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-13

Мельников Дмитро Володимирович

# **Тема роботи:**

Прості структури даних. Одновимірні масиви. Двовимірні Масиви. Алгоритми обробки

# **Мета роботи:**

Навчитись створювати одновимірні масиви,двовимірні.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

**Тема №1: Програмування на Lab#: VNS Lab 4**

**Тема №2: Програмування на Lab#: VNS Lab 5**

**Тема №3: Algotester lab 2**

**Тема №4: Algotester Lab 3**

**Тема №5: class practice**

* **Тема №1: Програмування на Lab#: VNS Lab 4**
  + Джерела Інформації:
  + <https://markoutte.me/students/pointers-of-cpp/>
  + https://www.youtube.com/watch?v=n0saIDd3H-M
  + Що опрацьовано:
    - Вказівники та масиви
  + Статус: Ознайомлений
  + Початок опрацювання теми: Дата 01.11.2023
  + Звершення опрацювання теми: Дата 08.11.2023
  + **Тема №3**: **Програмування на Lab#: VNS Lab 5**
  + Джерела Інформації:
    - <https://metanit.com/cpp/tutorial/4.7.php>
    - <https://www.youtube.com/watch?v=r1wWlUCZW08>
  + Що опрацьовано:
    - Масиви в функціях , передача масивів
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.11.2023
  + Звершення опрацювання теми: 08.11.2023

# **Виконання роботи:**

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

## **Програмування на Lab#: VNS Lab 4**

* Варіант 25

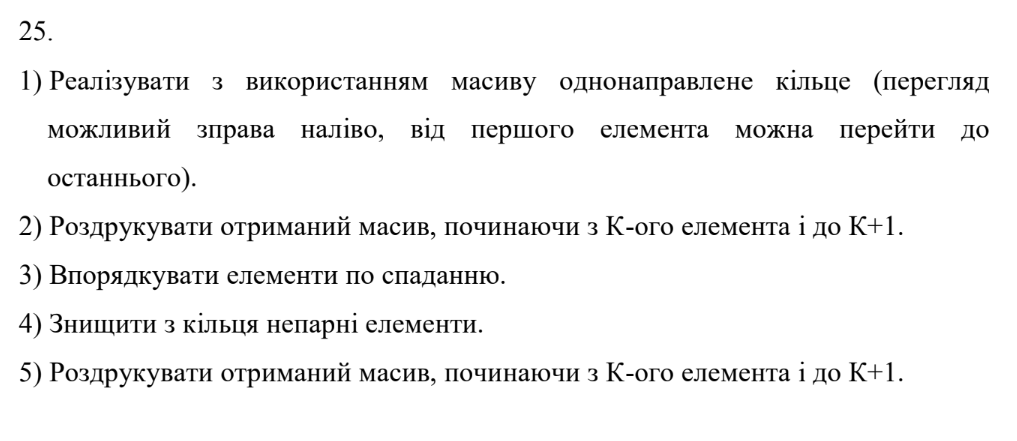


Рисунок : Програмування на Lab#: VNS Lab 4

**Lab# programing: VNS lab 5**

**-Варіант 25**

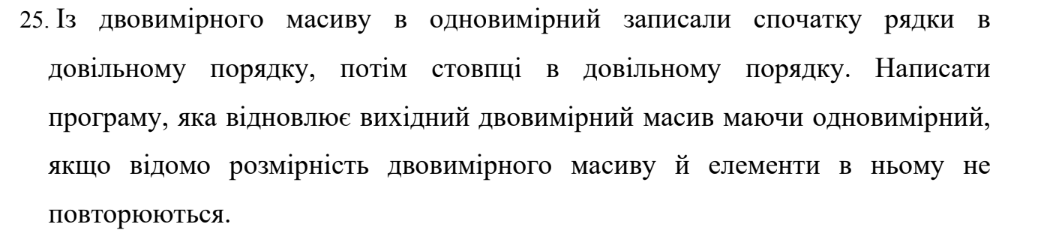


Рисунок : Програмування на Lab#: VNS Lab 5

**Algotester lab 2**

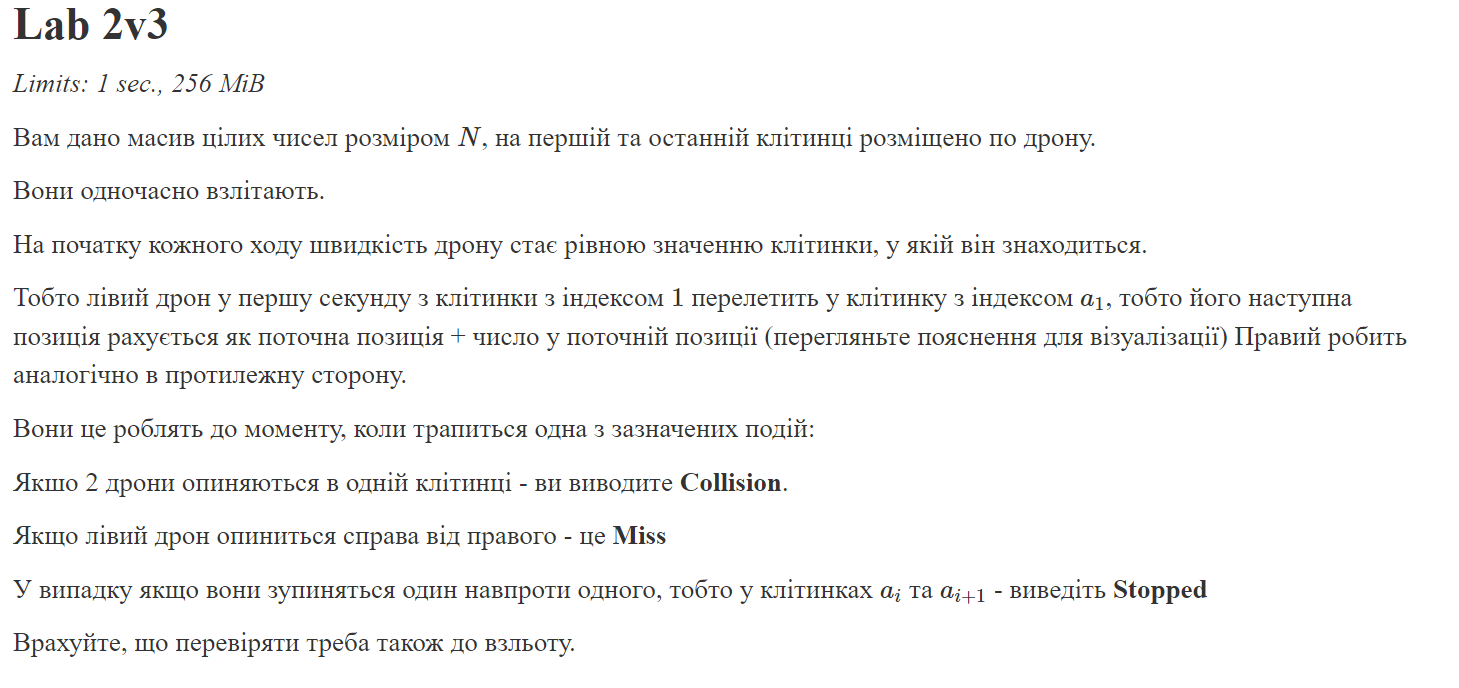
****

Рисунок algotester lab 2 v 3

**-Варіант 2**

**Algotester Lab 3**

**-Варіант 3**

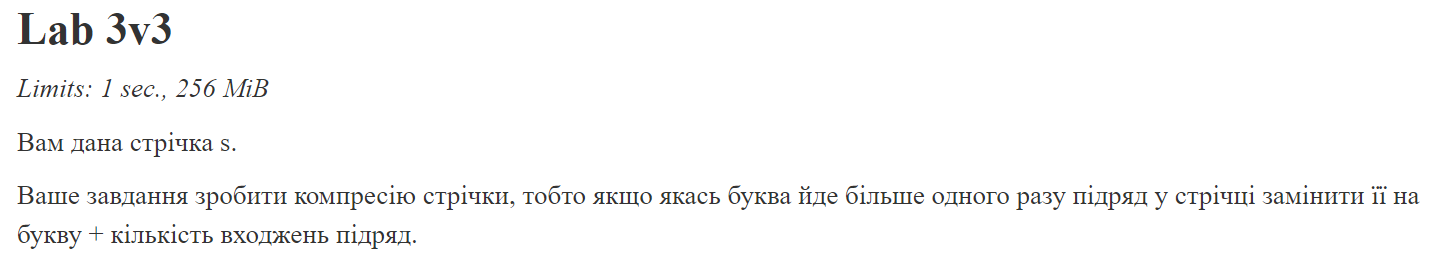
****

Рисунок algo lab 3 v 3

**Class practice**

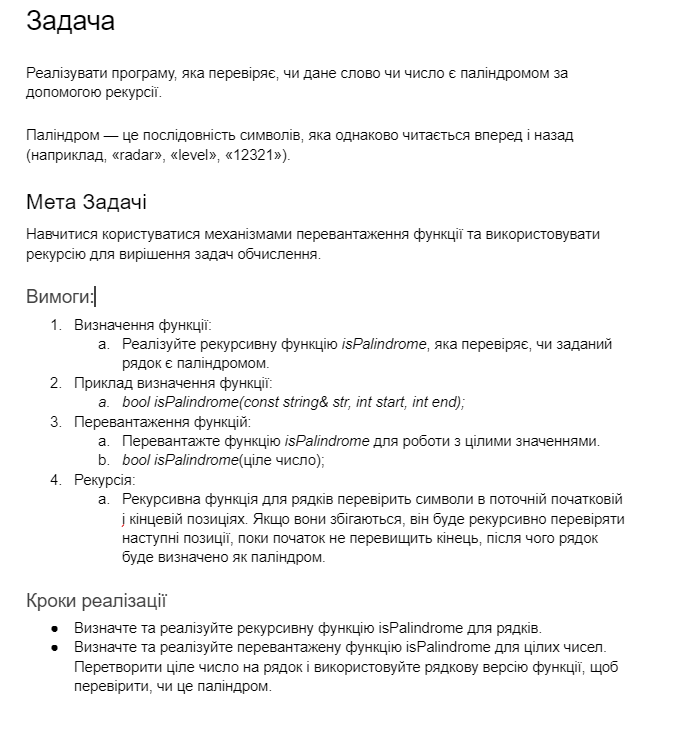
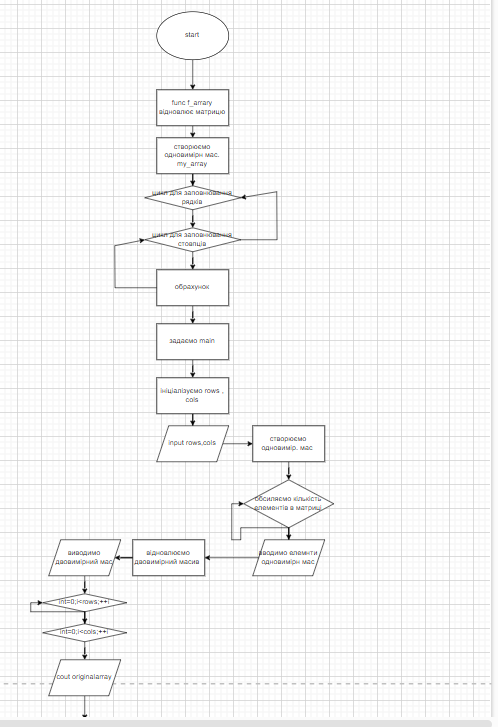


Рисунок Class practice

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

**Lab# programing: VNS lab 5**



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

**Програмування на Lab#: VNS Lab 4**

**Git link:**

#include<iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

    int n;

    cout << "enter size of ring: ";

    cin >> n;

    vector<int> ring(n);

    cout << "enter elements of ring: ";

    for (int i = 0; i < n; ++i) {

        cin >> ring[i];

    }

    int k;

    cout << "enter K: ";

    cin >> k;

    //підмасив від K до K+1

    cout << "subarray from k to k+1 ";

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

        cout << ring[i % n] << " ";

    }

    cout << endl;

    // Сортування у спадаючому порядку

    sort(ring.rbegin(), ring.rend());

    // Видалення непарних елементів

    ring.erase(remove\_if(ring.begin(), ring.end(), [](int x) { return x % 2 != 0; }), ring.end());

    // Вивід кільцевого масиву від K до K+1 після видалення непарних елементів

    cout << "Після видалення непарних елементів: ";

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

        cout << ring[i % ring.size()] << " ";

    }

    cout << endl;

    return 0;

}

Рисунок code for VNS Lab 4

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

**VNS Lab 4.**

enter size of ring: 3

enter elements of ring: 3

6

1

enter K: 1

subarray from k to k+1 6 1

Після видалення непарних елементів: 6 6

PS C:\Epic4>

Рисунок VNS Lab 4 result.

**Lab# programing: VNS lab 5**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

// Функція для відновлення двовимірного масиву

vector<vector<int>> f\_array(const vector<int>& value\_array, int rows, int cols)

{

    // Створюємо порожній двовимірний масив

    vector<vector<int>> my\_array(rows, vector<int>(cols, 0));

    // Заповнюємо масив елементами з одновимірного масиву

    for (int i = 0; i < rows; ++i) {

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

            my\_array[i][j] = value\_array[i \* cols + j];

        }

    }

    return my\_array;

}

int main() {

    // Задаємо розмірність двовимірного масиву

    int rows, cols;

    cout << "enter number of rows: ";

    cin >> rows;

    cout << "enter number of cols: ";

    cin >> cols;

    // Створюємо та виводимо випадковий одновимірний масив

    vector<int> value\_array(rows \* cols);

    cout << "enter elements of one-dimensional maisv:" << endl;

    for (int i = 0; i < rows \* cols; ++i) {

        cin >> value\_array[i];

    }

    // Відновлюємо двовимірний масив

    vector<std::vector<int>> originalArray = f\_array(value\_array, rows, cols);

    // Виводимо відновлений двовимірний масив

    cout << "Recovered two-dimensional array:"<< endl;

    for (int i = 0; i < rows; ++i) {

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

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

        }

        cout << '\n';

    }

    return 0;

}

Рисунок code for VNS lab 5

**Lab# programing: VNS lab 5**

**Code result.**

enter number of rows: 2

enter number of cols: 2

enter elements of one-dimensional maisv:

1

2

3

4

Recovered two-dimensional array:

1 2

3 4

PS C:\Epic4>

**Algotester lab 2 v 1**

#include <iostream>

#include <vector>

#include <algorithm>

int main() {

    int N;

    std::cin >> N;

    std::vector<int> r(N);

    for (int i = 0; i < N; ++i) {

        std::cin >> r[i];

    }

    std::sort(r.begin(), r.end());

    int min;

    if (N == 1 || r[0] == r[N - 2] || r[1] == r[N - 1]) {

        min = 0;

    } else {

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

    }

    std::cout << min << std::endl;

    return 0;

}

Рисунок algotester code 2 v1

4\ucrt64\bin\gdb.exe' '--interpreter=mi'

4

2

3

5

6

3

PS C:\Epic4>

Рисунок algotester 2 v 1 result

**Algotester lab 3 v 3**

#include <iostream>

#include <string>

int main() {

    std::string S;

    std::cin >> S;

    std::string compress;

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

    {

        char curr\_char = S[i];

        int count = 1;

        while (i + 1 < S.size() && S[i + 1] == curr\_char)

        {

            ++count;

            ++i;

        }

        compress += curr\_char;

        if (count > 1) {

            compress += std::to\_string(count);

        }

    }

    std::cout << compress << std::endl;

    return 0;

}

Рисунок algotester code 3 v 3

aadddwwdaaadc

a2d3w2da3dc

Рисунок algotester result 3 v 3

**Class practice**

#include <iostream>

#include <string>

bool isPalindrome(const std::string& str, int firstind, int endind) { // перевірка чи рядок є паліндромом

    if (firstind >= endind) {   //якщо початок перевищив кінець, рядок є паліндромом

        return true;

    }

    if (str[firstind] == str[endind]) { //порівнюємо символи спочатку і в кунці

        return isPalindrome(str, firstind + 1, endind - 1); // Рекурсія, виклик для наступної пари символів

    }

    return false;

}

// Перевантажена функція чи ціле число є паліндромом

bool isPalindrome(int number) {

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

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

}

int main() {

    std::string word;

    std::cout << "Введіть слово або число: ";

    std::cin >> word;

    // Виклик рекурсивної функції для рядка

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

        std::cout << "Це слово або число є паліндромом.\n";

    } else {

        std::cout << "Це слово або число не є паліндромом.\n";

    }

    return 0;

}

Рисунок class practive code

Введіть слово або число: level

Це слово або число є паліндромом.

PS C:\Epic4>

Рисунок class practice result

# **Висновки:**