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-14

Грицишин Максим Іванович

# **Тема роботи:**

# Ознайомлення з масивами, деякими простими струкутурами даних та їхніми алгоритмами обробки Практичне застосування знань на основі виконання лабораторних та практичних робіт.

# **Мета роботи:**

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

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Масиви
* Тема №2: Двовимірні масиви
* Тема №3: Генерація рандомних чисел
* Тема №4: Тип даних string

1. Індивідуальний план опрацювання теорії:

* Тема №1: Масиви
  + Джерела Інформації
    - <https://acode.com.ua/urok-77-masyvy/>
    - ChatGPT, BingAI
    - Курс «Мови та парадигми програмування»
  + Що опрацьовано:
    - Ознайомлений з ініціалізацією масиву та роботою з ним
    - Використано у лабораторних та практичних роботах
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.11.23
  + Звершення опрацювання теми: 29.11.23
* Тема №2: Двовимірні масиви
  + Джерела Інформації:
    - <https://acode.com.ua/urok-81-bagatovymirni-masyvy/>
    - ChatGPT, BingAI
  + Що опрацьовано:
    - Ознайомлений з ініціалізацією масиву та роботою з ним
    - Використано у лабораторних та практичних роботах
  + Статус: Ознайомлений
  + Початок опрацювання теми: 27.11.23
  + Звершення опрацювання теми: 30.11.23
* Тема №3: Генерація рандомних чисел
  + Джерела Інформації:
    - <https://acode.com.ua/urok-74-generatsiya-randomnyh-chysel/>
  + Що опрацьовано:
    - Ознайомлений з теорією
    - Викоритсав у лабораторній роботі
  + Статус: Ознайомлений
  + Початок опрацювання теми: 29.11.23
  + Звершення опрацювання теми: 29.11.23
* Тема №4 Тип даних string
  + Джерела Інформації:
    - <https://acode.com.ua/urok-210-dovzhyna-i-yemnist-std-string/>
  + Що опрацьовано:
    - Використав у лабораторних та практичній роботах
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 16.11.23
  + Звершення опрацювання теми: --

# **Виконання роботи:**

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

Завдання №1 Лабораторна робота №4 "Робота з одновимірними масивами"

* Варіант завдання: 6
* Деталі завдання: 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел;

2) Роздрукувати отриманий масив;

3) Знищити елемент із заданим номером;

4) Додати після першого парного елемента масиву елемент зі значенням M[ I-1]+2;

5) Роздрукувати отриманий масив.

* Важливі деталі для врахування в імплементації програми: При зменшенні або збільшенні довжини масиву необхідно змінювати його реальну довжину.

Завдання №2 Лабораторна робота №5 "Функції і масиви"

* Варіант завдання: 6
* 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: --

Завдання №3 Лабораторна робота №2 у Algotester

* Варіант завдання: 2
* Деталі завдання: Дано масив r розмiром N. Також дано 3 цiлих числа. Спочатку потрібно видалити з масиву цi 3 числа. Пiсля цього перетворити цей масив у масив сум, розмiром Nnew − 1 (розмiр нового масиву пiсля видалення елементiв), який буде вiдображати суми сусiднiх елементiв нового масиву. Далi необхiдно вивести масив сум на екран.
* Вхiднi данi: У першому рядку цiле число N - кiлькiсть чисел. У другому рядку масив r, який складається з N цiлих чисел. У третьому рядку 3 цiлих числа, a, b, c, якi треба видалити з масиву
* Вихідні дані: У першому рядку цiле число M - кiлькiсть чисел у масивi, який буде виведено. У наступному рядку M чисел - новий масив
* Важливі деталі для врахування в імплементації програми: є обмеження у часі, пам'яті та обмеження змінних.

Завдання №4 Лабораторна робота №3 у Algotester

* Варіант завдання: 3
* Деталі завдання: Дано стрiчку s. Потрібно зробити компресiю стрiчки, тобто якщо якась буква йде бiльше одного разу пiдряд у стрiчцi замiнити її на букву + кiлькiсть входжень пiдряд.
* Вхідні дані: У першому рядку стрiчка s.
* Вихідні дані: Стрiчка Scompressed
* Важливі деталі для врахування в імплементації програми: є обмеження у часі, пам'яті та обмеження змінних.

Завдання №5 Практичне завдання “Перевірка чи слово або число є паліндромом”

* Деталі завдання: Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.
* Важливі деталі для врахування в імплементації:

1. Визначення функції:

Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.

1. Приклад визначення функції:

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

1. Перевантаження функцій:
   1. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
   2. *bool isPalindrome*(ціле число);
2. Рекурсія:

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.

Завдання №6 Практичне завдання Лабораторна робота №3 з Algotester

* Варіант завдання: 2
* Деталі завдання: Дано 2 масиви розміром N та M. Значення у цих масивах унікальні.

Потрібно вивести у першому рядку кількість елементів, які наявні в обох масивах одночасно, у другому кількість унікальних елементів в обох масивах разом.

Вхідні дані:

У першому рядку ціле число N.

у другому рядку N цілих чисел a1..an.

У третьому рядку ціле число M.

у четвертому рядку M цілих чисел b1..bn

Вихідні дані: У першому рялку 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).

* Важливі деталі для імплементації: Користуватися std::set та std::map та їх похідними заборонено.

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

Програма №1 Лабораторна робота №4 "Робота з одновимірними масивами"

* Запланований час на реалізацію: 2год
* Важливі деталі для врахування в імплементації: щоб при компіляції генератор випадкових чисел видавав кожного разу різні числа, варто використовувати у srand() time().

Програма №2 Лабораторна робота №5 "Функції і масиви"

* Запланований час на реалізацію: 2год
* Важливі деталі для врахування в імплементації: --

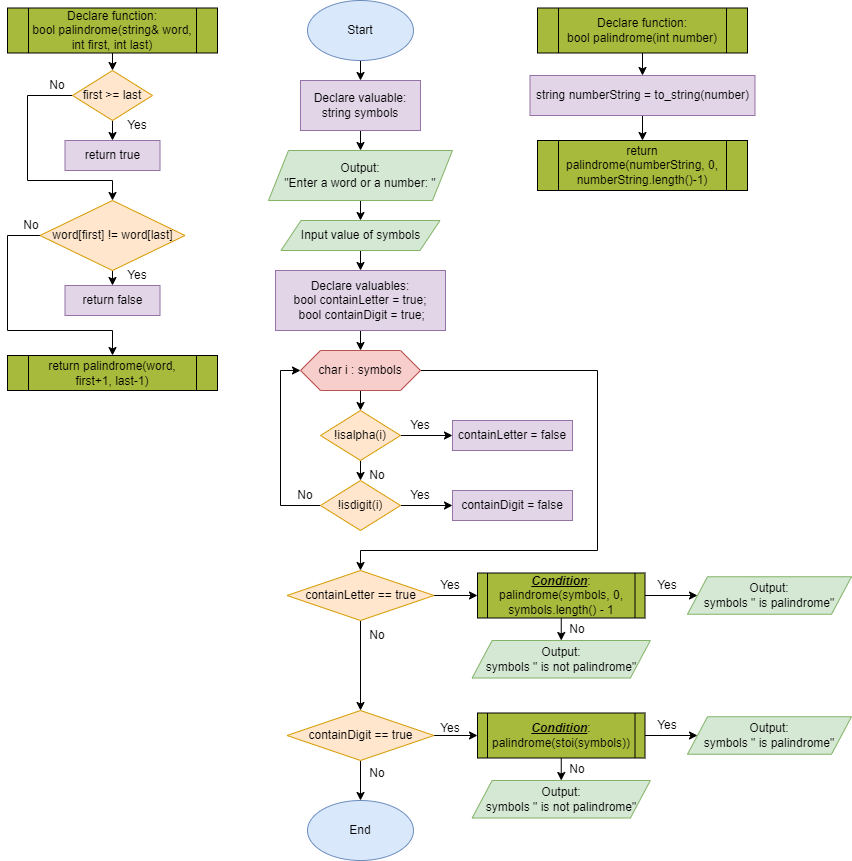
Програма №3 Лабораторна робота №2 у Algotester

* Запланований час на реалізацію: 1год
* Важливі деталі для врахування в імплементації: для масиву {a, b, c, d} масив сум виглядає так: {a+b, b+c, c+d}

Програма №4 Лабораторна робота №3 у Algotester

* Запланований час на реалізацію: 2год
* Важливі деталі для врахування в імплементації: великi та маленькi букви вважаються рiзними.

Програма №5 Практичне завдання “Перевірка чи слово або число є паліндромом”

* Запланований час на реалізацію: 2.5год
* Важливі деталі для врахування в імплементації: для функції palindrome потрібно використати і рекурсію, і перевантаження
* : 

Малюнок 1: Блок-схема до програми №5

Програма №6 Практичне завдання Лабораторна робота №3 з Algotester

* Запланований час на реалізацію: 1.5год
* Важливі деталі для врахування в імплементації: --

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

Завдання №1 Лабораторна робота №4 "Робота з одновимірними масивами"

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int main() {

    int size=12;

    srand(time(0));

    int array[size];

    for(int a=0; a<size; ++a) {

        array[a]=rand();

        cout << array[a] << " ";

    }

    int index;

    do {cout << "\nEnter the index of the array element you want to delete: ";

    cin >> index;}

    while (index<0 || index>(size-1));

    for (int a = index; a < size; ++a) {

        array[a] = array[a+1];

    }

    --size;

    int a=0;

    while(a<size-1) {

        if(array[a] % 2 == 0) {

            ++size;

            for (int b = size-1; b > a+1; --b) {

                array[b] = array[b-1];

            }

            array[a+1] = array[a] + 2;

            break;

        }

        a++;

    }

    for(int a=0; a<size; ++a) {

        cout << array[a] << " ";

    }

}

Деталі завдання: використано цикли do while, while і for, генератор випадкових чисел для введення чисел у масив. Тоді як цикли застосовано для видалення елемента, введення нового елемента та виведення масиву старого і нового.

Покликання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/661/files#diff-38281ad76d65251416bdb09e6d44d3a45ad96e84149cc34de997e41ccca36856>

Завдання №2 Лабораторна робота №5 "Функції і масиви"

#include <iostream>

using namespace std;

int main() {

    int rowNum, columnNum;

    cout << "Enter the amount of rows: ";

    cin >> rowNum;

    cout << "\nEnter the amount of columns: ";

    cin >> columnNum;

    int matrix[rowNum][columnNum];

    for (int i = 0; i<rowNum; ++i) {

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

            cin >> matrix[i][j];

        }

    }

    for (int i = 0; i<rowNum; ++i) {

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

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

        }

        cout << endl;

    }

    for (int i=0; i<rowNum; ++i){

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

            int interim = matrix[i][j];

            int maxRow = interim;

            int minColumn = interim;

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

                if(matrix[i][a]>maxRow) {

                    maxRow = matrix[i][a];

                }

            }

            for(int b=0; b<rowNum; ++b) {

                if(matrix[b][j]<minColumn) {

                    minColumn = matrix[b][j];

                }

            }

            if(maxRow == interim && minColumn == interim) {

                cout << "Saddle point: " << "(" << i << ", " << j << ")" << endl;

            }

        }

    }

    for (int i=0; i<rowNum; ++i){

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

            int interim = matrix[i][j];

            int minRow = interim;

            int maxColumn = interim;

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

                if(matrix[i][a]<minRow) {

                    minRow = matrix[i][a];

                }

            }

            for(int b=0; b<rowNum; ++b) {

                if(matrix[b][j]>maxColumn) {

                    maxColumn = matrix[b][j];

                }

            }

            if(minRow == interim && maxColumn == interim) {

                cout << "Saddle point: " << "(" << i << ", " << j << ")" << endl;

            }

        }

    }

    return 0;

}

Деталі завдання: використано цикли for та вкладені цикли для знаходження сідлової точки у двовимірному масиві.

Покликання на пул-реквест:

Завдання №3 Лабораторна робота №2 у Algotester

#include <iostream>

using namespace std;

int main() {

    int N;

    cin >> N;

    int r[N];

    for (int a = 0; a < N; ++a) {

        cin >> r[a];

    }

    int a, b, c;

    cin >> a >> b >> c;

    for(int i = 0; i < N; ++i) {

        if (r[i] == a || r[i] == b || r[i] == c) {

            for (int k = i; k < N - 1; ++k) {

                r[k] = r[k+1];

            }

            --N;

            --i;

        }

    }

    int M=N-1;

    if (M<0) {

        M = 0;

    }

    cout << M << endl;

    int sum[M];

    for(int i=0; i<M; ++i) {

        sum[i] = r[i] + r[i+1];

        cout << sum[i] << " ";

    }

    return 0;

}

## Деталі завдання: використано цикли для видалення чисел, введених користувачем, з масиву і виведення нового.

Покликання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/661/files#diff-db95ff256e3d7bbcb87d26b51b9be3951a448a77af224c31622747f38b31b03e>

Завдання №4 Лабораторна робота №3 у Algotester

#include <iostream>

#include <string>

using namespace std;

int main() {

    string s;

    cin >> s;

    string sComp;

    int amount=1;

    int a=0;

    while (a+1<s.length()) {

        if (s[a] == s[a+1]) {

            ++amount;

        } else {

            sComp += s[a];

            if (amount>1) {

                sComp += to\_string(amount);

            }

            amount = 1;

        }

        ++a;

    }

    sComp += s.back();

    if (amount>1) {

        sComp += to\_string(amount);

    }

    cout << sComp;

}

Деталі завдання: використано .length(), to\_string, .back() для проведення компресії введеної користувачем стрічки.

Покликання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/661/files#diff-e822a77571ce9d3d95bf343d4cfed47dc242dd8fd7921f2f02193c0171f7d386>

Завдання №5 Практичне завдання “Перевірка чи слово або число є паліндромом”

#include <iostream>

#include <string>

#include <cctype>

using namespace std;

bool palindrome(string& word, int first, int last) {

    if (first >= last) {

        return true;

    }

    if (word[first] != word[last]) {

        return false;

    }

    return palindrome(word, first+1, last-1);

}

bool palindrome(int number) {

    string numberString = to\_string(number);

    return palindrome(numberString, 0, numberString.length()-1);

}

int main() {

    string symbols;

    cout << "Enter a word or a number: ";

    cin >>  symbols;

    bool containLetter = true;

    bool containDigit = true;

    for (char i : symbols) {

        if (!isalpha(i)) {

            containLetter = false;

        }

        if (!isdigit(i)) {

            containDigit = false;

        }

    }

    if(containLetter == true) {

        if (palindrome(symbols, 0, symbols.length() - 1)) {

            cout << symbols << " is palindrome" << endl;

        } else {

            cout << symbols << " is not palindrome" << endl;

        }

    } else if (containDigit == true) {

        if (palindrome(stoi(symbols))) {

            cout << symbols << " is palindrome" << endl;

        } else {

           cout << symbols << " is not palindrome" << endl;

        }

    }

    return 0;

}

Деталі завдання: використано перевантаження функції та рекурсивну функцію, аби визначити чи є слово чи число паліндромом. Також використано stoi та to\_string.

Покликання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/661/files#diff-4afa4533ccec4cff1141ca6e602a0e2faa03c7eb81a8b4470396c272f5949c4f>

Завдання №6 Практичне завдання Лабораторна робота №3 у Algotester

#include <iostream>

using namespace std;

int main() {

    int N, M;

    cin >> N;

    int a[N];

    for(int i=0; i<N; ++i) {

        cin >> a[i];

    }

    cin >> M;

    int b[M];

    for(int i=0; i<M; ++i) {

        cin >> b[i];

    }

    int similar=0;

    int unique=N+M;

    for(int i=0; i<N; ++i) {

        for(int j=0; j<M; ++j) {

            if(a[i] == b[j]) {

                similar++;

                unique--;

            }

        }

    }

    cout << similar << endl;

    cout << unique << endl;

}

Деталі завдання: потрібно визначити кількість однакових і кількість унікальних елементів з двох масивів. Використано лише цикли for та умовний оператор if

Покликання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/661/files#diff-f63f3687415578f87bdae446cf0a4c0c9d42e86a463a4a68df40ff6b281e5ea8>

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

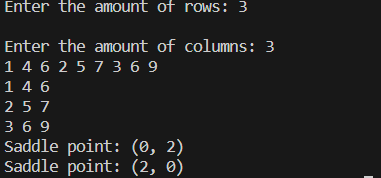
Завдання №1 Лабораторна робота №4 "Робота з одновимірними масивами"



Малюнок 2: Результат компіляції: видалено 4 елемент і додано елемент, що збільшено на 2 від першого парного елемента

Час затрачений на виконання завдання: 1.5год

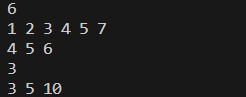
Завдання№2 Лабораторна робота №5 "Функції і масиви"



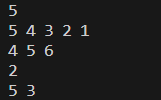
Малюнок 3: Успішна компіляція програми №2 (у масиві виявилися 2 сідлові точки)

Час затрачений на виконання завдання: 4год

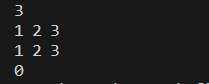
Завдання №3 Лабораторна робота №2 у Algotester



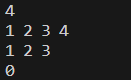
Малюнок 4: Успішна компіляція програми №3 (перших три рядки - input, решта - output) 1 із запропонованих варіантів stdin/stdout



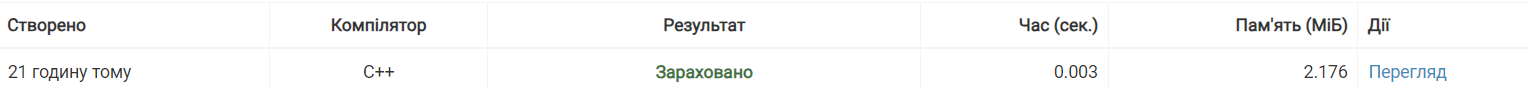
Малюнок 5: Успішна компіляція програми №3: 2 із запропонованих варіантів stdin/stdout



Малюнок 6: Успішна компіляція програми №3: 3 із запропонованих варіантів stdin/stdout



Малюнок 7: Успішна компіляція програми №3: 3 із запропонованих варіантів stdin/stdout



Малюнок 8: Підтвердження коректності програми

Час затрачений на виконання завдання: 1 год

Завдання №4 Лабораторна робота №3 у Algotester



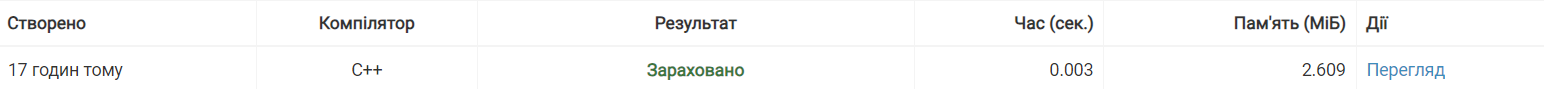
Малюнок 9: Успішна компіляція програми №4(1 рядок - input, 2 рядок - output): 1 із запропонованих варіантів stdin/stdout



Малюнок 10: Успішна компіляція програми №4: 2 із запропонованих варіантів stdin/stdout



Малюнок 11: Успішна компіляція програми №4: 3 із запропонованих варіантів stdin/stdout



Малюнок 12: Підтвердження коректності програми

Час затрачений на виконання завдання: 2 год

Завдання №5 Практичне завдання “ Перевірка чи слово або число є паліндромом ”



Малюнок 13: Успішна компіляція програми №5: перевірка слова на паліндром



Малюнок 14: Успішна компіляція програми №5: перевірка слова на паліндром



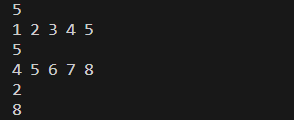
Малюнок 15: Успішна компіляція програми №5: перевірка числа на паліндром



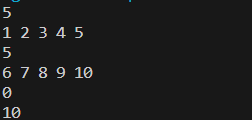
Малюнок 16: Успішна компіляція програми №5: перевірка числа на паліндром

Час затрачений на виконання завдання: 1.5 год.

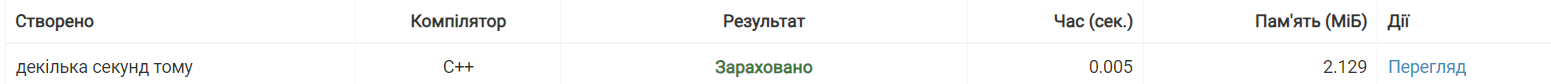
Завдання №6 Практичне завдання Лабораторна робота №3 у Algotester



Малюнок 17: Успішна компіляція програми №5(перші 4 рядки - input, 2 останні - output): 1 із запропонованих варіантів stdin/stdout



Малюнок 18: Успішна компіляція програми №5: 2 із запропонованих варіантів stdin/stdout



Малюнок 19: Підтвердження коректності програми

Час затрачений на виконання завдання: 1.5 год.

# **Висновки:**

* Ознайомлений з одновимірними масивами
* Ознайомлений з двовимірними масивами
* Ознайомлений з генератором рандомних чисел
* Ознайомлений з певними простими структурами даних та їхніми алгоритмами обробки
* Успішне використання нововивченого матеріалу під час виконання лабораторних та практичних робіт