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(ла):**

Студент групи ШІ-13

Шабанов Микита Миколайович

# **Тема роботи:**

Робота з різними видами масивів та їх обробка

# **Мета роботи:**

Дослідження масивів, створення функцій для їх обробки, вдосконалення знань, виконання лабораторних робіт із здобутими знаннями

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Вивчення теоретичних аспектів.
* Тема №2: Лабораторна робота 1, Завдання 1, 2: VNS Lab 4.
* Тема №3: Лабораторна робота 2, Завдання 1, 2: VNS Lab 5.
* Тема №4: Лабораторна робота 3, Завдання 1: Algotester Lab 2.
* Тема №5: Лабораторна робота 4, Завдання 1: Algotester Lab 3.
* Тема №6: Практичне завдання: Class Practice Task.
* Тема №7: Практичне завдання: Self Practice Task.
* Тема №8: Розуміння вимог та задач. Створення блок-схем для завдань 3-8.
* Тема №9: Результати документації та Звіт про виходи та заходи (документи та програми на GitHub).
* Тема №10: Оцінка результатів та Випуск.Індивідуальний план опрацювання теорії:
* Тема №1: Вивчення теоретичних аспектів.
  + Джерела Інформації
    - <https://acode.com.ua/urok-81-bagatovymirni-masyvy/#toc-0>
    - <http://cpp.dp.ua/rekursyvni-funktsiyi/>
    - <https://acode.com.ua/urok-77-masyvy/>
  + Що опрацьовано:
    - Робота з масивами та вказівниками.
    - Робота з двовимірними масивами
    - З динамічними масивами
    - Розуміння як працює рандом
    - Бінарний пошук
    - Директиви препроцесора
    - for each
  + Статус: Ознайомлений

# **Виконання роботи:**

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

Завдання №1 Лабораторна робота: VNS Lab 4.

* 6 варіант
* Робота з масивом, робота з випадковою генерацію чисел.

Завдання №2 Лабораторна робота: VNS Lab 5.

* 6 варіант
* Робота з двовимірним масивом.

Завдання №3 Практичне завдання: Class Practice Task.

* Загальний варіант
* Визначити чи слово буде паліндромом.

Завдання №4 Практичне завдання: Self Practice Task.

* Algotester lab3v2
* У першому рядку одне ціле число - кількість елементів, які наявні в обох масивах одночасно.
* У другому рядку кількість унікальних елементів в обох масивах (тобто кількість унікальних елементів у масиві, який буде об’єднанням двох даних)

Завдання №5 Практичне завдання: Algotester lab 2.

* 6 варіант.
* Завдання з дронами, про роботу з елементами масиву.

Завдання №6 Практичне завдання: Algotester lab 3.

* 2 варіант.
* Перевірка двох масивів на кількість однакових та унікальних значень.

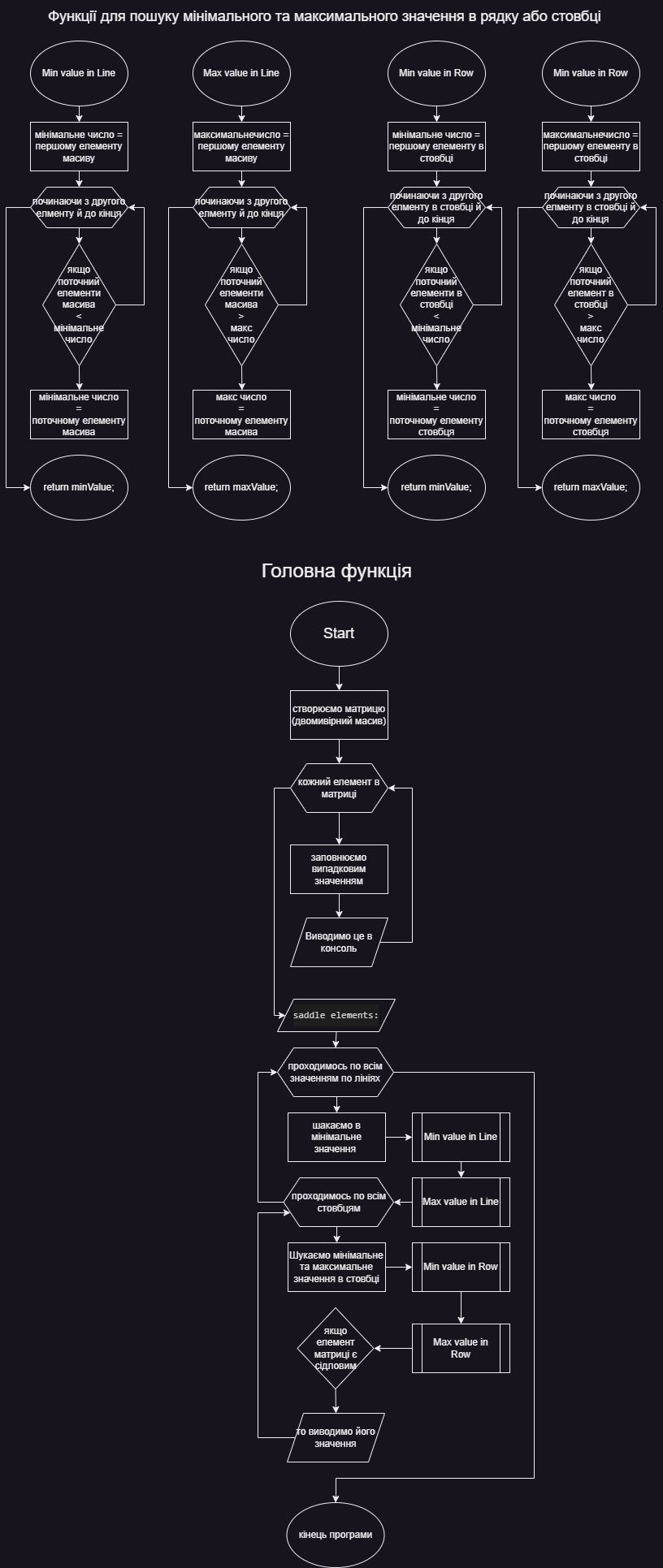
## **2. Дизайн та планована оцінка часу виконання завдань:**

* Завдання №1 Лабораторна робота: VNS Lab 4.
  + 1 година
* Завдання №2 Лабораторна робота: VNS Lab 5.
  + 3 години
* Програма №3 Class Practice Task
  + 2 години
* Програма №4 Self Practice Task.
  + Блок-схема: Рисунок 1 блок-схема Self Practice
  + 1 година
* Завдання №5 Практичне завдання: Algotester lab 2.
  + 1 година
* Завдання №6 Практичне завдання: Algotester lab 3.
  + 1 година

## **3. Блок схеми до завдань:**

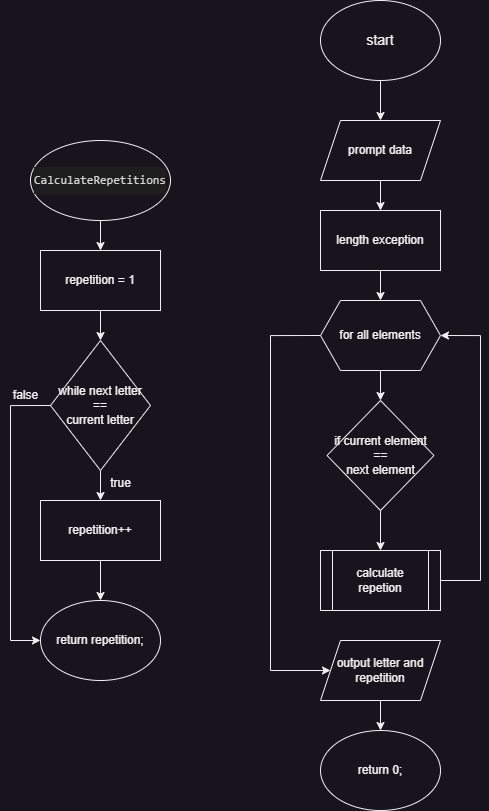
* **Програма №2 VNS Lab 5**

[*Link на файл в гугл драйв*](https://drive.google.com/file/d/1r-ch9iVEsqQoIq0mWYqvC7Ao2vXkjmuR/view?usp=sharing)

****

*Рисунок 1 VNS lab 5*

* **Програма №4 Self Practice Task**



*Рисунок 2 Self Practice Task.*

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

Завдання №1 Лабораторна робота: VNS Lab 4.

*#include <iostream>*

*#include <random>*

*#include <algorithm>*

*using namespace std;*

*void DestroyElement(int index, int arr[], int &size)*

*{*

*for (int i = index; i < size - 1; i++)*

*{*

*arr[i] = arr[i+1];*

*}*

*}*

*void AddAfterEven(int arr[], int size)*

*{*

*for (int i = size - 1; i > 0; i--)*

*{*

*if(i > 3)*

*arr[i] = arr[i - 1];*

*else*

*{*

*arr[i] = arr[i-1] + 2;*

*return;*

*}*

*}*

*}*

*int main()*

*{*

*random\_device rd;*

*mt19937 gen(rd());*

*uniform\_int\_distribution<int> distr(1, 20);*

*int arrSize = distr(gen);*

*int arr[arrSize];*

*for (int i = 0; i < arrSize; i++)*

*{*

*arr[i] = distr(gen);*

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

*}*

*cout << endl;*

*int deleteThisElement = 3;*

*DestroyElement(deleteThisElement, arr, arrSize);*

*AddAfterEven(arr, arrSize);*

*for (int i = 0; i < arrSize; i++)*

*{*

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

*}*

*return 0;*

*}*

*Код 1 VNS lab 4*

Завдання №2 Лабораторна робота: VNS Lab 5.

#include <iostream>

#include <random>

#include <algorithm>

using namespace std;

int findMinValue(int arr[], int size)

{

int minValue = arr[0];

for (int i = 1; i < size; ++i) {

if (arr[i] < minValue)

minValue = arr[i];

}

return minValue;

}

int findMaxValue(int arr[], int size)

{

int maxValue = arr[0];

for (int i = 1; i < size; ++i) {

if (arr[i] > maxValue)

maxValue = arr[i];

}

return maxValue;

}

int findMaxRow(int\*\* arr, int size, int row)

{

int mValue = arr[0][row];

// cout << "maxrow: " << mValue;

for (int i = 1; i < size; i++)

{

if (arr[i][row] > mValue)

mValue = arr[i][row];

// cout << " " << mValue;

}

return mValue;

}

int findMinRow(int\*\* arr, int size, int row)

{

int mValue = arr[0][row];

// cout << "minrow: " << mValue;

for (int i = 1; i < size; i++)

{

if (arr[i][row] < mValue)

mValue = arr[i][row];

// cout << " " << mValue;

}

return mValue;

}

int main()

{

random\_device rd;

mt19937 gen(rd());

uniform\_int\_distribution<int> distr2(0, 100);

int mLines = 3, mRows = 4;//distr(gen);

int\*\* matrix = new int\* [mLines];

matrix[0] = new int[mRows]{7,1,5,3};

matrix[1] = new int[mRows]{3,2,6,4};

matrix[2] = new int[mRows]{5,2,8,6};

for (int i = 0; i < mLines; i++)

{

matrix[i] = new int[mRows];

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

{

matrix[i][j] = distr2(gen);

cout << matrix[i][j] << " ";

}

cout << endl;

}

cout << endl;

// cout << matrix[3][1];

// cout << matrix[1][0] << " " << matrix[2][0] << endl;

cout << "saddle elements: ";

for (int i = 0; i < mLines; i++)

{

int minLine = findMinValue(matrix[i], mLines);

int maxLine = findMaxValue(matrix[i], mLines);

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

{

int minRow = findMinRow((int\*\*)matrix, mLines, j);

int maxRow = findMaxRow((int\*\*)matrix, mLines, j);

// cout << "min/max: " << minRow << " " << maxRow << endl;

if (matrix[i][j] == minLine && matrix[i][j] == maxRow

|| matrix[i][j] == maxLine && matrix[i][j] == minRow)

{

cout << matrix[i][j] << " ";

}

}

// cout << endl << "all valules: " << minLine << ", " << maxLine << ", " << minRow << ", " << maxRow << endl;

}

return 0;

}

*Код 2 VNS lab 5*

Завдання №3 Class Practice Task

#include <iostream>

#include <string>

//finding a palindrome using recursion

bool isPalindrome(const std::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);

}

//useless overload

bool isPalindrome(int num)

{

std::string str = std::to\_string(num);

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

}

int main()

{

//input

std::string input;

std::cout << "Enter a word or number to check if it's a palindrome:" << std::endl;

std::cin >> input;

//using magic on the next line

bool result = isPalindrome(input, 0, input.size() - 1);

//output

std::cout << std::endl << input;

result ? std::cout << " is a palindrome." : std::cout << " is not a palindrome.";

return 0;

}

*Код 3 Class Practice Task*

Завдання №4 Self practice

*#include <iostream>*

*#include <vector>*

*#include <algorithm>*

*using namespace std;*

*bool CheckIfNumberOccurs(vector<int> arr, int num)*

*{*

*for (int i = 0; i < arr.size(); i++)*

*{*

*if(arr[i] == num)*

*return true;*

*}*

*return false;*

*}*

*vector<int> sortedNumbers(vector<int> numbers)*

*{*

*vector<int> noRemainder, rem1, rem2;*

*for (int num : numbers)*

*{*

*if (num % 3 == 0)*

*noRemainder.push\_back(num);*

*else if (num % 3 == 1)*

*rem1.push\_back(num);*

*else*

*rem2.push\_back(num);*

*}*

*sort(noRemainder.begin(), noRemainder.end());*

*sort(rem2.begin(), rem2.end());*

*sort(rem1.begin(), rem1.end(), greater<int>());*

*vector<int> temp, result;*

*temp.insert(temp.end(), noRemainder.begin(), noRemainder.end());*

*temp.insert(temp.end(), rem1.begin(), rem1.end());*

*temp.insert(temp.end(), rem2.begin(), rem2.end());*

*for(int i = 0; i < temp.size(); i++)*

*{*

*if(!CheckIfNumberOccurs(result, temp[i]))*

*result.push\_back(temp[i]);*

*else continue;*

*}*

*return result;*

*}*

*int main()*

*{*

*//input*

*int n;*

*cin >> n;*

*vector<int> numbers(n);*

*for (int i=0; i<n; ++i)*

*cin >> numbers[i];*

*vector<int> result = sortedNumbers(numbers);*

*cout << result.size() << endl;*

*for (int i=0; i < result.size(); ++i)*

*cout << result[i] << " ";*

*return 0;*

*}*

*Код 4 Self practice*

## Завдання №5 Algotester lab 2

#include <iostream>

#define CHECK\_FOR\_REMOVED\_OBJECT nums[i] != del[0] && nums[i] != del[1] && nums[i] != del[2]

int main()

{

//prompting

//розмір масиву

int n;

std::cin >> n;

//самі ці числа

int nums[n];

for(int i = 0; i < n; i++)

std::cin >> nums[i];

//ті, які треба видалити

int del[3];

for(int i = 0; i < 3; i++)

std::cin >> del[i];

//main logic

//дізнаємось скільки чисел буде у новому масиві

int newSize = 0, remaining[n];

for(int i = 0, j = 0; i < n; i++)

{

if(CHECK\_FOR\_REMOVED\_OBJECT)

{ //якби було >3 числа, які треба перевірити, то можна було б циклом зробити

newSize++;

remaining[j++] = nums[i];

}

}

//одразу й виводимо

if(newSize <= 0)

{

std::cout << 0;

return 0;

}

std::cout << newSize - 1 << std::endl;

//створення й вивід нового масиву

int newNums[newSize - 1];

for(int i = 0, j = 0; i < newSize - 1; i++)

{

newNums[i] = remaining[i] + remaining[i+1];

std::cout << newNums[i] << " ";

}

return 0;

}

*Код 5 Algotester lab 2*

Завдання №6 Algotester lab 3

#include <iostream>

using namespace std;

bool SearachBinary(int arr[], int size, int key)

{

int mid, leftIndex = 0, rightIndex = size - 1;

while(rightIndex >= leftIndex)

{

mid = (leftIndex + rightIndex)/2;

if(key < arr[mid])

rightIndex = mid - 1;

else if(key > arr[mid])

leftIndex = mid + 1;

else

return true;

}

return false;

}

int main()

{

//prompting

int n, m;

cin >> n;

int nums1 [n];

for(int i = 0; i < n; i++)

cin >> nums1[i];

cin >> m;

int nums2[m];

for(int i = 0; i < m; i++)

cin >> nums2[i];

//serching elements in array

int result = 0, l = n + m;

for (int i = 0; i < n; i++)

{

int key = nums1[i];

if(SearachBinary(nums2, m, key))

{

result++;

l--;

}

}

cout << result << endl << l;

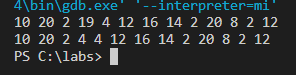
return 0;

}

*Код 5 Algotester lab 3*

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

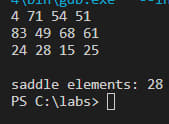
Завдання №1 VNS Lab 4



*Тест 1 VNS Lab 4*

Час затрачений на виконання завдання: 1 година

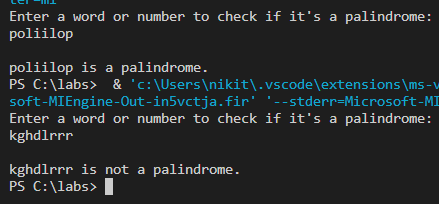
Завдання №2 VNS lab 5



*Тест 2 VNS lab 5*

Час затрачений на виконання завдання: 1 година

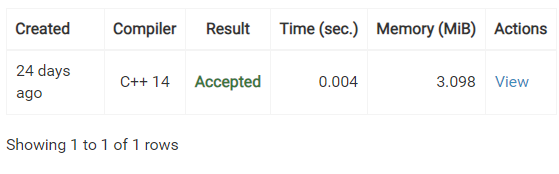
Завдання №3 Class Practice Task



*Тест 3 Class Practice Task*

Час затрачений на виконання завдання: 1 година 20 хвилин

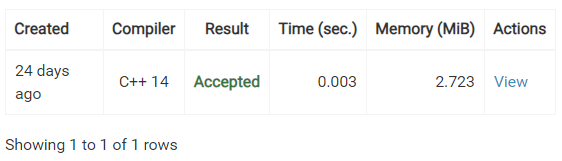
Завдання №4 Self Practice Task.



*Тест 4 Self Practice Task*

Час затрачений на виконання завдання : 1 година 20 хвилин

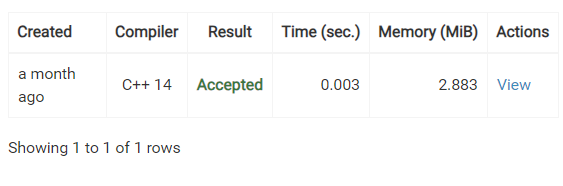
Завдання №5 Практичне завдання: Algotester lab 2.



*Тест 5 Algotester lab 2*

Час затрачений на виконання завдання: 3 години

Завдання №6 Практичне завдання: Algotester lab 3.



*Тест 5 Algotester lab 3*

Час затрачений на виконання завдання: 1 година

# **Висновки:**

Підчас виконання цієї роботи були отримані та удосконалені знання про роботу з масивами, вказівниками, функціями та їх створення. Удосконалено навички створення Flow Charts.