

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



## Звіт

**про виконання лабораторних та практичних робіт блоку №4**

На тему: «Програмування: алгоритм, програма, код. Системи числення. Двійкова система числення. Розробка та середовище розробки програми.»

**з дисципліни:** «Основи програмування»

до:

Практичних Робіт до блоку № 4

**Виконала:**

Студентка групи ШІ-13  
Бобринок Ангеліна Вадимівна

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

**Мета:** Розібратися з різними видами масивів, також навчитися використовувати вказівники та зрозуміти що таке структури і які вони бувають.

### **Теоретичні відомості:**

#### 1. Класи пам'яті у C++

- Статична пам'ять.
- Динамічна пам'ять.
- Поняття стеку.
- Виділення та вивільнення пам'яті.

#### 2. Вступ до Масивів і Вказівників:

- Основи масивів: визначення, важливість, приклади використання.
- Різниця між статичними та динамічними масивами.
- Основи вказівників: що це таке, як вони працюють.
- Взаємозв'язок між масивами та вказівниками.
- Вступ до посилань: основні концепції та відмінності від вказівників.

#### 3. Одновимірні Масиви:

- Створення та ініціалізація одновимірних масивів.
- Основні операції: індексація, присвоєння, читання.
- Цикли та обхід масивів.
- Використання функцій для роботи з масивами.
- Приклади алгоритмів сортування та пошуку.

#### 4. Вказівники та Посилання:

- Використання вказівників для доступу до елементів масиву.
- Арифметика вказівників.
- Різниця між вказівниками та посиланнями в контексті функцій.
- Динамічне виділення пам'яті з використанням вказівників.
- Використання вказівників для створення складних структур даних.

#### 5. Двовимірні Масиви:

- Оголошення та ініціалізація двовимірних масивів.
- Вкладені цикли для обходу двовимірних масивів.
- Практичні приклади використання двовимірних масивів.
- Передача двовимірних масивів у функції.
- Застосування двовимірних масивів для розв'язання задач.

#### 6. Динамічні Масиви:

- Основи динамічного виділення пам'яті.
- Створення та управління динамічними масивами.

- Використання операторів new та delete для управління пам'яттю.
  - Реалізація змінної розмірності масивів.
  - Передача динамічних масивів у функції.
7. Структури Даних:
- Оголошення та використання структур.
  - Використання масивів та вказівників у структурах.
  - Функції для обробки даних у структурах.
  - Використання структур для представлення складних даних.
  - Вкладені структури та їх використання.
  - Об'єднання (Union)
  - Переліки (enumerations)
8. Вкладені Структури:
- Поняття вкладених структур та їх оголошення.
  - Взаємодія з вкладеними структурами.
  - Використання вкладених структур для моделювання складних даних.
  - Передача вкладених структур у функції.
  - Приклади реального використання вкладених структур.
9. Використання структур
- Перевантаження операторів у структурі.
  - Вивід/ввід структури (operator<<);
  - Арифметичні операції з структурами (operator+, operator-);
  - Практичні задачі на виведення структур та операції з ними
10. Алгоритми обробки та робота з масивами та структурами:
- Алгоритми пошуку та сортування в масивах.
  - Обробка та маніпуляції з даними у структурах.
  - Використання циклів та умовних операторів для роботи з масивами та структурами.
  - Інтеграція масивів та структур у алгоритми.
  - Розв'язання практичних задач з використанням масивів та структур.

## **Виконання роботи:**

*1) Опрацювання завдання та вимог до програм та середовища:*

### **Завдання №1**

- 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити всі елементи, які дорівнюють 0.
- 4) Додати після першого парного елемента масиву елемент зі значенням  $M[I-1]+2$ .
- 5) Роздрукувати отриманий масив.

## Завдання №2

Написати процедуру для підсумовування матриць. З її допомогою скласти вихідну матрицю й транспоновану (тобто отриману поворотом вихідної на  $90^\circ$ ).

## Завдання №3

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 11 перелетить у клітинку з індексом  $a_{11}$ , 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:

Якщо 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  $a_i$  та  $a_{i+1}$  - виведіть **Stopped**

Врахуйте, що перевіряти треба також до взльоту.

## Завдання №4

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

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

Мале Бісеня любить гострити зуби. А Зла Тітонька любить до нього підходити і питатися: «Що, зуби гостриш?». Бісеняті таке не дуже подобається, тому воно придумало робити таке.

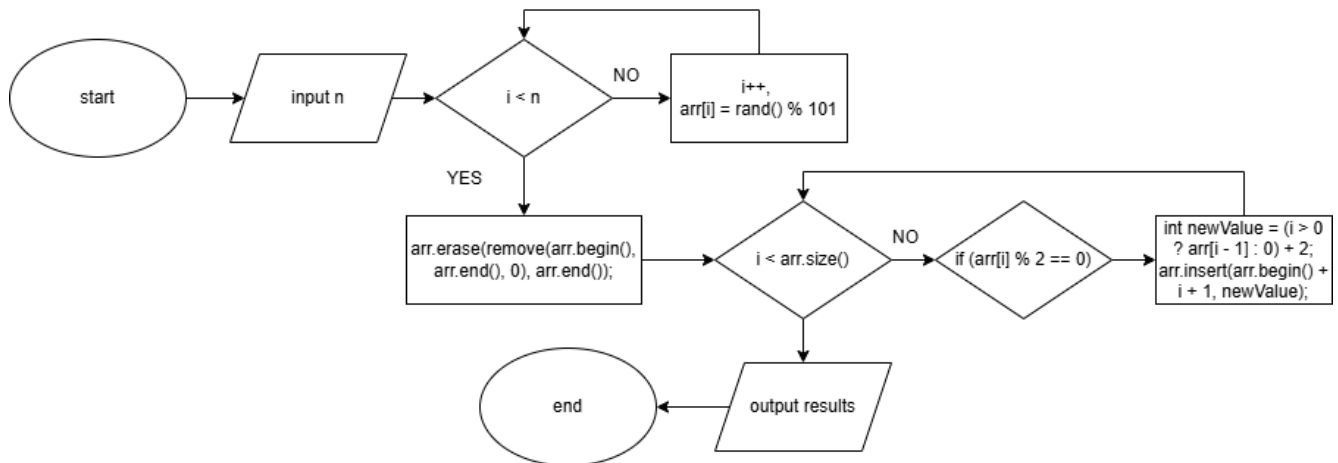
У Малого Бісеняти є  $n$  зубів. Кожен зуб має коефіцієнт загостреності  $a_i$ . Також існує межа загостреності  $k$ . Якщо коефіцієнт загостреності певного зуба є більшим чи рівним межі загостреності, то такий зуб вважається загостреним.

Мале Бісеня хоче наступного разу, коли Зла Тітонька його щось запитас, показати їй якнайбільше загострених зубів, що розташовані поспіль.

Допоможіть Малому Бісеняті дізнатися, скільки найбільше зубів воно зможе показати.

2) Дизайн та планована оцінка часу виконання завдань:

### Завдання №1

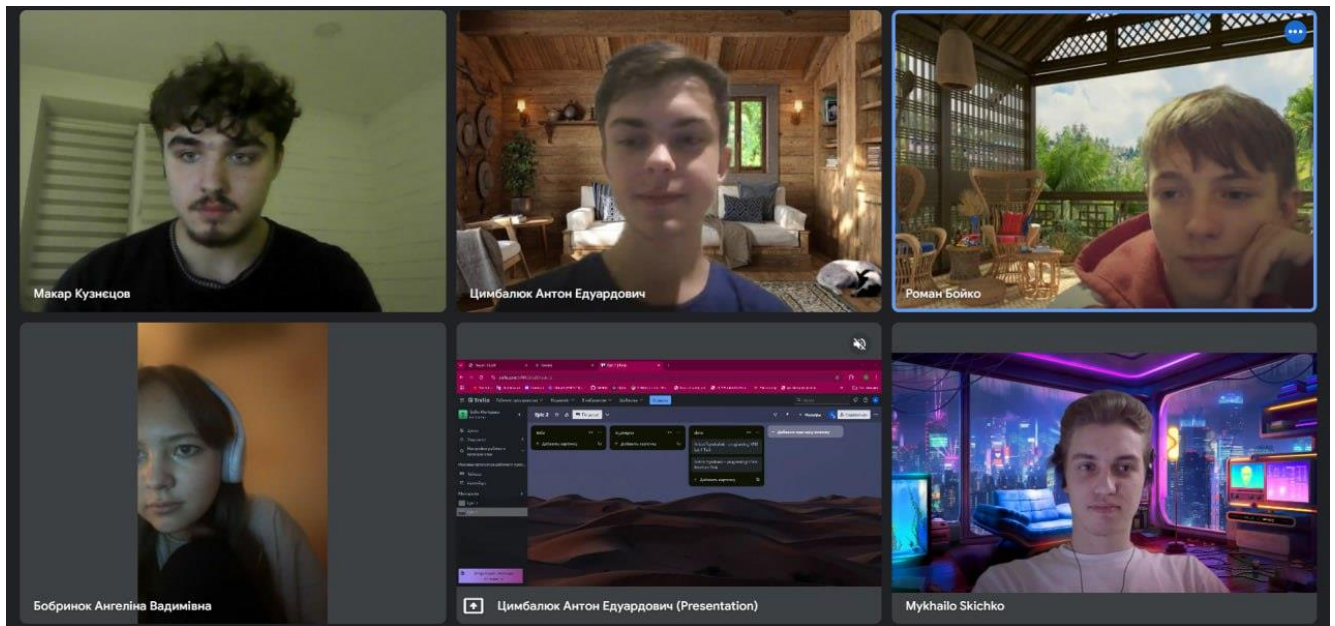


3) Конфігурація середовища до виконання завдань:

### Планування роботи в notion

	</> Task	Status	deadline
Anhelina	Theory Education Activities	In progress	22/11/2024
Anhelina	Requirements management (understand tasks) and design activities	Done	22/11/2024
Anhelina	Lab4 programming: VNS Lab 4	Done	22/11/2024
Anhelina	Lab4 programming: VNS Lab 5	Pending	22/11/2024
Anhelina	Lab4 programming: Algotester Lab 2	Pending	22/11/2024
Anhelina	Lab4 programming: Algotester Lab 3	Pending	22/11/2024
Anhelina	Practice4 programming: Class Practice Task	Pending	22/11/2024
Anhelina	Practice4 programming: Self Practice Task	Pending	22/11/2024
Anhelina	Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)	In progress	22/11/2024
Anhelina	Results Evaluation and Release	Pending	22/11/2024

Зустріч з командою та обговорення питань



4) Код програм з посиланням на зовнішні ресурси:

### Завдання №1

vns\_lab\_4\_anhelina\_bobrynok.cpp

```

1  #include <iostream>
2  #include <vector>
3  #include <cstdlib>
4  #include <ctime>
5  #include <algorithm>
6
7  using namespace std;
8
9  int main()
10 {
11     srand(time(0));
12
13     int n;
14     cout << "Enter size of array: ";
15     cin >> n;
16
17     vector<int> arr(n);
18
19     for (int i = 0; i < n; i++) {
20         arr[i] = rand() % 101;
21     }
22
23     cout << "Initial array: ";
24     for (int i = 0; i < n; i++) {
25         cout << arr[i] << " ";
26     }
27     cout << endl;
28
29     arr.erase(std::remove(arr.begin(), arr.end(), 0), arr.end());
30

```

```

31     cout << "Array after removal: ";
32     for (int num : arr) {
33         cout << num << " ";
34     }
35     cout << endl;
36
37     for (size_t i = 0; i < arr.size(); i++) {
38         if (arr[i] % 2 == 0) {
39             int newValue = (i > 0 ? arr[i - 1] : 0) + 2;
40             arr.insert(arr.begin() + i + 1, newValue);
41             break;
42         }
43     }
44
45     cout << "Array after insertion: ";
46     for (int num : arr) {
47         cout << num << " ";
48     }
49     cout << endl;
50
51     return 0;
52 }

```

## Завдання №2

vns\_lab\_5\_anhelina\_bobrynok.cpp

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  vector<vector<int>> sumMatrices(const vector<vector<int>>& mat1, const vector<vector<int>>& mat2) {
7      int rows = mat1.size();
8      int cols = mat1[0].size();
9      vector<vector<int>> result(rows, vector<int>(cols));
10
11      for (int i = 0; i < rows; ++i) {
12          for (int j = 0; j < cols; ++j) {
13              result[i][j] = mat1[i][j] + mat2[i][j];
14          }
15      }
16      return result;
17  }
18
19  vector<vector<int>> transposeMatrix(const vector<vector<int>>& mat) {
20      int rows = mat.size();
21      int cols = mat[0].size();
22      vector<vector<int>> transposed(cols, vector<int>(rows));
23
24      for (int i = 0; i < rows; ++i) {
25          for (int j = 0; j < cols; ++j) {
26              transposed[j][i] = mat[i][j];
27          }
28      }
29      return transposed;
30  }

```

```

31
32 void printMatrix(const vector<vector<int>>& mat) {
33     for (const auto& row : mat) {
34         for (const auto& elem : row) {
35             cout << elem << " ";
36         }
37         cout << endl;
38     }
39 }
40
41 int main() {
42
43     vector<vector<int>> mat1 = {{1, 3, 5}, {7, 9, 11}, {13, 15, 17}};
44     vector<vector<int>> mat2 = {{2, 4, 6}, {8, 10, 12}, {14, 16, 18}};
45
46     cout << "Matrix 1:" << endl;
47     printMatrix(mat1);
48     cout << "Matrix 2:" << endl;
49     printMatrix(mat2);
50
51     vector<vector<int>> summedMatrix = sumMatrices(mat1, mat2);
52     cout << "Summed Matrix:" << endl;
53     printMatrix(summedMatrix);
54
55     vector<vector<int>> transposedMatrix = transposeMatrix(summedMatrix);
56     cout << "Transposed Summed Matrix:" << endl;
57     printMatrix(transposedMatrix);
58
59     return 0;

```

### Завдання №3

algotester\_lab\_2\_anhelina\_bobrynok.cpp

```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6      int N;
7      cin >> N; // Розмір масиву
8      vector<int> arr(N + 1); // Масив (індексуємо з 1 для зручності)
9
10     for (int i = 1; i <= N; ++i) {
11         cin >> arr[i];
12     }
13
14     int left = 1; // Початкова позиція лівого дрона
15     int right = N; // Початкова позиція правого дрона
16
17     // Симуляція польоту дронів
18     while (true) {
19         // Перевірка перед польотом
20         if (left == right) {
21             cout << left << " " << right << endl;
22             cout << "Collision" << endl;
23             return 0;
24         }
25         if (left + 1 == right) {
26             cout << left << " " << right << endl;
27             cout << "Stopped" << endl;
28             return 0;
29         }
30     }

```



```

31     // Оновлення позицій
32     left += arr[left];
33     right -= arr[right];
34
35     // Перевірка після польоту
36     if (left > right) {
37         cout << left << " " << right << endl;
38         cout << "Miss" << endl;
39         return 0;
40     }
41 }
42
43 return 0;
44 }

```

## Завдання №4

algotester\_lab\_3\_anhelina\_bobrynok .cpp

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main() {
7      int N, M;
8      cin >> N;
9
10     vector<int> a(N);
11     for (int i = 0; i < N; i++) {
12         cin >> a[i];
13     }
14
15     cin >> M;
16
17     vector<int> b(M);
18     for (int i = 0; i < M; i++) {
19         cin >> b[i];
20     }
21
22     int same = 0;
23     for (int i = 0; i < N; i++) {
24         for (int j = 0; j < M; j++) {
25             if (a[i] == b[j]) {
26                 same++;
27                 break;
28             }
29         }
30     }

```

```

31
32     vector<int> unique;
33     for (int i = 0; i < N; i++) {
34         bool isUnique = true;
35         for (int j = 0; j < unique.size(); j++) {
36             if (a[i] == unique[j]) {
37                 isUnique = false;
38                 break;
39             }
40         }
41         if (isUnique) {
42             unique.push_back(a[i]);
43         }
44     }
45
46     for (int i = 0; i < M; i++) {
47         bool isUnique = true;
48         for (int j = 0; j < unique.size(); j++) {
49             if (b[i] == unique[j]) {
50                 isUnique = false;
51                 break;
52             }
53         }
54         if (isUnique) {
55             unique.push_back(b[i]);
56         }
57     }
58
59     cout << same << endl;
60     cout << unique.size() << endl;
61
62     return 0;
63 }

```

## Завдання №5

practice\_work\_anhelina\_bobrynok.cpp

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  // Рекурсивна функція для перевірки паліндрома для рядків
6  bool isPalindrome(const string& str, int start, int end) {
7
8      if (start >= end) {
9          return true;
10     }
11
12     if (str[start] != str[end]) {
13         return false;
14     }
15
16     return isPalindrome(str, start + 1, end - 1);
17 }
18
19 // Перевантажена функція для перевірки паліндрома для цілих чисел
20 bool isPalindrome(int num) {
21     if (num < 0) {
22         return false;
23     }
24
25     string str = to_string(num);
26
27     return isPalindrome(str, 0, str.length() - 1);
28 }
29

```

```

30  int main() {
31      string word;
32      int number;
33
34      cout<<"enter a word\n";
35      cin>>word;
36      cout<<"enter a number\n";
37      cin>>number;
38
39      if (isPalindrome(word, 0, word.length() - 1)) {
40          cout << word << " is a palindrome!" << endl;
41      } else {
42          cout << word << " is not a palindrome!" << endl;
43      }
44      if (isPalindrome(number)) {
45          cout << number << " is a palindrome!" << endl;
46      } else {
47          cout << number << " is not a palindrome!" << endl;
48      }
49
50      return 0;
51  }

```

## Завдання №6

self\_practice\_work\_anhelina\_bobrynok.cpp

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5
6  int main() {
7      int n, k; //кількість зубів та межа загостреності відповідно
8      cin >> n >> k;
9
10     vector<int> a(n);
11     for (int i = 0; i < n; i++) {
12         cin >> a[i];
13     }
14
15     int max_count = 0;
16     int current_count = 0;
17
18     for (int i = 0; i < n; i++) {
19         if (a[i] >= k) {
20             current_count++;
21         } else {
22             max_count = max(max_count, current_count);
23             current_count = 0;
24         }
25     }
26
27     max_count = max(max_count, current_count);
28
29     cout << max_count << endl;
30

```

5)Результати виконання завдань, тестування та фактично затрачений час:

## Завдання №1

```
Enter size of array: 5
Initial array: 17 40 38 63 35
Array after removal: 17 40 38 63 35
Array after insertion: 17 40 19 38 63 35
```

## Завдання №2

```
Matrix 1:
1 3 5
7 9 11
13 15 17
Matrix 2:
2 4 6
8 10 12
14 16 18
Summed Matrix:
3 7 11
15 19 23
27 31 35
Transposed Summed Matrix:
3 15 27
7 19 31
11 23 35
```

## Завдання №3

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.002	1.207	<a href="#">Перегляд</a>

## Завдання №4

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.003	1.422	<a href="#">Перегляд</a>

## Завдання №5

```
enter a word
word
enter a number
1221
word is not a palindrome!
1221 is a palindrome!
```

## Завдання №6

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.043	1.473	<a href="#">Перегляд</a>

**Висновки:** Завдяки цій лабораторній роботі я мала змогу розібратися з різними видами масивів, також навчитися використовувати вказівники та зрозуміти що таке структури і які вони бувають.