

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Кафедра систем штучного інтелекту



## 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:** «Основи програмування»

до:

ВНС Лабораторної Роботи №4

ВНС Лабораторної Роботи №5

Алготестер Лабораторної Роботи №2

Алготестер Лабораторної Роботи №3

Практичних Робіт до блоку №4

**Виконав:**

Студент групи ІІІ-12  
Стик Назарій Олегович

Львів 2024

## **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:**

Ознайомлення з основами роботи з одновимірними та двовимірними масивами, а також з концепціями вказівників та посилань. Вивчення динамічних масивів та структур даних, зокрема вкладених структур, для організації і зберігання даних. Окрему увагу приділено алгоритмам обробки масивів та структур, а також практичним аспектам роботи з ними.

## **Теоретичні відомості:**

У даній роботі розглядаються основи роботи з масивами, зокрема одновимірними та двовимірними масивами, що дозволяють зберігати та обробляти великі обсяги даних у зручній формі. Вивчаються вказівники та посилання, які надають можливість ефективного доступу до елементів пам'яті. Окрему увагу приділено динамічним масивам, які дозволяють змінювати розмір під час виконання програми. Досліджуються також структури даних, їхнє використання для організації складних типів даних, а також вкладені структури для створення багаторівневих структурованих даних. Розглядаються основні алгоритми обробки масивів і структур, що забезпечують оптимізацію роботи з великими обсягами інформації.

## **Джерела:**

- Декілька відео на YouTube:
  - Масиви – <https://www.youtube.com/watch?v=QFrJQq6Iox8>
  - Двовимірні Масиви - <https://www.youtube.com/watch?v=Vh4krbTnTAA>
  - Вказівники та Посилання - <https://www.youtube.com/watch?v=slzcWKWCMBg>
  - Динамічні масиви - <https://www.youtube.com/watch?v=jzJlq35dQII>
  - Структури даних - <https://www.youtube.com/watch?v=SFEROgwxicA>
- Певну інформацію брав на сайтах:
  - <https://acode.com.ua/urok-77-masyvy/>
  - <https://acode.com.ua/urok-84-vkazivnyky/>
  - <https://acode.com.ua/urok-90-dynamichni-masyvy/>
  - <https://acode.com.ua/urok-64-struktury/>
- Також вивчив багато інформації за допомогою ChatGPT.

## Виконання роботи:

- **Завдання №1:** Theory Education Activities

Очікувано часу: 4 дні.

Витрачено часу: 3 дні.

- **Завдання №2:** Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7)

Очікувано часу: 1 година.

Витрачено часу: 1 година.

- **Завдання №3:** Lab# programming: VNS Lab 4 (Variant 19)

### Код:

```
1  #include <iostream>
2
3  using namespace std;
4
5  // Функція для друку масиву з K-ого елемента до K+1
6  void printFromKToKPlus1(int arr[], int size, int K)
7  {
8      for (int i = 0; i < size; ++i)
9      {
10         cout << arr[(K + 1) % size] << " ";
11     }
12     cout << endl;
13 }
14
15 // Функція для додавання першого і останнього елементів в кільце
16 void addFirstAndLast(int arr[], int& size)
17 {
18     if (size > 0)
19     {
20         int first = arr[0];
21         int last = arr[size - 1];
22         arr[size++] = first;
23         arr[size++] = last;
24     }
25 }
26
27 // Функція для видалення непарних елементів з кільця
28 void removeOddElements(int arr[], int& size)
29 {
30     int newSize = 0;
31     for (int i = 0; i < size; ++i)
32     {
33         if (arr[i] % 2 == 0)
34         {
35             arr[newSize++] = arr[i];
36         }
37     }
38     size = newSize;
39 }
40
41 int main() {
42     const int MAX_SIZE = 20;
43     int arr[MAX_SIZE] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
44     int size = 10;
45     int K = 3;
46
47     // 1) Створення однонаправленого кільця
48     // Масив вже є однонаправленим кільцем
49
50     // 2) Друк масиву з K-ого елемента до K+1
51     cout << "Initial ring from K to K+1: ";
52     printFromKToKPlus1(arr, size, K);
53
54     // 3) Додавання першого і останнього елементів в кільце
55     addFirstAndLast(arr, size);
56     cout << "Ring after adding first and last elements: ";
57     printFromKToKPlus1(arr, size, K);
58
59     // 4) Видалення непарних елементів з кільця
60     removeOddElements(arr, size);
61     cout << "Ring after removing odd elements: ";
62     printFromKToKPlus1(arr, size, K);
63
64     // 5) Друк масиву з K-ого елемента до K+1
65     cout << "Final ring from K to K+1: ";
66     printFromKToKPlus1(arr, size, K);
67
68     return 0;
69 }
```

Очікувано часу: **30 хвилин.**

Витрачено часу: **1 година.**

- **Завдання №4:** Lab# programming: VNS Lab 5 (Variant 19)

**Код:**

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  // Функція для перевірки, чи можна розбити рядок на підстрічки довжиною N
7  bool canFormMatrix(const string& str, int N)
8  {
9      if (str.length() != N * N)
10     {
11         return false;
12     }
13
14     // Створення двовимірного масиву N x N
15     char matrix[100][100]; // Припускаємо, що N <= 100
16
17     // Заповнення двовимірного масиву
18     int index = 0;
19     for (int i = 0; i < N; ++i)
20     {
21         for (int j = 0; j < N; ++j)
22         {
23             matrix[i][j] = str[index++];
24         }
25     }
26
27     // Перевірка, чи елементи першого стовпця розташовані в порядку зростання
28     for (int i = 1; i < N; ++i)
29     {
30         if (matrix[i][0] < matrix[i - 1][0])
31         {
32             return false;
33         }
34     }
35
36     return true;
37 }
38
39 int main()
40 {
41     string input;
42     int N;
43
44     // Введення рядка та значення N
45     cout << "Enter the string: ";
46     cin >> input;
47     cout << "Enter the value of N: ";
48     cin >> N;
49
50     // Перевірка, чи можна розбити рядок на підстрічки довжиною N і розташувати їх у порядку зростання в першому стовпці
51     if (canFormMatrix(input, N))
52     {
53         cout << "Yes, it is possible to form the matrix with the first column in ascending order." << endl;
54     }
55     else
56     {
57         cout << "No, it is not possible to form the matrix with the first column in ascending order." << endl;
58     }
59
60     return 0;
61 }
```

Очікувано часу: **30 хвилин.**

Витрачено часу: **1 година.**

- **Завдання №5:** Lab# programming: Algotester Lab 2 (Variant 1)

**Код:**

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main()
7  {
8      int N;
9      cin >> N;
10     vector<int> r(N);
11
12     for (int i = 0; i < N; i++)
13     {
14         cin >> r[i];
15     }
16
17     if (N == 1)
18     {
19         cout << 0 << endl;
20         return 0;
21     }
22
23     // Ініціалізуємо змінні для найменших та найбільших елементів
24     int min1 = 100000, min2 = 100000;
25     int max1 = 0, max2 = 0;
26
27     for (int i = 0; i < N; i++)
28     {
29         // Оновлюємо найменші елементи
30         if (r[i] < min1)
31         {
32             min2 = min1;
33             min1 = r[i];
34         }
35         else if (r[i] < min2)
36         {
37             min2 = r[i];
38         }
39
40         // Оновлюємо найбільші елементи
41         if (r[i] > max1)
42         {
43             max2 = max1;
44             max1 = r[i];
45         }
46         else if (r[i] > max2)
47         {
48             max2 = r[i];
49         }
50     }
51
52     // Обчислюємо можливі варіанти мінімальної втоми
53     int option1 = max1 - min2; // Видаляємо min1
54     int option2 = max2 - min1; // Видаляємо max1
55
56     // Знаходимо мінімальне значення між двома варіантами
57     int m = min(option1, option2);
58
59     cout << m << endl;
60
61     return 0;
62 }
63

```

Очікувано часу: **30** хвилин.

Витрачено часу: **1** година.

- Завдання №6: Lab# programming: Algotester Lab 3 (Variant 2)

Код:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int N, M;
8      cin >> N;
9      int a[N]; // Перший масив
10     for (int i = 0; i < N; i++)
11     {
12         cin >> a[i];
13     }
14
15     cin >> M;
16     int b[M]; // Другий масив
17     for (int i = 0; i < M; i++)
18     {
19         cin >> b[i];
20     }
21
22     // Знаходження кількості спільних елементів
23     int countSame = 0;
24     for (int i = 0; i < N; i++)
25     {
26         for (int j = 0; j < M; j++)
27         {
28             if (a[i] == b[j])
29             {
30                 countSame++;
31                 break; // Знайшли спільний елемент, переходимо до наступного елемента з a
32             }
33         }
34     }
35     cout << countSame << endl;
36
37     // Знаходження кількості унікальних елементів
38     bool unique[201] = {false}; // Масив для відстеження унікальних елементів (0-100 для a і 101-200 для b)
39     for (int i = 0; i < N; i++)
40     {
41         unique[a[i]] = true;
42     }
43     for (int i = 0; i < M; i++)
44     {
45         unique[b[i]] = true;
46     }
47
48     int countUnique = 0;
49     for (int i = 0; i <= 100; i++)
50     {
51         if (unique[i])
52         {
53             countUnique++;
54         }
55     }
56     cout << countUnique << endl;
57
58     return 0;
59 }
```

Очікувано часу: 30 хвилин.

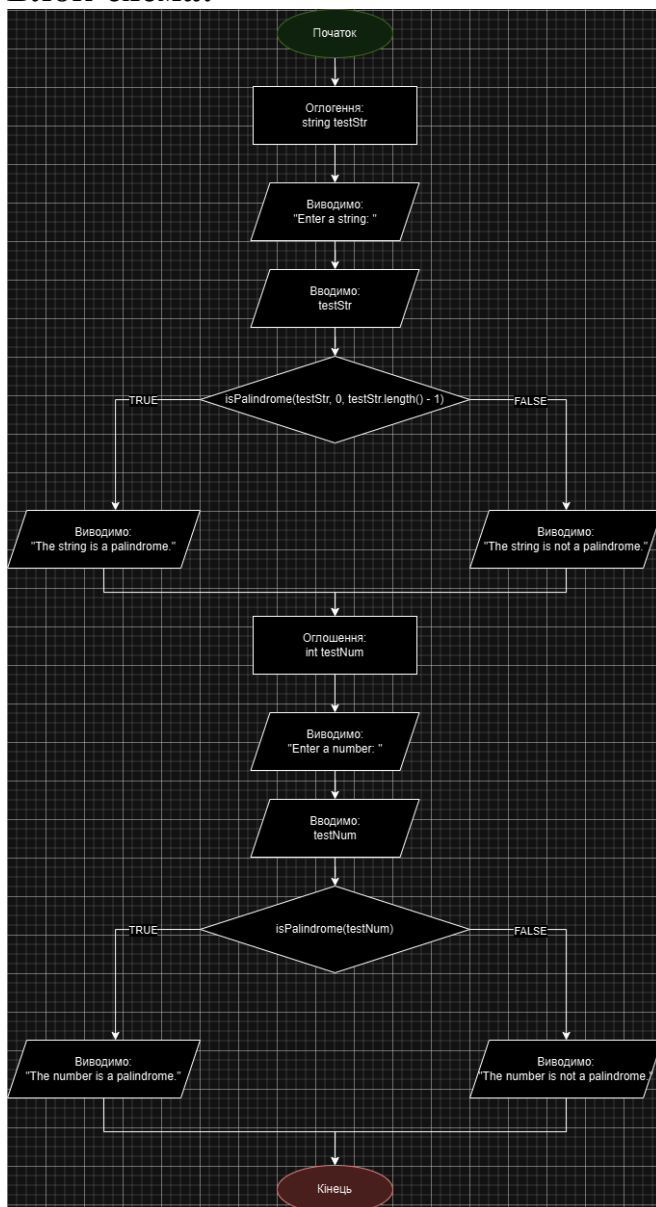
Витрачено часу: 1 година.

- Завдання №7: Practice# programming: Class Practice Task

Код:

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  // Рекурсивна функція для перевірки, чи є рядок паліндромом
7  bool isPalindrome(const string& str, int start, int end)
8  {
9      // Якщо початок перевищує кінець, рядок є паліндромом
10     if (start >= end)
11     {
12         return true;
13     }
14     // Перевірка символів на початку і в кінці
15     if (str[start] != str[end])
16     {
17         return false;
18     }
19
20     return isPalindrome(str, start + 1, end - 1);
21 }
22
23 // Перевантажена функція для перевірки, чи є ціле число паліндромом
24 bool isPalindrome(int number)
25 {
26     // Перетворення числа на рядок
27     string str = to_string(number);
28     return isPalindrome(str, 0, str.length() - 1);
29 }
30
31 int main()
32 {
33     string testStr;
34     cout << "Enter a string: ";
35     cin >> testStr;
36     if (isPalindrome(testStr, 0, testStr.length() - 1))
37     {
38         cout << "The string is a palindrome." << endl;
39     }
40     else
41     {
42         cout << "The string is not a palindrome." << endl;
43     }
44
45     int testNum;
46     cout << "Enter a number: ";
47     cin >> testNum;
48     if (isPalindrome(testNum))
49     {
50         cout << "The number is a palindrome." << endl;
51     }
52     else
53     {
54         cout << "The number is NOT a palindrome." << endl;
55     }
56
57     return 0;
58 }
```

## Блок-схема:



Очікувано часу: **1 година.**

Витрачено часу: **1 година.**



- Завдання №8: Practice# programming: Self Practice Task

Код:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int girlCount[4];
8      for (int i = 0; i < 4; i++)
9      {
10         cin >> girlCount[i];
11     }
12
13     int presentsCount = 0;
14     for (int i = 0; i < 4; i++)
15     {
16         presentsCount += girlCount[i];
17     }
18
19     cout << presentsCount << endl;
20
21     return 0;
22 }
23
```

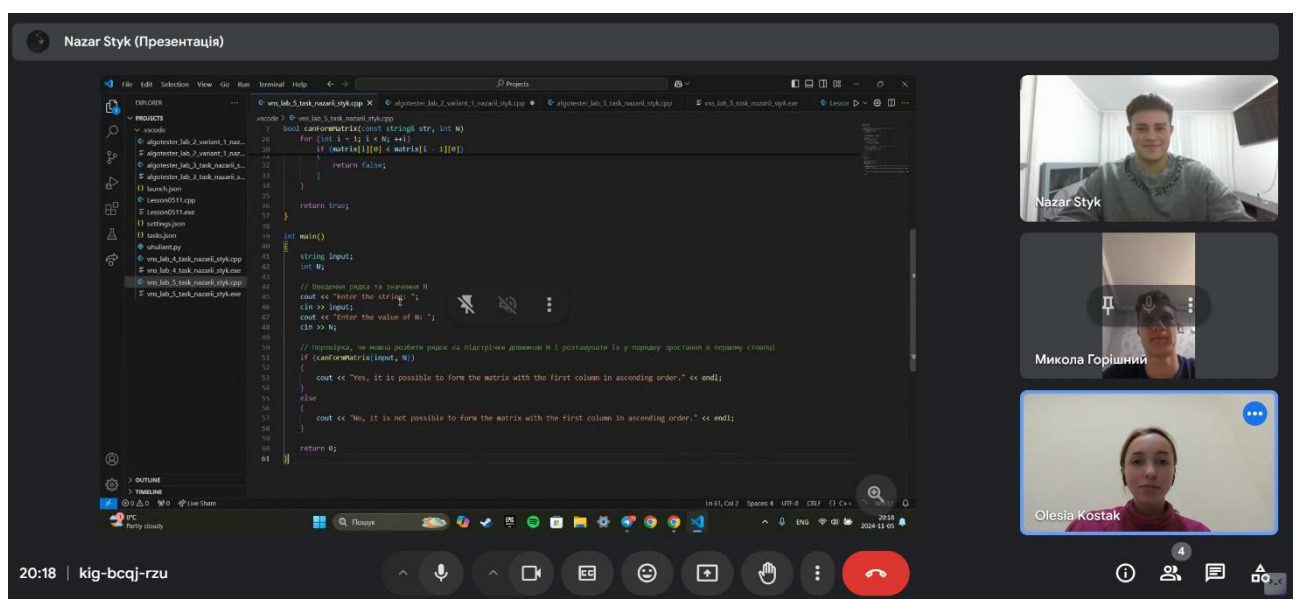
Завдання на алготестері:

<https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/40261>

Очікувано часу: 30 хвилин.

Витрачено часу: 10 хвилин.

Робота у команді:



## **Pull-Request:**

**Висновок:** У результаті виконання цієї роботи я засвоїв основи роботи з одновимірними та двовимірними масивами, а також динамічними структурами даних. Я набув практичних навичок використання вказівників та посилань для ефективного управління пам'яттю. Крім того, я дослідив особливості вкладених структур та алгоритмів обробки даних, що дозволяє створювати та організовувати складні типи даних. Отримані знання допоможуть мені оптимізувати програмний код та підвищити його продуктивність.