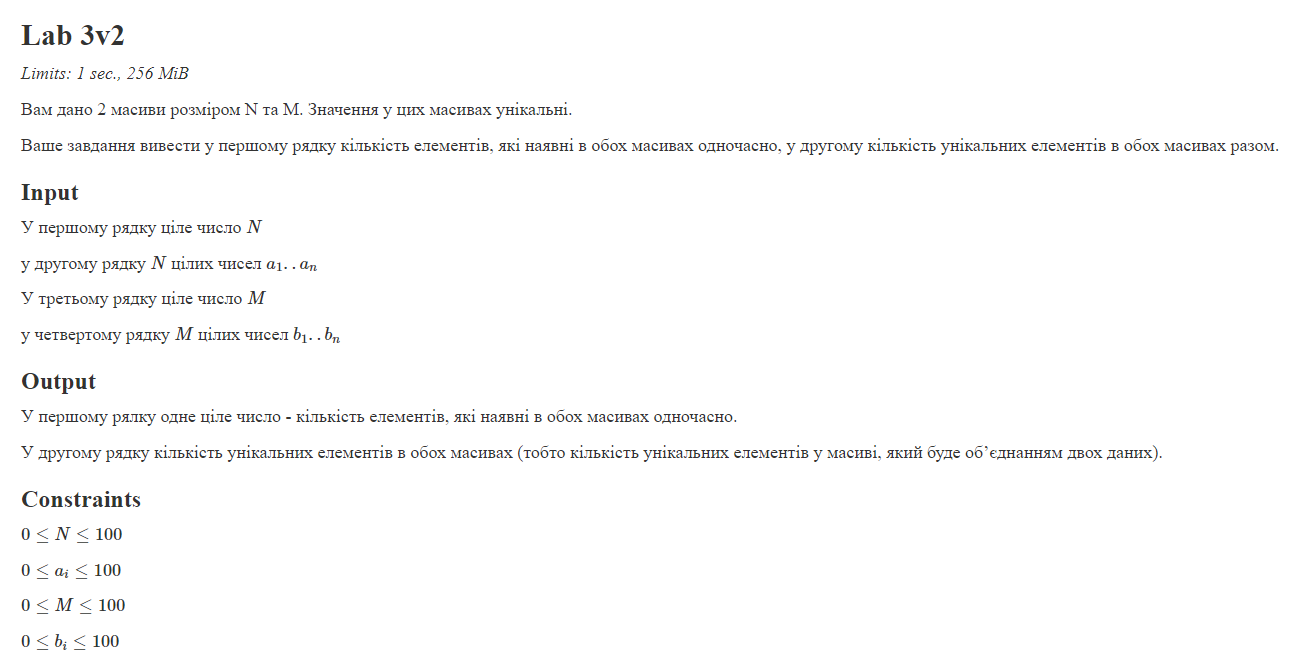
* Деталі завдання:

Малюнок(3) - умова завдання 3.

Завдання №4 Практична# Програмування: Algotester Lab 3

Варіант завдання 2

* Деталі завдання:



Малюнок(4) - умова завдання 4.

Завдання №5 Практична# Програмування: Class Practice Task

Варіант завдання (відсутній)

* Деталі завдання:

# 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»).

## Мета Задачі

Навчитися користуватися механізмами перевантаження функції та використовувати рекурсію для вирішення задач обчислення.

### Вимоги:

1. Визначення функції:
   1. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:
   1. *bool isPalindrome(const string& str, int start, int end);*
3. Перевантаження функцій:
   1. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
   2. *bool isPalindrome*(ціле число);
4. Рекурсія:
   1. 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 Практична# Програмування: Self Practice Task

Варіант завдання (відсутній)

* Деталі завдання:

Мале Бісеня любить гострити зуби. А Зла Тітонька любить до нього підходити і питатися: «Що, зуби гостриш?». Бісеняті таке не дуже подобається, тому воно придумало робити таке.У Малого Бісеняти є n зубів. Кожен зуб має коефіцієнт загостреності ai. Також існує межа загостреності k. Якщо коефіцієнт загостреності певного зуба є більшим чи рівним межі загостреності, то такий зуб вважається загостреним.Мале Бісеня хоче наступного разу, коли Зла Тітонька його щось запитає, показати їй якнайбільше загострених зубів, що розташовані поспіль.Допоможіть Малому Бісеняті дізнатися, скільки найбільше зубів воно зможе показати.

Вхідні дані: У першому рядку задані два цілих числа n та k— кількість зубів та межа загостреності відповідно.В другому рядку задано n цілих чисел ai— коефіцієнти загостреності зубів.

Вихідні дані: Єдине ціле число — відповідь на задачу.

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

Програма №1 VNS Lab 4 (Завдання 1)

* Планований час на реалізацію: 60 хвилин.

Програма №2 Лаб# Програмування: VNS Lab 5 (Завдання 1)

* Планований час на реалізацію: 90 хвилин.

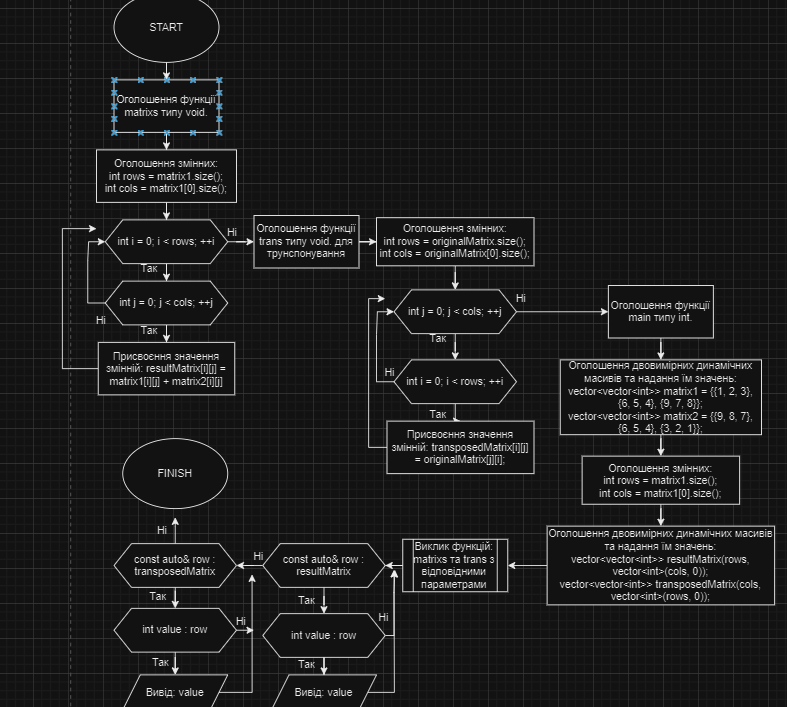
Програма №3: Лаб# Програмування: Algotester Lab 2 (Завдання 1)

* Планований час на реалізацію 60 хвилин.

Програма №4: Практична# Програмування: Algotester Lab 3

* Планований час на реалізацію 1.5 години.

Програма №5: Практична# Програмування: Class Practice Task



Малюнок(4(б)) - блок-схема до Програми №5 .

* Планований час на реалізацію 140 хвилин.

Програма №6: Практична# Програмування: Self Practice Task

* Планований час на реалізацію 30 хвилин.

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

Завдання №1: Лаб# Програмування: VNS Lab 4 (Завдання 1)

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int main(){

srand(time(0));

const int lenght = 10;

int arr[lenght];

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

arr[i] = rand() % 10;

}

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

cout << arr[i] << " ";

}

cout << endl;

int\* newArr = new int[lenght];

int newSize = 0;

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

if (arr[i] != 0) {

newArr[newSize++] = arr[i];

}

}

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

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

for (int j = newSize; j > i; --j){

newArr[j] = newArr[j - 1];

}

newArr[i + 1] = newArr[i] + 2;

++newSize;

break;

}

}

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

cout << newArr[i] << " ";

}

cout << endl;

delete[] newArr;

return 0;

}

Завдання №2: Лаб# Програмування: VNS Lab 5 (Завдання 1)

#include <iostream>

#include <vector>

using namespace std;

void matrixs(const vector<vector<int>>& matrix1, const vector<vector<int>>& matrix2, vector<vector<int>>& resultMatrix){

int rows = matrix1.size();

int cols = matrix1[0].size();

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

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

resultMatrix[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

}

void trans(const vector<vector<int>>& originalMatrix, vector<vector<int>>& transposedMatrix){

int rows = originalMatrix.size();

int cols = originalMatrix[0].size();

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

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

transposedMatrix[i][j] = originalMatrix[j][i];

}

}

}

int main(){

vector<vector<int>> matrix1 = {{1, 2, 3}, {6, 5, 4}, {9, 7, 8}};

vector<vector<int>> matrix2 = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};

int rows = matrix1.size();

int cols = matrix1[0].size();

vector<vector<int>> resultMatrix(rows, vector<int>(cols, 0));

vector<vector<int>> transposedMatrix(cols, vector<int>(rows, 0));

matrixs(matrix1, matrix2, resultMatrix);

trans(resultMatrix, transposedMatrix);

cout << "Вихідна матриця:" << endl;

for (const auto& row : resultMatrix){

for (int value : row){

cout << value << " ";

}

cout << endl;

}

cout << "\nТранспонована матриця:" << endl;

for (const auto& row : transposedMatrix){

for (int value : row){

cout << value << " ";

}

cout << endl;

}

return 0;

}

Завдання №3: Лаб# Програмування: Algotester Lab 2 (Завдання 1)

#include <iostream>

using namespace std;

int main(){

int a;

cin >> a;

int\* m = new int[a];

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

cin >> m[i];

}

int left = 1;

int right = a;

while(true){

if (right == left){

cout << left << " " << right << endl;

cout << "Collision";

break;

}

else if (right == left + 1){

cout << left << " " << right << endl;

cout << "Stopped";

break;

}

else if (right < left){

cout << left << " " << right << endl;

cout << "Miss";

break;

}

left = left + m[left - 1];

right = right - m[right - 1];

}

}

Завдання №4: Практична# Програмування: Algotester Lab 3

#include <iostream>

#include <algorithm>

using namespace std;

int main(){

int N, M;

cin >> N;

int arrayA[100];

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

cin >> arrayA[i];

}

cin >> M;

int arrayB[100];

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

cin >> arrayB[i];

}

sort(arrayA, arrayA + N);

sort(arrayB, arrayB + M);

int commonElements = 0;

int uniqueElements = 0;

int i = 0, j = 0;

while (i < N && j < M){

if (arrayA[i] < arrayB[j]){

++i;

} else if (arrayB[j] < arrayA[i]){

++j;

} else{

++commonElements;

++i;

++j;

}

}

uniqueElements = N + M - commonElements;

cout << commonElements << endl;

cout << uniqueElements << endl;

return 0;

}

Завдання №5: Практична# Програмування: Class Practice Task

#include <iostream>

#include <string>

using namespace std;

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

if (start >= end){

return true;

}

if (str[start] != str[end]){

return false;

}

return isPalindrome(str, start + 1, end - 1);

}

bool isPalindrome(int number){

string str = to\_string(number);

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

}

int main(){

string word;

cout << "Enter a word: ";

cin >> word;

if (isPalindrome(word, 0, word.length() - 1)){

cout << "palindrome." << endl;

} else{

cout << "not a palindrome." << endl;

}

int number;

cout << "Enter a number: ";

cin >> number;

if (isPalindrome(number)){

cout << "palindrome." << endl;

} else{

cout << "not a palindrome." << endl;

}

return 0;

}

Завдання №6: Практична# Програмування: Self Practice Task

#include <iostream>

using namespace std;

int main(){

int N;

long K;

cin >> N >> K;

long arr[N];

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

cin >> arr[i];

}

long counter = 0;

long maxcounter = 0;

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

if(arr[i] >= K){

counter++;

continue;

}

if(counter > maxcounter){

maxcounter = counter;

}

counter=0;

}

if(counter > maxcounter){

maxcounter = counter;

}

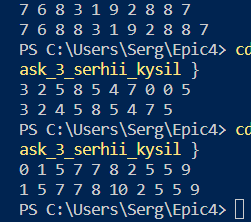
cout << maxcounter;

return 0;

}

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

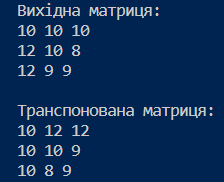
Завдання №1: Лаб# Програмування: VNS Lab 4 (Завдання 1)



Малюнок(5) - результат виконання завдання 1.

Затрачений час: 50 хв.

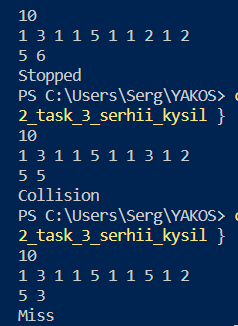
Завдання №2: Лаб# Програмування: VNS Lab 5 (Завдання 1)



Малюнок(6) - результат виконання завдання 2.

Затрачений час: 2 години.

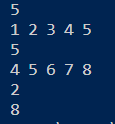
Завдання №3: Лаб# Програмування: Algotester Lab 2 (Завдання 1)



Малюнок(7) - результат виконання завдання 3.

Затрачений час: 1.5 години(or less).

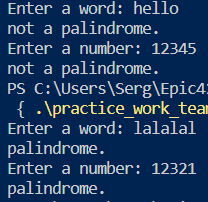
Завдання №4: Практична# Програмування: Algotester Lab 3



Малюнок(8) - результат виконання завдання 4.

Затрачений час: 2 години.

Завдання №5: Практична# Програмування: Class Practice Task



Малюнок(9) - результат виконання завдання 5.

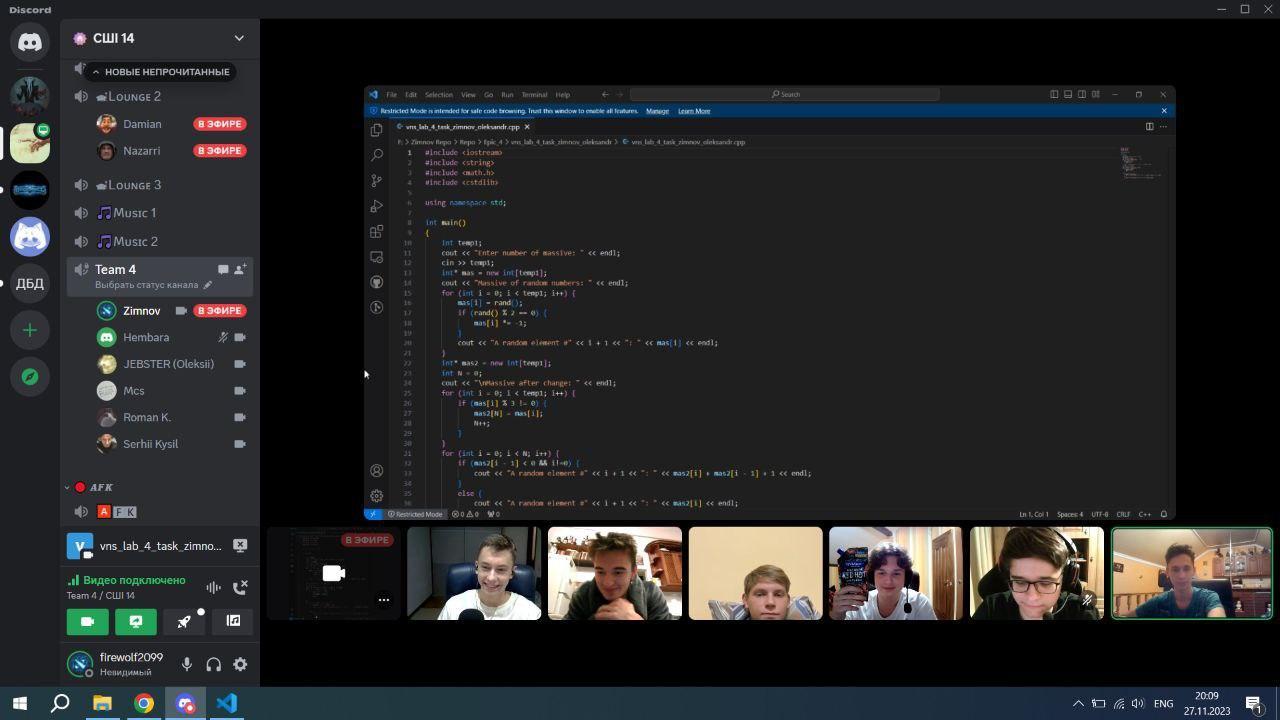
Затрачений час: 60 хвилин.

Завдання №6: Практична# Програмування: Self Practice Task



Малюнок(10) - результат виконання завдання 6.

Затрачений час: 30 хвилин.



Малюнок(11) - скрін-підтвердження активності команди.

# **Висновки:**

Отже, в ході роботи над епіком 4 я отримав знання про те, що таке масиви і які вони бувають. Були вивчені одновимірні та багатовимірні масиви, принципи їх ініціалізації та доступу до членів. Окремо було вивчено поняття поінтерів та референсів, зокрема як їх ефективно використовувати в функціях та масивах. Було розглянуто поняття адресної арифметики та доступу до елементів масиву через вказівники. Я вивчив поняття динамічних масивів,як їх ініціалізовувати та яку роль в цьому грають вказівники. Крім того, в якості додаткової інформації я вивчив std::array та std::vector як модифіковані різновиди статичних та динамічних масивів.

PR: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/719>