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-12

Пастухов Олександр Сергійович

# **Тема роботи:**

Ознайомитися зі структурами даних,такими як одновимірними та двовимірними масиви та з алгоритмами обробки масивів.

# **Мета роботи:**

Вивчити що таке структури даних. Також з одновимірними та двовимірними масивами і використати їх на практиці і вивчити алгоритми обробки масивів даних.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

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

1. Індивідуальний план опрацювання теорії:

* Тема №1: Структури даних.
  + Джерела Інформації
    - Стаття. <https://acode.com.ua/urok-64-struktury/>
  + Що опрацьовано:
  + Ініціалізація та вкладені структури.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 17.12.2023
  + Звершення опрацювання теми: 19.12.2023
* Тема №2: Одновимірні масиви.
  + Джерела Інформації
    - Стаття. <https://acode.com.ua/urok-77-masyvy/>
  + Що опрацьовано:
    - Опрацьовано що таке масив, з чого він складається.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 18.12.2023
  + Звершення опрацювання теми: 19.12.2023
* Тема №3: Двовимірні Масиви.
  + Джерела Інформації
    - Стаття. <https://acode.com.ua/urok-81-bagatovymirni-masyvy/>
  + Що опрацьовано:
    - Опрацьовано що таке двовимірні масиви, методи заповнювання та звернення до елементів.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 18.12.2023
  + Звершення опрацювання теми: 18.12.2023
* Тема №4 Алгоритми обробки масивів.
  + Джерела Інформації
    - Статті. <http://cpp.dp.ua/obrobka-odnovymirnyh-masyviv/>  
      <https://cherto4ka.xyz/2020/02/08/%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D1%96-%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D1%97-%D0%BE%D0%B1%D1%80%D0%BE%D0%B1%D0%BA%D0%B8-%D0%B4%D0%B2%D0%BE%D0%B2%D0%B8%D0%BC%D1%96%D1%80%D0%BD%D0%B8%D1%85/>
  + Що опрацьовано:
    - Опрацьовано методи одновимірних та двовимірних масивів на прикладах.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 18.19.2023
  + Звершення опрацювання теми: 19.12.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 4 - Task 1

* Варіант завдання-19
* Деталі завдання - Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел. Роздрукувати отриманий масив. Знищити елемент із заданим номером. Додати після першого парного елемента масиву елемент зі значенням M[ I-1]+2. Роздрукувати отриманий масив.
* Важливі деталі для врахування в імплементації програми – використати генератор випадкових чисел та одновимірний масив.

Завдання №2 VNS Lab 5 - Task 1

* Варіант завдання-19
* Деталі завдання - Елемент матриці є сідловою точкою, якщо він є найменшим у своєму рядку й найбільшим у своєму стовпці (або навпаки: найбільшим у своєму рядку й найменшим у своєму стовпці). Для заданої матриці визначити всі сідлові точки.
* Важливі деталі для врахування в імплементації програми – використання двовимірного масиву

Завдання № 3Algotester Lab 2

* Варіант завдання - 3
* Деталі завдання <https://algotester.com/uk/ContestProblem/DisplayWithEditor/134637>
* Важливі деталі для врахування в імплементації програми – використання одновимірного масиву

Завдання №4 Algotester Lab 3

* Варіант завдання - 3
* Деталі завдання <https://algotester.com/uk/ContestProblem/DisplayWithEditor/134639>
* Важливі деталі для врахування в імплементації програми – використання циклу for для перевірки

Завдання №5 Class Practice Work

* Варіант завдання - немає
* Деталі завдання - реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.
* Важливі деталі для врахування в імплементації програми – використання механізмів переванження функції та використання рекурсії

Завдання №6 Self Practice Work

* Варіант завдання - немає
* Деталі завдання - <https://algotester.com/uk/ArchiveProblem/Display/20040>

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

Програма №2 Algotester 2

* Блок-схема
* Планований час на реалізацію – 1 година

Програма №1 VNS Lab 4 - Task 1

* Планований час на реалізацію -1 година

Програма №2 VNS Lab 5 - Task 1

* Планований час на реалізацію-1,5 годин

Програма №4 Algotester Lab 3

* Планований час на реалізацію-1,5 годин

Програма №5 Class Practice Work

* Планований час на реалізацію-20 хвилин

Програма №6 Self Practice Work

* Планований час на реалізацію-30 хвилин

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

Додаткової конфігурації середовища не потрібно

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

Завдання №1 VNS Lab 4 - Task 1

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1137>

#include<iostream>

void arrayOutput(int arr[],int start,int end){

    std::cout << "Масив від " << start << " до " << end << ": ";

    for(int i = start;i <= end;i++)

        std::cout << arr[i] << " ";

    std::cout << std::endl;

}

int main(){

using namespace std;

    int size = 10;

int StartArr[size];

    for(int i = 0; i < size;i++){

        StartArr[i] = i + 1;

        cout << StartArr[i] << " ";

    }

    cout << endl;

    int k = 3;

arrayOutput(StartArr,k,k+1);

    int NewSize = size+2;

    int NewArray[NewSize];                           //у методичці не було векторів, тому я не знаю як додати перший елемент в існуючий масив

    NewArray[0] = NewSize - 1;

    NewArray[NewSize-1] = NewSize;

    for(int i = 1;i < NewSize - 1;i++)

        NewArray[i] = StartArr[i-1];

    int distance = 1;

    for(int i = 0;i < NewSize;i++)

        if(i % 2 != 0){

            NewArray[i-distance] = NewArray[i];

            distance +=1;

        }

arrayOutput(NewArray,k,k+1);

    return 0;

}

Завдання №2 VNS Lab 5 - Task 1

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1137>

#include <iostream>

#include <algorithm>

void printMatrix(int\*\* matrix, int N){

    for(int i = 0; i < N;i++){

        for(int j = 0;j < N;j++){

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

        }

        std::cout << std::endl;

    }

}

int\*\* createMatrix(std::string inputString,int N){

    if (inputString.length() != N \* N){

        std::cout << "Некоректна довжина рядка." << std::endl;

        return nullptr;

}

    int\*\* matrix = new int\*[N];

    for(int i = 0;i < N;i++)

        matrix[i] = new int[N];

    for(int i = 0;i < N;i++){

        std::string SubString = inputString.substr(i \* N,N);

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

        for (int j = 0;j < N;j++) {

            matrix[j][i] = SubString[j] - '0';

        }

}

    return matrix;

}

void freeMatrixMemory(int\*\* matrix, int N) {

    for (int i = 0;i < N;i++) {

        delete[] matrix[i];

    }

    delete[] matrix;

}

int main() {

    using namespace std;

int N;

    cout << "Введіть N: ";

cin >> N;

    string inputString;

    cout << "Введіть рядок довжиною у " << N \* N << " цифр: ";

cin >> inputString;

    int\*\* resultMatrix = createMatrix(inputString, N);

    if (resultMatrix != nullptr) {

        cout << "Двовимірний масив N x N з цифрами в порядку зростання в першому стовпці:" << endl;

        printMatrix(resultMatrix, N);

}

    return 0;

}

Завдання №3 Algotester 2

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1137>

#include<iostream>

#include<vector>

int main(){

using namespace std;

    int n;

//    cout << "Enter size of your array: " << endl;

    cin >> n;

    vector<int> arr(n);

    for (int i = 0; i < n; i++) {

        cin >> arr[i];

}

    int FirstDronePos = 1;

int SecondDronePos = n;

while(true){

        if(FirstDronePos == SecondDronePos){

            cout << FirstDronePos << " " << SecondDronePos << endl;

            cout << "Collision" << endl;

            break;

        }

        else if(FirstDronePos > SecondDronePos){

            cout << FirstDronePos << " " << SecondDronePos << endl;

            cout << "Miss" << endl;

            break;

        }

        else if(FirstDronePos + 1 == SecondDronePos){

            cout << FirstDronePos << " " << SecondDronePos << endl;

            cout << "Stopped" << endl;

            break;

        }

        FirstDronePos += arr[FirstDronePos-1];

        SecondDronePos -= arr[SecondDronePos-1];

    }

    return 0;

}

Завдання №4 Algotester 3

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1137>

#include<iostream>

#include<string>

int main(){

    using namespace std;

    string StartString,FinalString;

cin >> StartString;

int counter = 1;

    for(int i = 0;i < StartString.size();i++){

        if(StartString[i] == StartString[i+1])

            counter += 1;

        else{

            FinalString += StartString[i] + (counter > 1 ? to\_string(counter) : "");

            counter = 1;

        }

    }

    cout << FinalString << endl;

    return 0;

}

Завдання №5 Class Practice Work

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1137>

#include<iostream>

#include<string>

bool IsPalindrome(std::string word,int LeftPos,int RightPos){

    if(LeftPos >= RightPos)

        return true;

    if(word[LeftPos] == word[RightPos])

        return IsPalindrome(word,LeftPos += 1,RightPos -= 1);

    else

        return false;

}

int main(){

using namespace std;

    string WordToCheck;

    cout << "Enter word to detect palindrome: " << endl;

cin >> WordToCheck;

IsPalindrome(WordToCheck,0,WordToCheck.size() - 1);

    return 0;

}

Завдання №6 Self Practice Work

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1137>

#include<iostream>

#include<vector>

#include <algorithm>

void countElements(const std::vector<int> array1,const std::vector<int> array2){

int commonCount = 0;

std::vector<int> uniqueElements;

    for(int element : array1){

        auto it = std::find(array2.begin(),array2.end(),element);

        if(it != array2.end())

            commonCount++;

        if(std::find(uniqueElements.begin(),uniqueElements.end(),element) == uniqueElements.end())

            uniqueElements.push\_back(element);

}

    for (int element : array2)

        if (std::find(uniqueElements.begin(),uniqueElements.end(),element) == uniqueElements.end())

            uniqueElements.push\_back(element);

    std::cout << commonCount << std::endl;

    std::cout << uniqueElements.size() << std::endl;

}

int main(){

    using namespace std;

    int n;

    cin >> n;

    vector<int> array1;

    for (int i = 0;i < n;i++){

        int element;

        cin >> element;

        array1.push\_back(element);

}

    int m;

    cin >> m;

    vector<int> array2;

    for (int i = 0;i < m;i++) {

        int element;

        cin >> element;

        array2.push\_back(element);

    }

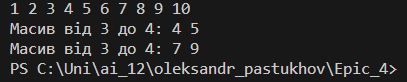
countElements(array1, array2);

    return 0;

}

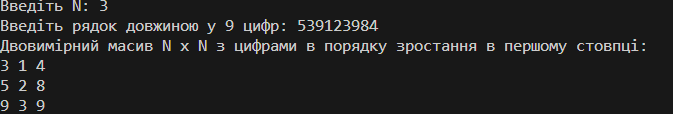
## **5. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання № VNS Lab 4 - Task 1



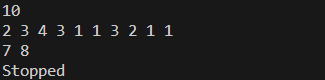
Час затрачений на виконання завдання -30 хвилин

Завдання № VNS Lab 5 - Task 1



Час затрачений на виконання завдання -2 години

Завдання № Algotester 2



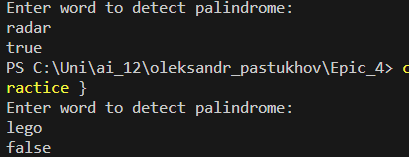
Час затрачений на виконання завдання -2 години

Завдання № Algotester 3



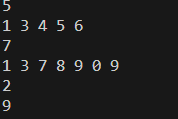
Час затрачений на виконання завдання -30 хвилин

Завдання № Class Practice Work



Час затрачений на виконання завдання -15 хвилин

Завдання № Self Practice Work



Час затрачений на виконання завдання -50 хвилин

# **Висновки:**

Я вивчив що таке масиви(одновимірні та двовимірні) по теорії та вже на практиці використав їх під час лаборатораторних та практичних робіт. Також ознайомився з деякими методами модуля vector.