

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 ШІ-11  
Фарина Арсеній Петрович

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:

Do afterwards.

## Теоретичні відомості:

- Тема №1: Класи пам'яті в C++.
- Тема №2: Вступ до Масивів і Вказівників.
- Тема №3: Одновимірні Масиви.
- Тема №4: Вказівники та Посилання.
- Тема №5: Двовимірні Масиви.
- Тема №6: Динамічні Масиви.
- Тема №7: Структури Даних.
- Тема №8: Вкладені Структури.
- Тема №9: Використання Структур.
- Тема №10: Алгоритми обробки та робота з Масивами та

структурами.

### 1) Індивідуальний план опрацювання теорії:

- Тема №1: Класи пам'яті в C++:
  - o Джерела інформації:
    - Статті.  
<http://cpp.dp.ua/klasy-pam-yati-u-c-builder/>
  - Що опрацьовано:
    - o Статична пам'ять.
    - o Динамічна пам'ять.
    - o Поняття стеку.
    - o Виділення та вивільнення пам'яті.
- Запланований час на вивчення 30 хвилин.  
Витрачений час 30 хвилин.
- Тема №2: Вступ до Масивів і Вказівників:
  - o Джерела інформації:
    - Статті.  
<https://www.youtube.com/watch?v=ULdbOaMBPYc&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=40>  
<https://www.youtube.com/watch?v=zopWRIYOWw&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=57>
  - Що опрацьовано:
    - o Основи масивів: визначення, важливість, приклади використання.
    - o Різниця між статичними та динамічними масивами.
    - o Основи вказівників: що це таке, як вони працюють.
    - o Взаємозв'язок між масивами та вказівниками.
    - o Вступ до посилань: основні концепції та відмінності від вказівників.
- Запланований час на вивчення 1 година.  
Витрачений час 1 година.
- Тема №3: Одновимірні масиви:
  - o Джерела інформації:
    - Статті.

<https://www.youtube.com/watch?v=ULdbOaMBPYc&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=40>

- Що опрацьовано
    - 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.
- Запланований час на вивчення 1 година.  
Витрачений час 1 година.

- Тема №4: Вказівники та Посилання:

- Джерела інформації:
  - Статті.

<https://www.youtube.com/watch?v=ULdbOaMBPYc&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=40>

- Що опрацьовано
    - 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.
- Запланований час на вивчення 2 години.  
Витрачений час 2 години.

- Тема № 5: Двовимірні Масиви:

- Джерела інформації:
  - Статті.

<https://www.youtube.com/watch?v=hcYgFCgeZzQ>

- Що опрацьовано
    - 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.
- Запланований час на вивчення 1 година.  
Витрачений час 1 година..

- Тема №6: Динамічні Масиви:

- Джерела інформації:
  - Статті.

<https://www.youtube.com/watch?v=OGR9VJEh8Hk&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=60>

[https://www.youtube.com/watch?v=\\_N3zkbnCTw0](https://www.youtube.com/watch?v=_N3zkbnCTw0)

- Що опрацьовано
    - 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.
- Запланований час на вивчення 1 година.  
Витрачений час 1 година.

- Тема №7: Структури Даних:

○ Джерела інформації:

- Статті.

[https://www.youtube.com/watch?v=999IE-6b7\\_s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=63](https://www.youtube.com/watch?v=999IE-6b7_s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=63)

[https://www.youtube.com/watch?v=D79J\\_q5SGzc&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=64](https://www.youtube.com/watch?v=D79J_q5SGzc&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=64)

<https://www.youtube.com/watch?v=ifdk1fYSk1Y&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=65>

- Що опрацьовано

- 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)

Запланований час на вивчення 2.5 години.

Витрачений час 2.5 години.

- Тема №8: Вкладені Структури:

○ Джерела інформації:

- Статті.

[https://www.youtube.com/watch?v=999IE-6b7\\_s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=63](https://www.youtube.com/watch?v=999IE-6b7_s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=63)

- Що опрацьовано

- 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.

Запланований час на вивчення 1 година.

Витрачений час 1 година.

- Тема №9: Використання Структур:

○ Джерела інформації:

- Статті.

[https://www.youtube.com/watch?v=999IE-6b7\\_s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=63](https://www.youtube.com/watch?v=999IE-6b7_s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=63)

- Що опрацьовано

- Перевантаження операторів у структурі.
- Вивід/ввід структури (operator<<);
- Арифметичні операції з структурами (operator+, operator-);
- Практичні задачі на виведення структур та операції з ними

Запланований час на вивчення 1 година.

Витрачений час 1 година.

- Тема №10: Алгоритми обробки та робота з Масивами та структурами:

○ Джерела інформації:

- Статті.

<https://www.youtube.com/watch?v=uQxG9gBROog&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=51>

<https://www.youtube.com/watch?v=maB87eyn7h8&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=52>

- Що опрацьовано

- 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
- Запланований час на вивчення 1.5 години.  
Витрачений час 1.5 години.

## **Виконання роботи:**

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

Завдання №1

#### **VNS LAB 4 – TASK 1 (VARIANT 2)**

- 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити перший елемент, який дорівнює 0.
- 4) Додати після кожного парного елемента масиву елемент зі значенням  $M[I-1]+2$ .
- 5) Роздрукувати отриманий масив.

Завдання №2

#### **VNS LAB 5 – TASK 1 (VARIANT 2)**

Написати функцію для обміну рядків двовимірної масиви з її допомогою відсортувати масив по елементах третього стовпця.

Завдання №3

#### **ALGOTESTER LAB 2 (VARIANT 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 елемента. Ви хочете мінімізувати втому, але все що ви можете зробити - викинути одне число з дороги, тобто забрати його з масиву.

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

Завдання №4

#### **ALGOTESTER LAB 3 (VARIANT 3)**

Вам дана стрічка  $s$ . Ваше завдання зробити компресію стрічки, тобто якщо якась буква йде більше одного разу підряд у стрічці замінити її на букву + кількість входжень підряд.

Завдання №5

#### **CLASS PRACTICE WORK**

Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.

#### **Вимоги**

1. Визначення функції:

1. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:
  1. *bool isPalindrome(const string& str, int start, int end);*
3. Перевантаження функцій:
  1. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
  2. *bool isPalindrome(ціле число);*
4. Рекурсія:
  1. Рекурсивна функція для рядків перевірить символи в поточній початковій і кінцевій позиціях. Якщо вони збігаються, він буде рекурсивно перевіряти наступні позиції, поки початок.

## Завдання №6

### SELF PRACTICE WORK ALGOTESTER

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 запитає, показати їй якнайбільше загострених зубів, що розташовані поспіль.

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

## Вхідні дані

У першому рядку задані два цілих числа  $n$  та  $k$  — кількість зубів та межа загостреності відповідно.

В другому рядку задано  $n$  цілих чисел  $a_i$  — коефіцієнти загостреності зубів.

## Вихідні дані

Єдине ціле число — відповідь на задачу.

## 2. Дизайн та планувальна оцінка часу виконання завдань:

### Програма №1

- Важливі деталі для реалізації програми.  
Використати одновимірний масив цілих чисел для реалізації, використати при цьому генератор цілих чисел, за допомогою циклу знищити перший елемент, який дорівнює 0, після кожного парного елемента масиву додати  $M[I - 1] + 2$ , і роздрукувати отриманий масив.
- Плановий час на реалізацію 1.5 години.

### Програма №2

- Важливі деталі для реалізації програми.  
Використовуючи функції, розв'язати зазначене у варіанті завдання. Масив

повинен передаватися у функцію як параметр. Використати двовимірний масив.

- Плановий час на реалізацію 2 години.

### **Програма №3**

- Важливі деталі для реалізації програми.

Використати динамічний масив для реалізації, і за допомогою функцій бібліотеки <climits> знайти найбільше і найменше значення, а після цього на екран вивести їхню різницю.

Плановий час на реалізацію 3 години.

### **Програма №4**

- Важливі деталі для реалізації програми.

Використовувати бібліотеку <string> і її функції, щоб вирішити завдання.

- Плановий час на реалізацію 2 години.

### **Програма №5**

- Блок – схема.

- Важливі деталі для реалізації програми.

Визначити та реалізувати рекурсивну функцію isPalindrome для рядків.

Визначити та реалізувати перевантажену функцію isPalindrome для цілих чисел. Використати математичний підхід щоб перевірити чи число є паліндромом.

- Плановий час на реалізацію 3 години.

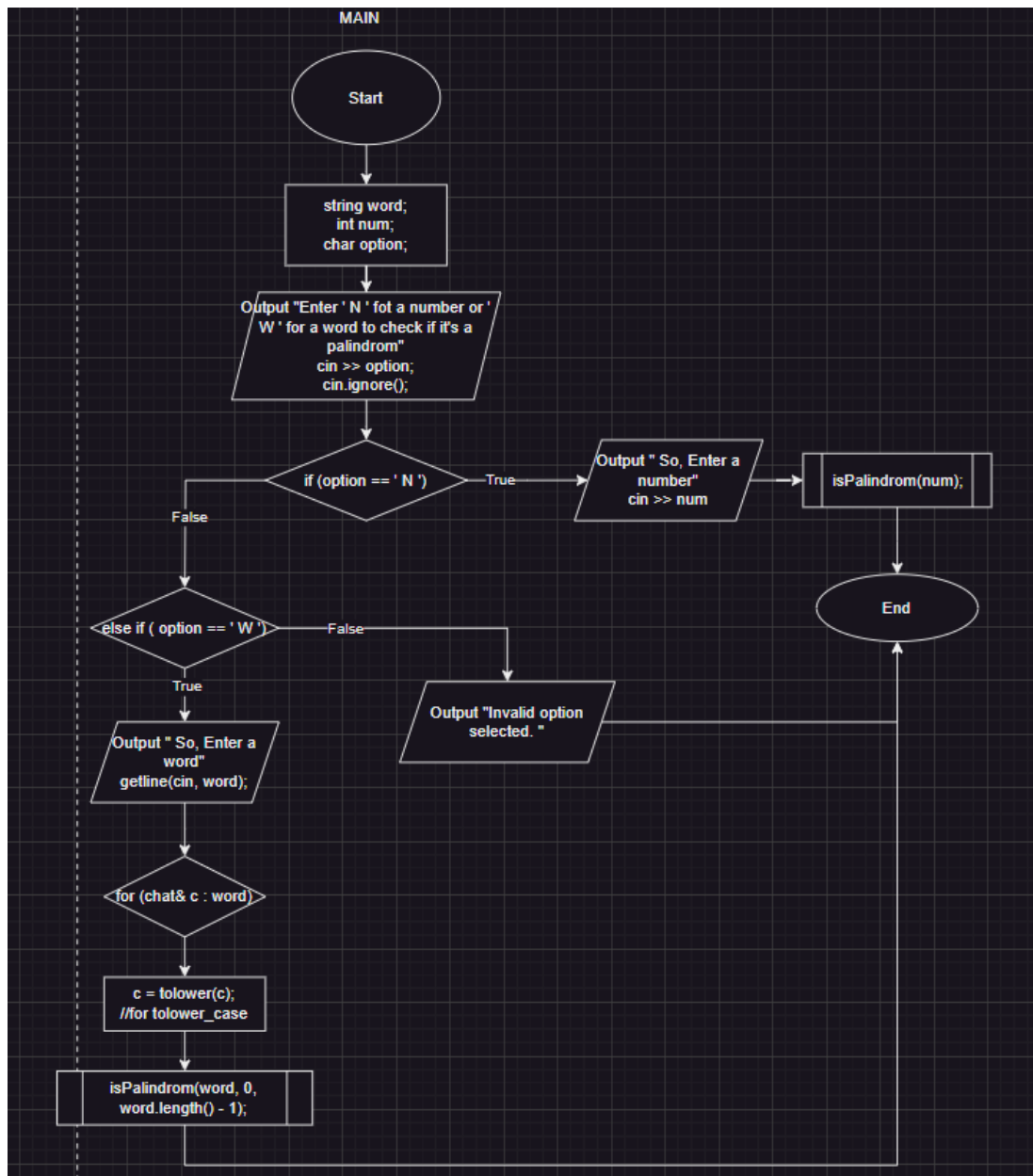


Рисунок 2.1. Блок-схема до функції main

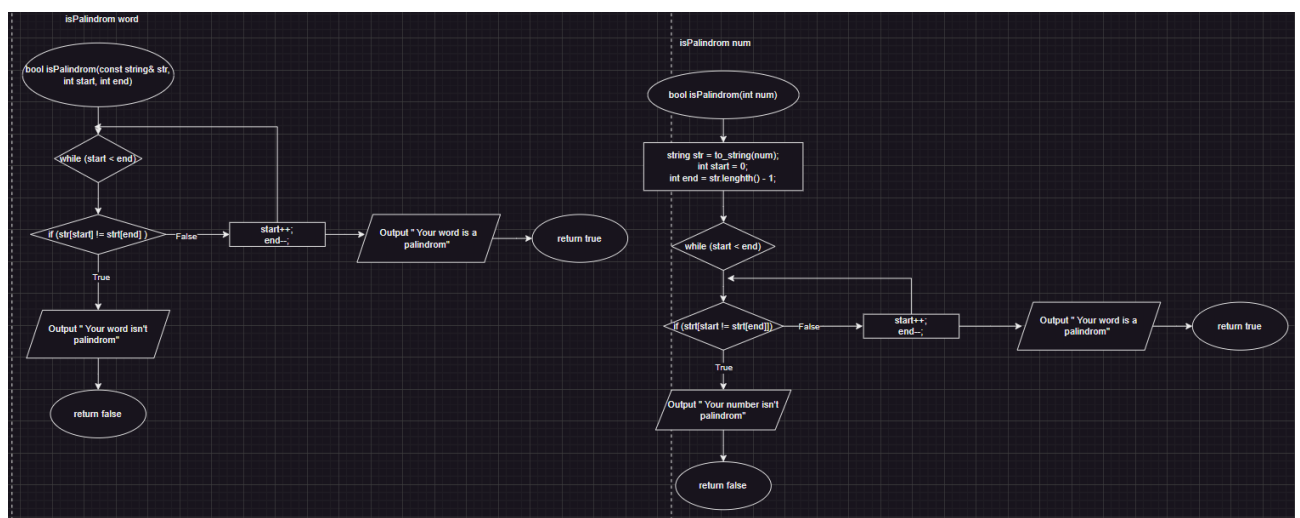


Рисунок 2.2. Блок-схема до функцій які викликаю у main

## Програма №6

- Важливі деталі для реалізації програми.



Використовуючи вектор записати “рівень загострення зуба”, а потім за допомогою циклу реалізувати знаходження в масиві елементів, які більші або рівні за коефіцієнт загострення.

- Плановий час на реалізацію 2 години.

### 3. Код програм з посиланням на зовнішні ресурси та фактично затрачений час:

#### Завдання №1

```
1  #include <iostream>
2
3  using namespace std;
4
5  //prototype of function
6  void filled_array(int* arr[], int& user_input);
7  void print_array(const int* arr[], int& user_input);
8  void delete_first_element(int* arr[], int& user_input);
9  void add_element(int* arr[], int& user_input);
10
11 void filled_array(int arr[], int user_input) {
12     for (int i = 0; i < user_input; i++){
13         arr[i] = rand() % 100;
14     }
15 }
16
17 void print_array(const int arr[], int user_input){
18     for (int i = 0; i < user_input; i++){
19         cout << arr[i] << " ";
20     }
21     cout << endl;
22 }
23
24 void delete_first_element(int arr[], int user_input){
25     for (int i = 0; i < user_input - 1; i++){
26         arr[i] = arr[i + 1];
27     }
28     arr[user_input - 1] = 0; //for deleted element which will be replaced in the end of massive
29     user_input--;
30 }
31
32 void add_element(int arr[], int user_input) {
33     const int MAX_SIZE = 100;
34
35     for (int i = 0; i < user_input; i++) {
36         if (arr[i] % 2 == 0 && user_input < MAX_SIZE) {
37             for (int k = user_input; k > i + 1; k--) {
38                 arr[k] = arr[k - 1];
39             }
40             arr[i + 1] = arr[i] + 2;
41             user_input++;
42             i++;
43         }
44     }
45 }
46
47 int main() {
48
49     srand(static_cast<unsigned> (time(nullptr)));
50
51     int user_input;
52     const int SIZE = 100;
53     int arr[SIZE];
54     cout << "Enter the array length: ";
55     cin >> user_input;
56
57     if (user_input > SIZE){
58         cout << "Error! Maximum allowed input is: " << SIZE << endl;
59         return -1;
60     }
61
62     filled_array(arr, user_input);
63     cout << "Array filled with random element: " << endl;
64     print_array(arr, user_input);
65
66     delete_first_element(arr, user_input);
67     cout << "Array after deleting first element: " << endl;
68     print_array(arr, user_input);
69
70     add_element(arr, user_input);
71     cout << "Array after adding to pair element in massive " << endl;
72     print_array(arr, user_input);
73
74     return 0;
75 }
76
```

Рисунок 3.1. Код до програми № 1

```
Enter the array length: 5
Array filled with random element:
36 54 27 49 23
Array after deleting first element:
54 27 49 23 0
Array after adding to pair element in massive
54 56 27 49 23
```

Рисунок 3.2. Приклад виконання програми № 1

Спочатку створюємо прототипи функцій, а нижче їх реалізовуємо. За допомогою функції `srand` будемо заповнювати випадковими числами наш масив. У головній функції (`main`) будемо їх викликати і робити відповідні перетворення з масивом. Фактично затрачений час 2 години.

### Посилання на файл у пулл реквесті

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/219](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/219)

### Завдання №2

```
1  #include <iostream>
2  #include <ctime>
3
4  using namespace std;
5
6  //prototype of function
7  void exchange_rows(int arr[5][5]);
8  void print_matrix(int arr[5][5]);
9  void exchange_3rd_row(int arr[5][5]);
10 void filled_matrix(int arr[5][5]);
11
12 //for swapping
13 void exchange_rows(int arr[5][5], int row1, int row2){
14     for(int col = 0; col < 5; col++){
15         int temp = arr[row1][col];
16         arr[row1][col] = arr[row2][col];
17         arr[row2][col] = temp;
18     }
19 }
20
21 //for swapping 3rd column
22 void exchange_by_3rd_row(int arr[5][5]){
23     for (int i = 0; i < 4; i++){
24         for (int j = i + 1; j < 5; j++){
25             if (arr[i][2] > arr[j][2]){
26                 exchange_rows(arr, i, j);
27             }
28         }
29     }
30 }
31
32 //for filled matrix with random numbers from 0 to 10
33 void filled_matrix(int arr[5][5]){
34     for (int i = 0; i < 5; i++){
35         for (int j = 0; j < 5; j++) {
36             arr[i][j] = rand() % 10;
37         }
38     }
39 }
40
41 void print_matrix(int arr[5][5]){
42     for (int i = 0; i < 5; i++){
43         for (int j = 0; j < 5; j++){
44             cout << arr[i][j] << " ";
45         }
46         cout << endl;
47     }
48 }
49
50 int main (){
51
52     srand(static_cast<unsigned> (time(nullptr)));
53
54     int arr [5][5];
55
56     cout << "Generated matrix 5x5 is: " << endl;
57     filled_matrix(arr);
58     print_matrix(arr);
59
60     cout << "Swapped matrix is: " << endl;
61     exchange_rows(arr, 0, 2);
62     print_matrix(arr);
63
64     cout << "Swapped by 3rd column: " << endl;
65     exchange_by_3rd_row(arr);
66     print_matrix(arr);
67
68     return 0;
69 }
70
```

Рисунок 3.3. Код до програми № 2

```
Generated matrix 5x5 is:
3 2 8 0 9
1 6 4 4 3
2 3 3 4 8
7 4 1 7 5
4 1 8 3 7
Swapped matrix is:
2 3 3 4 8
1 6 4 4 3
3 2 8 0 9
7 4 1 7 5
4 1 8 3 7
Swapped by 3rd column:
7 4 1 7 5
2 3 3 4 8
1 6 4 4 3
3 2 8 0 9
4 1 8 3 7
```

Рисунок 3.4. Приклад виконання програми № 2

Тут теж створюємо прототипи функцій і їх реалізовуємо відповідно до умови завдання. У цій програмі уже будемо створювати двовимірний масив, або матрицю і будемо відсортовувати за елементами 3 стовпця.

Фактично затрачений час 2 години.

**Посилання на файл у пулл реквесті**

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/219](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/219)

## Завдання №3

```
1 #include <iostream>
2 #include <climits>
3
4 using namespace std;
5
6 int find_min_tiredness(int arr[], int N) {
7     if (N <= 1) {
8         return 0;
9     }
10
11     int min_val = INT_MAX;
12     int max_val = INT_MIN;
13     for (int i = 0; i < N; i++){
14         min_val = min(min_val, arr[i]);
15         max_val = max(max_val, arr[i]);
16     }
17
18     int initial_tiredness = max_val - min_val;
19
20     int min_tiredness = initial_tiredness;
21     for (int i = 0; i < N; i++) {
22         if (arr[i] == min_val || arr[i] == max_val) {
23             int new_min = INT_MAX;
24             int new_max = INT_MIN;
25             for (int k = 0; k < N; k++){
26                 if (k != i){
27                     new_min = min(new_min, arr[k]);
28                     new_max = max(new_max, arr[k]);
29                 }
30             }
31             min_tiredness = min(min_tiredness, new_max - new_min);
32         }
33     }
34     return min_tiredness;
35 }
36
37 int main() {
38     int N;
39     cin >> N;
40
41     int* arr = new int[N];
42     for (int i = 0; i < N; i++) {
43         cin >> arr[i];
44     }
45
46
47     int result = find_min_tiredness(arr, N);
48     cout << result << endl;
49
50     delete[] arr;
51     return 0;
52 }
```

Рисунок 3.5. Код до програми №3

```
5
1 2 2 4 4
2
```

Рисунок 3.6. Приклад виконання програми №3

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
a few seconds ago	C++ 23	Accepted	0.003	1.188	<a href="#">View</a>

Рисунок 3.7. Статус задачі на алготестері

Нам потрібно знайти мінімальну втому, тобто різницю між найменшим та найбільшим значенням у масиві, у якому ми по черзі викидаємо кожен елемент. За допомогою циклу який умовно викидає кожен елемент будемо шукати найменше і найбільше значення і їхню різницю.

Фактично затрачений час 3.5 години.

## Посилання на файл у пулл реквесті

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/219](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/219)

### Завдання №4

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  string compress_string(const string& s) {
7
8      string s_compressed;
9
10     int count = 1;
11
12     for (int i = 1; i < s.length(); i++) {
13         if (s[i] == s[i - 1]) {
14             count++;
15         } else {
16             s_compressed += s[i - 1];
17             if (count > 1) {
18                 s_compressed += to_string(count);
19             }
20             count = 1;
21         }
22     }
23
24     s_compressed += s[s.length() - 1];
25     if (count > 1){
26         s_compressed += to_string(count);
27     }
28
29     return s_compressed;
30 }
31
32 int main() {
33     string s;
34
35     getline(cin, s);
36
37     string result = compress_string(s);
38
39     cout << result << endl;
40
41     return 0;
42 }
43
```

Рисунок 3.8. Код до програми №4

AAAABBBBCCDDEE  
A4B4C3D2E2

Рисунок 3.9. Приклад виконання програми №4

3 hours ago	C++ 23	Accepted	0.003	1.195	<a href="#">View</a>
-------------	--------	----------	-------	-------	----------------------

Рисунок 3.10. Статус задачі на алготестері

Нам потрібно записати у числах скільки саме тих букв є у стрічці. Тут за допомогою циклу будемо проходитися по нашій стрічці і порівнювати елемент з індексом 1 і елемент з індексом  $[i - 1]$ . Якщо в якийсь момент вони не рівні ми будемо скидувати наш лічильник і шукати іншу групу таких елементів і відповідно виводити на екран їх.

Фактично затрачений час 2.5 години.

**Посилання на файл у пулл реквесті**

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/219](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/219)

## Завдання №5

```
1  #include <iostream>
2  #include <string>
3  #include <cctype> // for tolower
4
5  using namespace std;
6
7  // function for check if word is palindrom
8  bool isPalindrom(const string& str, int start, int end) {
9      while (start < end) {
10         if (str[start] != str[end]) {
11             cout << "Your word isn't a palindrome." << endl;
12             return false;
13         }
14         start++;
15         end--;
16     }
17     cout << "Your word is a palindrome!" << endl;
18     return true;
19 }
20
21 // function for check if number is palindrom
22 bool isPalindrom(int num) {
23
24     string str = to_string(num); // Convert number to string
25     int start = 0;
26     int end = str.length() - 1;
27
28     while (start < end) {
29         if (str[start] != str[end]) {
30             cout << "Your number isn't a palindrome." << endl;
31             return false;
32         }
33         start++;
34         end--;
35     }
36     cout << "Your number is a palindrome!" << endl;
37     return true;
38 }
39
40 int main() {
41
42     string word;
43     int num;
44     char option;
45
46     cout << "Enter 'N' for a number or 'W' for a word to check if it's a palindrome:" << endl;
47     cin >> option;
48     cin.ignore();
49
50     if (option == 'N') {
51         cout << "So, Enter a number: ";
52         cin >> num;
53         isPalindrom(num);
54     } else if (option == 'W') {
55         cout << "So, Enter a word: ";
56         getline(cin, word);
57
58         for (char& c : word){
59             c = tolower(c);
60         }
61         isPalindrom(word, 0, word.length() - 1);
62     } else {
63         cout << "Invalid option selected." << endl;
64     }
65
66     return 0;
67 }
68
```

Рисунок 3.11. Код до програми №5

```
Enter 'N' for a number or 'W' for a word to check if it's a palindrome:
N
So, Enter a number: 12344321
Your number is a palindrome!

Enter 'N' for a number or 'W' for a word to check if it's a palindrome:
W
So, Enter a word: LeVeL
Your word is a palindrome!
```

Рисунок 3.12. Приклади виконання програми №5

Створюємо дві перевантажені функції isPalindrom для перевірки числа і слова на паліндром. За допомогою циклу while будемо рухатися з права і зліва до середини й перевіряти чи рівні елементи між собою. У головній функції використовуємо перевірку на те, що користувач буде вводити чи слово, чи число і відповідно будемо викликати наші функції.

Фактично затрачений час 2.5 години.

**Посилання на файл у пулл реквесті**

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/219](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/219)



## Завдання №6

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main() {
7      int n, k;
8      cin >> n >> k;
9      vector<int> a(n);
10
11     int max_length = 0;
12     int current_length = 0;
13
14     for (int i = 0; i < n; i++) {
15         cin >> a[i];
16     }
17
18     for (int i = 0; i < n; i++) {
19         if (a[i] >= k) {
20             current_length++;
21         }
22         else {
23             if (max_length < current_length){
24                 max_length = current_length;
25             }
26             current_length = 0;
27         }
28     }
29     if (max_length < current_length){
30         max_length = current_length;
31     }
32
33     cout << max_length << endl;
34
35     return 0;
36 }
37
38
```

Рисунок 3.13. Код до програми №6

```
7 4
7 1 4 6 7 3 2
3
```

Рисунок 3.14. Приклад виконання програми №6

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.044	1.469	<a href="#">Перегляд</a>

Рисунок 3.15. Статус задачі на алготестері

За допомогою вектора будемо зберігати значення які введе користувач. У циклі for будемо проходитися по елементах нашого вектора і шукати чи рівні або більші вони за коефіцієнт нагостреності, якщо так то ми виводимо на екран групу елементів які стоять саме поряд, тому що така умова задачі.

Фактично затрачений час 2.5 години.

## Посилання на файл у пулл реквесті

### 4. Робота з командою:

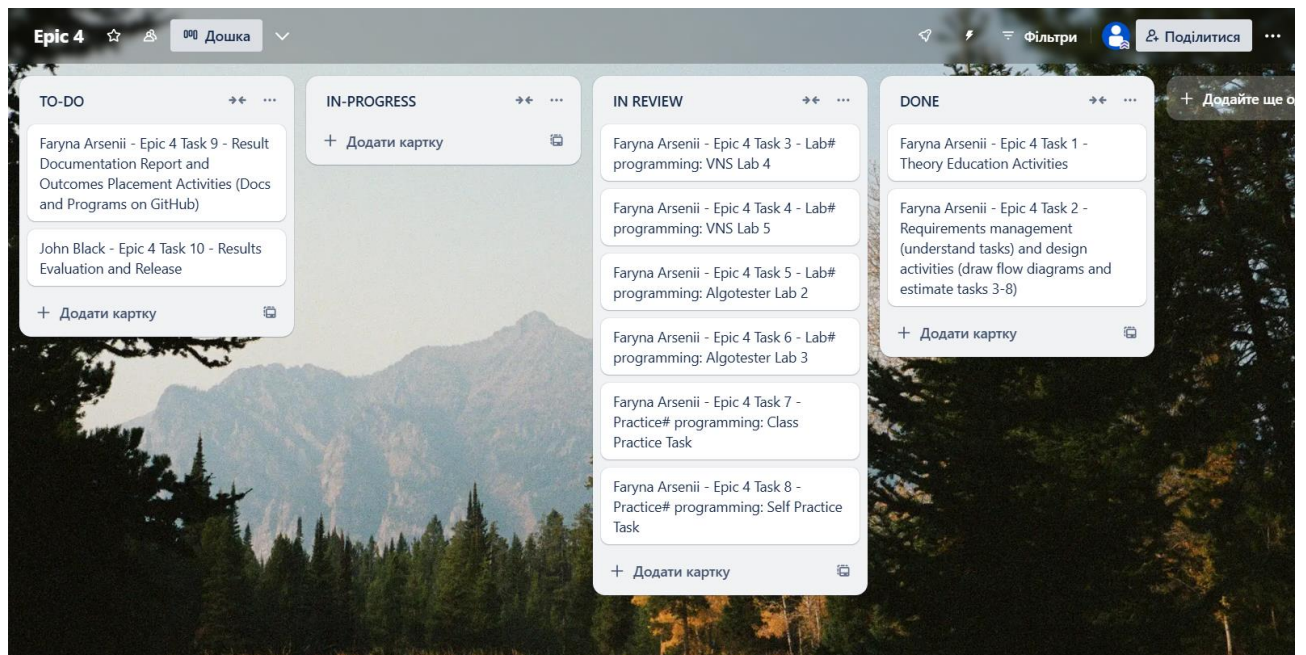


Рисунок 4.1. Командна дошка в Trello

**Висновок:** У межах практичних та лабораторних робіт блоку №4, я вивчив багато нового матеріалу, такого як: різні масиви одновимірні та двовимірні, вказівники та посилання, динамічні масиви та структури даних. Застосувавши вивчений матеріал на практиці краще зрозумів як це все працює і як це реалізовувати. Також створив блок-схему до найважчої задачі, і краще зрозумів як працює програма. Створив дошку в Trello для роботи з командою.