Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 4 , 5**

***з дисципліни:*** «Мови та парадигми програмування»

***з розділу***: “Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.”

***Виконaла:***

студентка групи ШІ-11

Педченко Юлія Сергіївна

# **Тема роботи:**

# Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.

# **Мета роботи:**

Вивчити вищеперелічені теми та навчитися застосовувати знання з них на практиці.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Одновимірні масиви. Двовимірні масиви. Динамічні масиви.
* Тема №2: Вказівники та Посилання.
* Тема №3: Структури даних. Вкладені структури.
* Тема №4: Алгоритми обробки та робота з масивами та структурами.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Одновимірні масиви. Двовимірні масиви. Динамічні масиви.
  + Джерела Інформації
    - <https://acode.com.ua/urok-77-masyvy/>
    - <http://cpp.dp.ua/vykorystannya-masyviv/>
    - <https://acode.com.ua/urok-81-bagatovymirni-masyvy/>
    - <https://www.bestprog.net/ru/2017/03/04/%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D1%87%D0%B0%D1%81%D1%82%D1%8C-2-%D0%B4%D0%B2%D1%83%D0%BC%D0%B5%D1%80%D0%BD%D1%8B%D0%B5-%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2%D1%8B-%D0%BC%D0%B0%D1%81/>
    - Практичні та Лекції
  + Що опрацьовано:
    - Принцип роботи з одновимірним і двовимірним масивом в С++
    - Принцип роботи динамічного масиву (vector)
  + Статус: Ознайомлена
  + Початок опрацювання теми: 05.12.2023
  + Звершення опрацювання теми: 09.12.2023
* Тема №2: Вказівники та Посилання.
  + Джерела Інформації:
    - Практичні та Лекції
    - <https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85>
  + Що опрацьовано:
    - Що таке вказівники та посилання
  + Статус: Ознайомлена
  + Початок опрацювання теми: 05.12.2023
  + Звершення опрацювання теми: 09.12.2023
* Тема №3: Структури даних. Вкладені структури.
  + Джерела Інформації:

▪ Практичні та Лекції

* + - <https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85>
  + Що опрацьовано:
    - Що таке структури даних
  + Статус: Ознайомлена
  + Початок опрацювання теми: 05.12.2023
  + Звершення опрацювання теми: 09.12.2023
* Тема №4: Алгоритми обробки та робота з масивами та структурами.
  + Джерела Інформації:

▪ Практичні та Лекції

* + Що опрацьовано:
    - Алгоритми сортування масивів, та їх написання в C++
  + Статус: Ознайомлена
  + Початок опрацювання теми: 05.12.2023
  + Звершення опрацювання теми: 09.12.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 4

* Варіант 1
* 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор
* випадкових чисел.
* 2) Роздрукувати отриманий масив.
* 3) Знищити елемент із номером К.
* 4) Додати після кожного парного елемента масиву елемент зі значенням 0.
* 5) Роздрукувати отриманий масив.

В результаті цієї дії, яку мінімальну втому ви можете отримати в кінці дороги?

Завдання №2 VNS Lab 5

* Варіант 1
* У двовимірному масиві записані слова, що представляють собою
* послідовність цифр, що завершуються 0. Необхідно роздрукувати слова через
* кому, взявши надрукований рядок у дужки. Довжина друкованого рядка 60
* символів. Добування слова оформити у вигляді функції.

Завдання №3 Algotester Lab 2

* Варіант 1
* У вас є дорога, яка виглядає як N чисел. Після того як ви по ній пройдете - вашу втому можна визначити як різницю максимального та мінімального елементу. Ви хочете мінімізувати втому, але все що ви можете зробити - викинути одне число з дороги, тобто забрати його з масиву.

Завдання №4 Algotester Lab 3

* Варіант 2
* Вам дано 2 масиви розміром N та M. Значення у цих масивах унікальні. Ваше завдання вивести у першому рядку кількість елементів, які наявні в обох масивах одночасно, у другому кількість унікальних елементів в обох масивах разом.

Завдання №5 Class Practice Task

* 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»).

Завдання №6 Self Practice Task

Вам дано масив цілих чисел розміром N, на першій та останній клітинці розміщено по дрону. Вони одночасно взлітають. На початку кожного ходу швидкість дрону стає рівною значенню клітинки, у якій він знаходиться. Тобто лівий дрон у першу секунду з клітинки з індексом 1 перелетить у клітинку з індексом a1, тобто його наступна позиція рахується як поточна позиція + число у поточній позиції (перегляньте пояснення для візуалізації) Правий робить аналогічно в протилежну сторону.

Вони це роблять до моменту, коли трапиться одна з зазначених подій:

Якшо 2 дрони опиняються в одній клітинці - ви виводите Collision.

Якщо лівий дрон опиниться справа від правого - це Miss

У випадку якщо вони зупиняться один навпроти одного, тобто у клітинках

ai та ai+1 - виведіть Stopped

Врахуйте, що перевіряти треба також до взльоту.

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

Завдання №1 VNS Lab 4

Плановий час на реалізацію: 30 хвилин

Завдання №2 VNS Lab 5

Плановий час на реалізацію: 15 хвилин

Завдання №3 Algotester Lab 2

Плановий час на реалізацію: 30 хвилин

Завдання №4 Algotester Lab 3

Плановий час на реалізацію: 20 хвилин

Завдання №5 Class Practice Task

Плановий час на реалізацію: 1 година

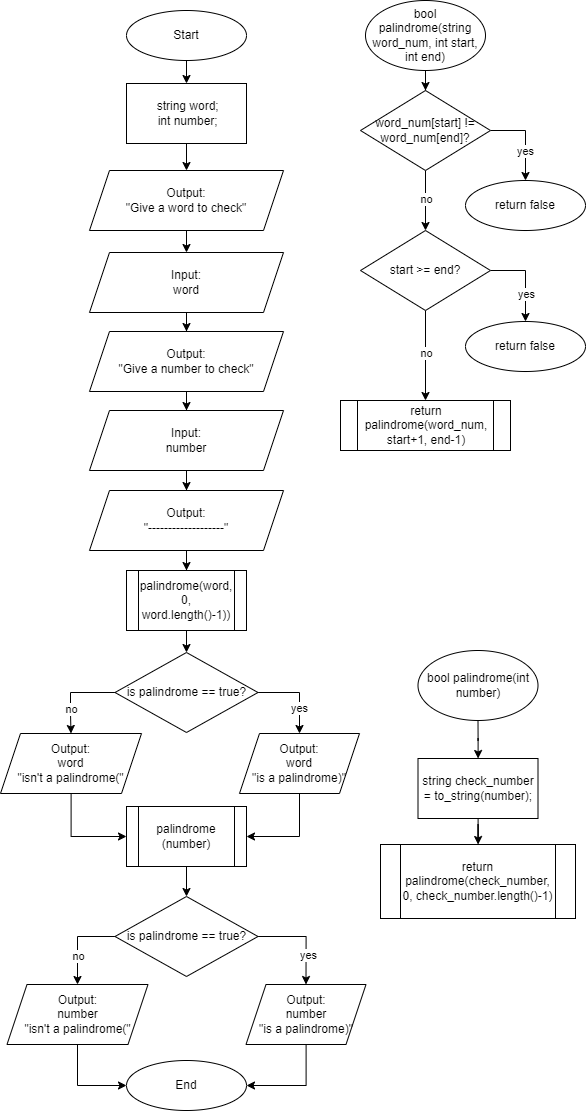


Рисунок 1. Блок-схема до завдання 5

Завдання №6 Self Practice Task

Плановий час на реалізацію: 30 хвилин

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

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/868>

## Завдання №1 VNS Lab 4

#include <iostream>

#include <cstdlib>

using *namespace* std;

*int* main()

{

*int* min = 3;

*int* max = 10;

    srand(time(NULL));

*int* num = min + rand() % (max - min + 1);

    // рандомізує кількість елементів в масиві. Оскільки функція rand() виводить дуже великі числа, я знайшла

    // формулу, яка встановить мінімальне і максимальне число, яке може вивести рандомайзер

*int* a = num;

*int* mass[a];

    min = 1;

    max = 50;

    for (*int* i = 0; i < a; i++)

    {

        mass[i] = min + rand() % (max - min + 1);

        cout << mass[i] << "  ";

    }

    cout << endl;

    // все те саме, що і зверху, лиш межі трохи інакші

*int* k;       //номер елемента, який треба видалити

    cout << "Який елемент треба видалити?" << endl;

    cin >> k;

    if (k > a || k < 1)

    {

        cout << "Елементу з таким номером в цьому масиві немає";

        return 0;

    }

*int* mass2[a - 1]; // вводимо новий масив з кількістю елементів а-1, адже один елемент ми видалимо

    k = k - 1;

    for (*int* i = 0; i < a; i++)

    {

        if (i < k)

        {

            mass2[i] = mass[i];

            cout << mass2[i] << "  ";

        }

        else if (i > k)

        {

            mass2[i - 1] = mass[i];

            cout << mass2[i - 1] << "  ";

        }  // до того, як цикл дійде до елементу k, ми прирівнюємо елементи нового масиву до відповідних в

           // старому. Елемент k ми пропускаємо, а після нього прирівнюємо елементи нового масиву до попереднього

           // елементу старого

    }

    cout << endl;

*int* m;

    if ((a-1)%2==0) m = (a-1)\*1.5;  //робимо перевірку. Якщо кількість елементів 2 масиву парна, то в фінальному

    else m = 1 + (a-2)\*1.5;         //масиві кількість елементів буде більше за кількість елементів 2 на половину,

*int* final\_mass[m];              //тобто збільшена в 1.5 рази. Якщо кількість елементів непарна, то після

                                    //елементу 0 ставитись не буде, тож ми віднімаємо від кількості 1, множимо

                                    //на 1.5 і додаємо 1.

    for (*int* j = 1; j <= m; j++) {          //для цього циклу мені легше було позначити за 1, адже далі я

*int* r = j/3;                        //я використовую формули. j не завжди ділиться націло на 3, але

        if (j%3==0) final\_mass[j-1]=0;      //тип даних int записує лиш цілочисельну частину, а мені це підходить.

        else final\_mass[j-1]=mass2[j-1-r];    //оскільки я позначила j за 1, то маю писати усюди в циклі j-1

        cout << final\_mass[j-1] << "  ";

    }

    return 0;

}

## Завдання №2 VNS Lab 5

#include <iostream>

using *namespace* std;

*void* word(*int* *mass*[3][20]) {

    for (*int* i=0; i<3; i++) {

        cout << "(";

        for (*int* j=0; j<20; j++)

        if (*mass*[i][j]==0) cout << ",";

        else cout << *mass*[i][j];

        cout << ")";

    }

}

*int* main() {

    // for (int i = 0; i < 3; i++) {

    //     for (int j = 0; j < 20; j++) {

    //         cin >> mass[i][j];

    //     }

    //     cout << endl;

    // }                                   - ввід вручну, але то довго

*int* mass[3][20] = { {1,2,3,0,2,3,4,0,2,3,0,3,4,5,0,4,0,2,3,8},

                    {2,3,4,4,5,0,2,3,4,5,6,7,0,1,0,2,3,4,5,6},

                    {2,3,4,4,5,5,6,7,7,6,7,0,4,5,0,5,6,7,7,6}  };

word(mass);

    return 0;

}

## Завдання №3 Algotester Lab 2

#include <iostream>

using *namespace* std;

*void* sort(*int* *r*[], *int* *N*) {                     //функція сортування елементів масиву від меншого до більшого,

    for (*int* i = 0; i < (*N* - 1); i++)           //щоб можна було в подальшому застосувати формулу

        for (*int* j = 0; j < (*N* - i - 1); j++)

            if (*r*[j] > *r*[j + 1])

            swap(*r*[j], *r*[j + 1]);

}

*int* main() {

*int* N;

*int* m;

cin >> N;

if (N<1) return 0;

*int* r[N];

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

    cin >> r[i];

    if (r[i]<0) return 0;

}

if (N==1) {cout << 0; return 0;}

sort(r, N);

*int* v1 = r[N-2]-r[0];       //перевіряємо, яка з різниць менша: різниця останнього і другого, чи

*int* v2 = r[N-1]-r[1];       //передостаннього і першого

if (v1 < v2) m = v1;

else m = v2;

cout << m;

}

## Завдання №4 Algotester Lab 3

#include <iostream>

using *namespace* std;

*int* main() {

*int* M, N;

cin >> M;

if (M < 0 || M > 100) return 0;

*int* mass1[M];

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

    cin >> mass1[i];

    if (mass1[i]<0 || mass1[i]>100) return 0;

}

cin >> N;

if (N < 0 || N > 100) return 0;

*int* mass2[N];

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

    cin >> mass2[i];

    if (mass2[i]<0 || mass2[i]>100) return 0;

}

*int* similar = 0;

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

    for (*int* j = 0; j < N; j++){

        if (mass1[i]==mass2[j]) similar++;

    }

}

*int* different = M + N - similar;

cout << similar << endl << different;

}

## Завдання №5 Class Practice Task

#include <iostream>

#include <string>

using *namespace* std;

*bool* palindrome(string *word\_num*, *int* *start*, *int* *end*){

    if(*word\_num*[*start*] != *word\_num*[*end*]) return false;

    if(*start* >= *end*) return true;

    return palindrome(*word\_num*, *start*+1, *end*-1);

}

*bool* palindrome(*int* *number*){

    string check\_number = to\_string(*number*);

    return palindrome(check\_number, 0, check\_number.length()-1);

}

*int* main(){

    string word;

*int* number;

    cout << "Give a word to check" << endl;

    cin >> word;

    cout << "Give a number to check" << endl;

    cin >> number;

    cout << "---------------------" << endl;

    if (palindrome(word, 0, word.length()-1)==true){

        cout << word << " " << "is a palindrome)" << endl; }

    else {

        cout << word << " " << "isn't a palindrome(" << endl; }

    if (palindrome(number)==true){

        cout << number << " " << "is a palindrome)" << endl; }

    else {

        cout << number << " " << "isn't a palindrome(" << endl; }

    return 0;

}

## Завдання №6 Self Practice Task

#include <iostream>

using *namespace* std;

*int* main() {

*int* N;

    cin >> N;

    if (N<1 || N>1000) return 0;

*int* ldrone = 0;

*int* rdrone = N-1;

*int* way[N];

    for (*int* i = 0; i < N; i++) {

        cin >> way[i];

        if (way[i]<1 || way[i]>5) return 0;

    }

    for (*int* i = 0; i < N; i++) {

        if (ldrone == rdrone) {

            cout << ldrone+1 << " " << rdrone+1 << endl;

            cout << "Collision";

            return 0;

        }

        else if (ldrone == rdrone-1) {

            cout << ldrone+1 << " " << rdrone+1 << endl;

            cout << "Stopped";

            return 0;

        }

        else if (ldrone > rdrone) {

            cout << ldrone+1 << " " << rdrone+1 << endl;

            cout << "Miss";

            return 0;

        }

        else {

            ldrone = ldrone + way[ldrone];

            rdrone = rdrone - way[rdrone];

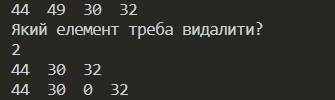
        }

    }

}

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

Завдання №1 VNS Lab 4



Затрачений час: 50 хвилин

Завдання №2 VNS Lab 5



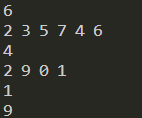
Затрачений час: 20 хвилин

Завдання №3 Algotester Lab 2



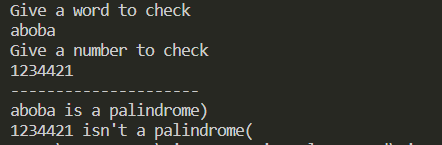
Затрачений час: 30 хвилин

Завдання №4 Algotester Lab 3



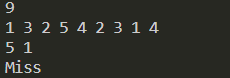
Затрачений час: 20 хвилин

Завдання №5 Class Practice Task



Затрачений час: 2 години

Завдання №6 Self Practice Task



Затрачений час: 30 хвилин

# **Висновки:**

Вивчено та засвоєно такі теми як:

* Одновимірні масиви.
* Двовимірні Масиви.
* Вказівники та Посилання.
* Динамічні масиви.
* Структури даних. Вкладені структури.
* Алгоритми обробки та робота з масивами та структурами.

З допомогою цих технологій успішно написано код для всіх програм