Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

**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-13

Степанюк Артем Костянтинович

# **Тема роботи:** Прості структури даних. Одновимірні масиви. Двовимірні Масиви. Алгоритми обробки.

**Мета роботи:** Вивчити використання простих структур даних, одновимірних та двовимірних масивів

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

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

1. Індивідуальний план опрацювання теорії:

* Тема №1: Прості структури даних.
  + Джерела Інформації
    - [Data Structures Explained for Beginners - How I Wish I was Taught - YouTube](https://www.youtube.com/watch?v=SFEROgwxicA)
  + Що опрацьовано:
    - Вивчив прості структури даних по туторіалу в ютубі
  + Статус: Ознайомлено
  + Початок опрацювання теми: 04.12.2023
  + Звершення опрацювання теми: 05.12.2023
* Тема №2: Одновимірні масиви
  + Джерела Інформації:
    - [C++ Теорія ⦁ Урок 40 ⦁ Одновимірні масиви - YouTube](https://www.youtube.com/watch?v=ULdbOaMBPYc)
  + Що опрацьовано:
    - Вивчив одновимірні масиви та їх використання по туторіалу на ютубі
  + Статус: Ознайомлено
  + Початок опрацювання теми: 04.12.2023
  + Звершення опрацювання теми: 05.12.2023
* Тема №3: Двовимірні Масиви
  + Джерела Інформації:
    - [Двовимірні масиви С++ (youtube.com)](https://www.youtube.com/watch?v=6dagAMyWTdY)
  + Що опрацьовано:
    - Ознайомився з двовимірними масивами по роліку на ютубі.
  + Статус: Ознайомлено
  + Початок опрацювання теми: 04.12.2023
  + Звершення опрацювання теми: 05.12.2023
* Тема №4: Алгоритми обробки.
  + Джерела Інформації:
    - [C++ Теорія ⦁ Урок 175 ⦁ STL ⦁ Алгоритми - YouTube](https://www.youtube.com/watch?v=m9ZHTiW-P7I)
  + Що опрацьовано:
    - Вивчив застосування алгоритмів обробки по відеоролику на ютубі
  + Статус: Ознайомлено
  + Початок опрацювання теми: 04.12.2023

Звершення опрацювання теми: 05.12.2023

* Тема №5: Сортування вибіркою
  + Джерела Інформації:
    - [C++ Теорія ⦁ Урок 51 ⦁ Сортування вибіркою - YouTube](https://www.youtube.com/watch?v=uQxG9gBROog)
  + Що опрацьовано:
    - Вивчив застосування сортування вибіркою в коді по відеоролику на ютубі
  + Статус: Ознайомлено
  + Початок опрацювання теми: 04.12.2023

Звершення опрацювання теми: 05.12.2023

# **Виконання роботи:**

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

Завдання №1 Робота з одновимірними масивами

**Деталі завдання:**

1) Реалізувати з використанням масиву двонаправлене кільце (перегляд

можливий в обидва боки, від останнього елемента можна перейти до

першого).

2) Роздрукувати отриманий масив, починаючи з К-ого елемента і до К-1 (по

кільцю вліво).

3) Знищити з кільця перший й останній елементи.

4) Роздрукувати отриманий масив, починаючи з К-ого елемента (і до К+1 по

кільцю вправо).

**Важливі деталі для врахування в імплементації програми:**

Операції над вказівниками.

присвоювання (=);

отримання значення об'єкта, на який посилається вказівник (\*);

отримання адреси самого вказівника (&).

**Завдання №2** Двовимірні масиви

**Деталі завдання:** Задано двовимірний масив. Знайти суму елементів першого стовпця без

одного останнього елемента, суму елементів другого стовпця без двох

останніх, суму елементів третього стовпця без трьох останніх і т.д. Останній

стовпець не обробляється. Серед знайдених сум знайти максимальну.

**Важливі деталі для врахування в імплементації програми:**

Використовуючи функції, розв’язати зазначене у варіанті завдання. Масив повинен передаватися у функцію як параметр.

**Завдання №3 Algotester Lab 2v1.**

**Деталі завдання:** 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:**

Вхiднi данi

У першому рядку цiле число N - кiлькiсть чисел

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

Вихiднi данi

Єдине цiле число m - мiнiмальна втома, яку можна отримати

Обмеження

1 ≤ N ≤ 105

0 ≤ ri ≤ 105

**Завдання №4 Algotester Lab 3v2**.

**Деталі завдання:** Вам дано 2 масиви розмiром N та M. Значення у цих масивах унiкальнi.

Ваше завдання вивести у першому рядку кiлькiсть елементiв, якi наявнi в обох масивах одночасно, у другому кiлькiсть унiкальних елементiв в обох масивах разом.

**Важливі деталі для врахування в імплементації програми:**

Вхiднi данi

У першому рядку цiле число N

у другому рядку N цiлих чисел a1..an

У третьому рядку цiле число M

у четвертому рядку M цiлих чисел b1..bn

Вихiднi данi

У першому рялку одне цiле число - кiлькiсть елементiв, якi наявнi в обох масивах одночасно.

У другому рядку кiлькiсть унiкальних елементiв в обох масивах (тобто кiлькiсть унiкальних

елементiв у масивi, який буде об’єднанням двох даних).

Обмеження

0 ≤ N ≤ 100

0 ≤ ai ≤ 100

0 ≤ M ≤ 100

0 ≤ bi ≤ 100

Завдання №5 Практичне завдання

**Деталі завдання:** Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.

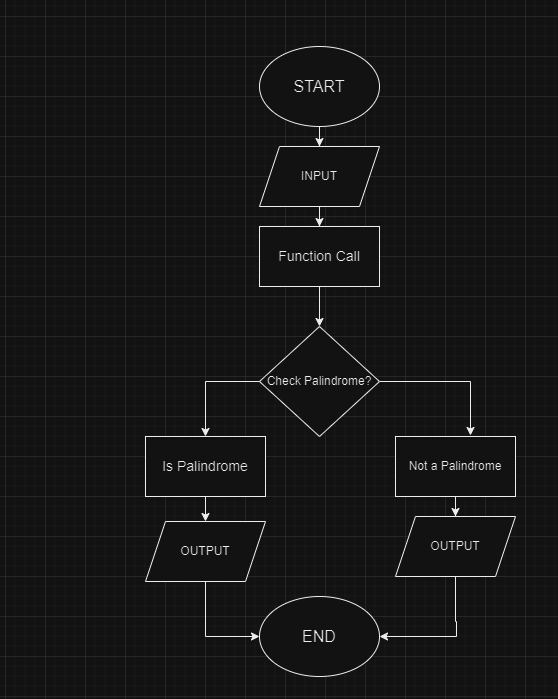
**Важливі деталі для врахування в імплементації програми:**

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.

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

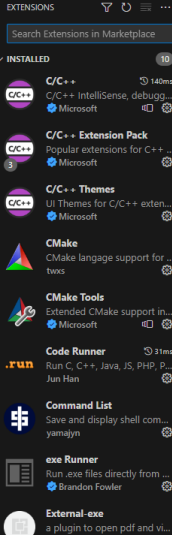
Завдання №5: Практична



*Блок-схема для практичної*

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

Завдання №1. Конфігурація середовища

. ****

*Конфігурація мого середовища*

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

**Завдання №1: Робота з одновимірними масивами**

#include <iostream>

const int ARRAY\_SIZE = 10;

class CircularArray {

private:

int mas[ARRAY\_SIZE];

int head;

public:

CircularArray() : head(0) {

for (int i = 0; i < ARRAY\_SIZE; ++i) {

mas[i] = i + 1;

}

}

int getElement(int index) const {

return mas[index];

}

void setElement(int index, int value) {

mas[index] = value;

}

void printLeftCircular(int k) const {

for (int i = k; i != (k - 1 + ARRAY\_SIZE) % ARRAY\_SIZE; i = (i - 1 + ARRAY\_SIZE) % ARRAY\_SIZE) {

std::cout << getElement(i) << " ";

}

std::cout << std::endl;

}

void removeFirstAndLast() {

head = (head + 1) % ARRAY\_SIZE;

setElement((head - 1 + ARRAY\_SIZE) % ARRAY\_SIZE, getElement(head));

head = (head + 1) % ARRAY\_SIZE;

}

void printRightCircular(int k) const {

for (int i = k; i != (k + 1 + ARRAY\_SIZE) % ARRAY\_SIZE; i = (i + 1) % ARRAY\_SIZE) {

std::cout << getElement(i) << " ";

}

std::cout << std::endl;

}

};

int main() {

CircularArray circularArray;

std::cout << "Початковий масив: ";

for (int i = 0; i < ARRAY\_SIZE; ++i) {

std::cout << circularArray.getElement(i) << " ";

}

std::cout << std::endl;

int k = 3;

std::cout << "Масив, починаючи з " << k << "-го елемента і до " << k - 1 << ": ";

circularArray.printLeftCircular(k);

circularArray.removeFirstAndLast();

std::cout << "Масив, починаючи з " << k << "-го елемента і до " << k + 1 << ": ";

circularArray.printRightCircular(k);

return 0;

}

**Завдання №2 "Двовимірні масиви"**

#include <iostream>

#include <vector>

int findColumnSum(const std::vector<std::vector<int>>& matrix, int column, int excludeLast) {

int sum = 0;

for (int i = 0; i < matrix.size() - excludeLast; ++i) {

sum += matrix[i][column];

}

return sum;

}

int findMaxColumnSum(const std::vector<std::vector<int>>& matrix) {

int maxSum = 0;

for (int i = 0; i < matrix[0].size() - 1; ++i) {

int sum = findColumnSum(matrix, i, i);

if (sum > maxSum) {

maxSum = sum;

}

}

return maxSum;

}

int main() {

std::vector<std::vector<int>> matrix = {

{1, 2, 3, 4},

{5, 6, 7, 8},

{9, 10, 11, 12},

{13, 14, 15, 16}

};

int maxSum = findMaxColumnSum(matrix);

std::cout << "Максимальна сума: " << maxSum << std::endl;

return 0;

}

**Завдання №3: Algotester Lab 2**

#include <iostream>

#include <algorithm>

#include <vector>

int main() {

int N;

std::cin >> N;

std::vector<int> r(N);

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

std::cin >> r[i];

}

if (N == 1) {

std::cout << 0 << std::endl;

return 0;

}

std::sort(r.begin(), r.end());

int minFatigue = std::min(r[N-2] - r[0], r[N-1] - r[1]);

std::cout << minFatigue << std::endl;

return 0;

}

**Завдання №4 Algotester Lab 3.**

#include <iostream>

#include <unordered\_set>

int main() {

int N;

std::cin >> N;

std::unordered\_set<int> setA;

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

int element;

std::cin >> element;

setA.insert(element);

}

int M;

std::cin >> M;

std::unordered\_set<int> setB;

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

int element;

std::cin >> element;

setB.insert(element);

}

int commonElements = 0;

for (int element : setA) {

if (setB.find(element) != setB.end()) {

commonElements++;

}

}

std::unordered\_set<int> unionSet(setA);

unionSet.insert(setB.begin(), setB.end());

int uniqueElements = unionSet.size();

std::cout << commonElements << std::endl;

std::cout << uniqueElements << std::endl;

return 0;

}

**Завдання №5 Практична.**

#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 num) {

string s = to\_string(num);

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

}

int main() {

cout << "Enter a word or number to check for palindrome: ";

string input;

cin >> input;

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

cout << "\"" << input << "\" is a palindrome." << endl;

} else {

cout << "\"" << input << "\" is not a palindrome." << endl;

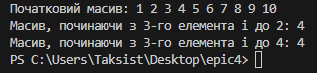
}

return 0;

}

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

Завдання №1 Робота з одновимірними масивами

**

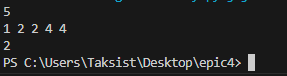
*Те, що мені видає програма 1*

Завдання №2 Двовимірні масиви

**

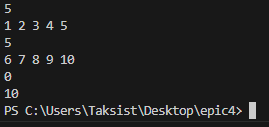
*Те, що мені видає програма 2*

Завдання №3: Algotester Lab 2



*Те, що мені видає програма 3*

Завдання №4 Algotester Lab 3.

**

*Те, що мені видає програма 4*

Завдання №5 Class Practice Work.



*Те, що мені видає програма 5*



*Те, що мені видає програма 5(2)*

**Висновки:** Я провів велику роботу з вивченням простих структур даних та використанням їх для своїх програм у с++