

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

Левченко Денис

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

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

[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=PLiPRE8VmJzOpn6](https://www.youtube.com/watch?v=D79J_q5SGzc&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=64)

[6PzYf0higmCEyGzo2A5g&index=64](https://www.youtube.com/watch?v=D79J_q5SGzc&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=64)

[https://www.youtube.com/watch?v=ifdk1fYSk1Y&list=PLiPRE8VmJzOpn6](https://www.youtube.com/watch?v=ifdk1fYSk1Y&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=65)

[PzYf0higmCEyGzo2A5g&index=65](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-](https://www.youtube.com/watch?v=999IE-6b7_s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=63)

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

[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=PLiPRE8VmJzOpn](https://www.youtube.com/watch?v=uQxG9gBROog&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=51)

[6PzYf0higmCEyGzo2A5g&index=51](https://www.youtube.com/watch?v=uQxG9gBROog&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=51)

[https://www.youtube.com/watch?v=maB87eyn7h8&list=PLiPRE8VmJzOpn](https://www.youtube.com/watch?v=maB87eyn7h8&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=52)

[6PzYf0higmCEyGzo2A5g&index=52](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 16)

Реалізувати з використанням масиву чергу (перший прийшов, перший пішов), для чого організувати додавання, знищення елементів у масиві і друк масиву після кожної операції.

Завдання №2

VNS LAB 5 – TASK 1 (VARIANT 16)

Задано двовимірний масив $N \times N$. Дозволяється довільно переставляти елементи усередині будь-якого стовпця. Перевірити, чи можна виконавши скінчену кількість перестановок у стовпцях, розташувати на бічній діагоналі елементи так, щоб вони зростали.

Завдання №3

ALGOTESTER LAB 2 (VARIANT 3)

Вам дано масив цілих чисел розміром NN , 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_{ii} та a_{i+1i+1} - виведіть **Stopped**

Врахуйте, що перевіряти треба також до взльоту.

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

Ви з'явилися у світі під назвою Атод посеред Пустелі Безправ'я. Так сталося, що Ви попали саме в той час і місце, де ведеться битва між чаклункою Ліною і темними силами, які хочуть знищити цей світ. На жаль, трапилась халепа, бо деякі слова із книги чар були пошкоджені під час битви. Одне таке слово можна відновити виконавши ритуал зцілення над пошкодженими буквами. Ритуал зцілення можна виконати на всіх **підряд** розташованих **пошкоджених** буквах. Вам не залишається нічого іншого як допомогти Ліні відновити ці слова і сказати скільки мінімально треба провести таких ритуалів, щоб прочитати одне з наймогутніших у цьому світі заклять - Поневолення Дракона!

Вхідні дані

У першому рядку N - кількість рядків у заклятті.

В наступних N рядках - набір слів w_1, \dots, w_M , розділених пробілами, де кожне слово може містити малі латинські літери та символ **##**, який позначає пошкоджену букву.

Вихідні дані

Єдине ціле число - мінімальна кількість ритуалів, які потрібно провести, щоб відновити закляття

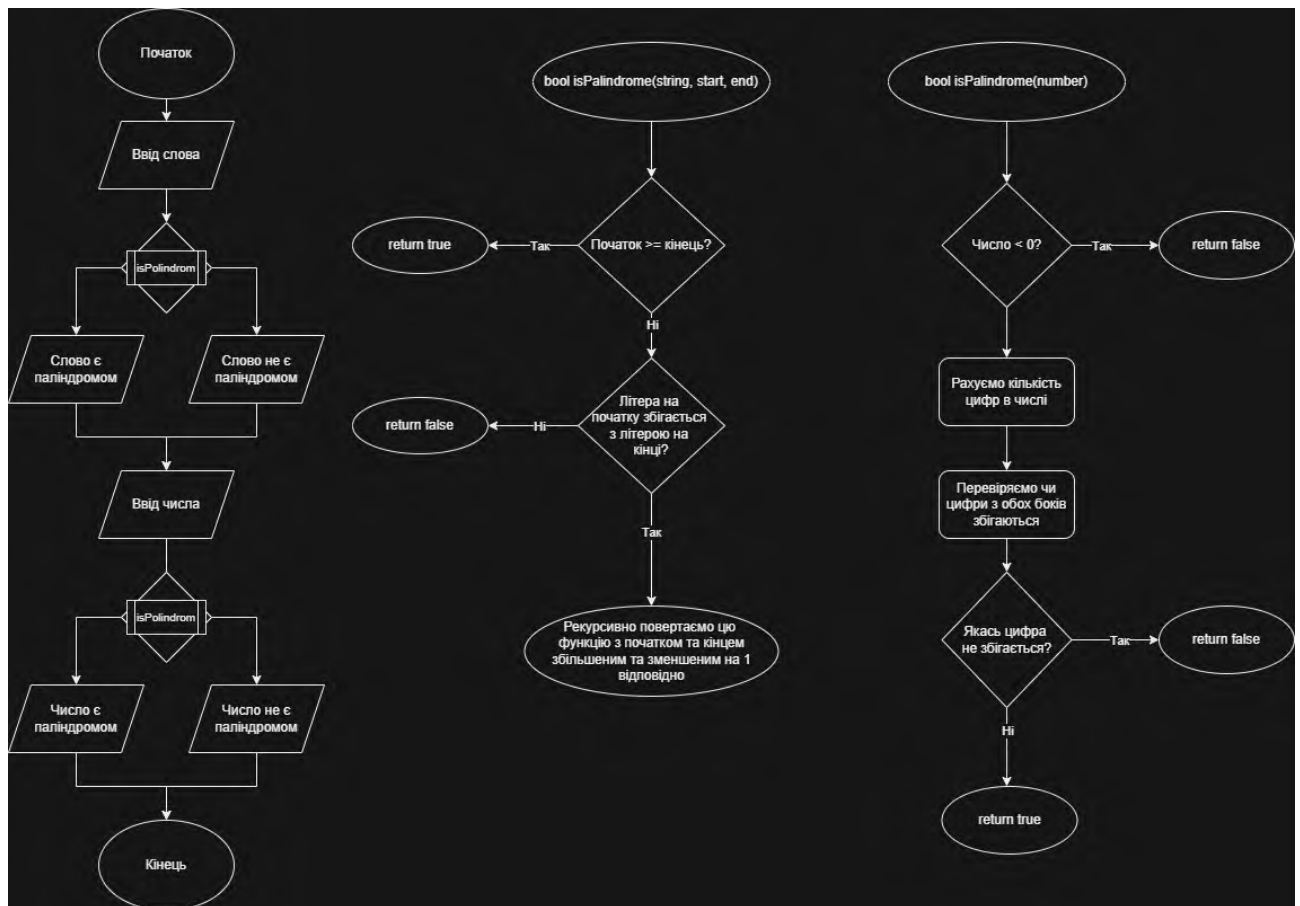


Рисунок 2.1. Блок-схема до CLASS PRACTICE WORK

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

Завдання №1

```
1 #include <iostream>
2 using namespace std;
3
4 const int N = 100;
5
6 class Queue {
7 private:
8     int a[N];
9     int front, rear, size;
10
11 public:
12     Queue() : front(0), rear(0), size(0) {}
13
14     void enqueue(int value) {
15         if (size == N) {
16             cout << "Черга переповнилась!" << endl;
17             return;
18         }
19         a[rear] = value;
20         rear = (rear + 1) % N;
21         size++;
22         printQueue();
23     }
24
25     void dequeue() {
26         if (size == 0) {
27             cout << "Черга порожня!" << endl;
28             return;
29         }
30         front = (front + 1) % N;
31         size--;
32         printQueue();
33     }
34
35     void printQueue() {
36         if (size == 0) {
37             cout << "Черга порожня!" << endl;
38             return;
39         }
40         cout << "Черга: ";
41         for (int i = 0; i < size; i++) {
42             cout << a[(front + i) % N] << " ";
43         }
44         cout << endl;
45     }
46 };
47
48 int main() {
49     Queue queue;
50     int n;
51     cout << "Введіть кількість елементів у черзі (не більше " << N << "): ";
52     cin >> n;
53
54     if (n > N || n < 0) {
55         cout << "Некоректна кількість елементів!" << endl;
56         return 1;
57     }
58
59     for (int i = 0; i < n; i++) {
60         int value;
61         cout << "Введіть елемент " << i + 1 << ": ";
62         cin >> value;
63         queue.enqueue(value);
64     }
65
66     int choice;
67     do {
68         cout << "\nОберіть операцію:\n";
69         cout << "1. Додати елемент\n";
70         cout << "2. Вивести елемент\n";
71         cout << "3. Вихід\n";
72         cout << "Введіть номер операції: ";
73         cin >> choice;
74
75         switch (choice) {
76             case 1:
77                 int value;
78                 cout << "Введіть значення для додавання: ";
79                 cin >> value;
80                 queue.enqueue(value);
81                 break;
82             case 2:
83                 queue.dequeue();
84                 break;
85             case 3:
86                 cout << "Вихід з програми." << endl;
87                 break;
88             default:
89                 cout << "Неправильний вибір!" << endl;
90         }
91     } while (choice != 3);
92
93     return 0;
94 }
```

Рисунок 3.1. Код до програми № 1


```
Введіть кількість елементів у черзі (не більше 100): 4
Введіть елемент 1: 10
Черга: 10
Введіть елемент 2: 21
Черга: 10 21
Введіть елемент 3: 5
Черга: 10 21 5
Введіть елемент 4: 12
Черга: 10 21 5 12

Оберіть операцію:
1. Додати елемент
2. Видалити елемент
3. Вихід
Введіть номер операції: 1
Введіть значення для додавання: 5
Черга: 10 21 5 12 5

Оберіть операцію:
1. Додати елемент
2. Видалити елемент
3. Вихід
Введіть номер операції: 2
Черга: 21 5 12 5

Оберіть операцію:
1. Додати елемент
2. Видалити елемент
3. Вихід
Введіть номер операції: 3
Вихід з програми.
PS C:\Users\razzie>
```

Рисунок 3.2. Приклад виконання програми № 1

Фактично затрачений час 2 години.

Завдання №2

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <limits>
5
6  using namespace std;
7
8  bool canSortAntiDiagonal(const vector<vector<int>>& matrix) {
9      int n = matrix.size();
10     vector<vector<int>> column_elements(n);
11
12
13     for (int i = 0; i < n; ++i) {
14         for (int row = 0; row < n; ++row) {
15             column_elements[i].push_back(matrix[row][n - 1 - i]);
16         }
17         sort(column_elements[i].begin(), column_elements[i].end());
18     }
19
20     int prev_value = numeric_limits<int>::min();
21
22     for (int i = 0; i < n; ++i) {
23         auto it = upper_bound(column_elements[i].begin(), column_elements[i].end(), prev_value);
24         if (it == column_elements[i].end()) {
25             return false;
26         }
27         prev_value = *it;
28     }
29
30     return true;
31 }
32
33 int main() {
34     int n;
35     cout << "Введіть розмір матриці N: ";
36     cin >> n;
37
38     vector<vector<int>> matrix(n, vector<int>(n));
39
40     for (int i = 0; i < n; ++i) {
41         for (int j = 0; j < n; ++j) {
42             cout << "Введіть елемент [" << i + 1 << "][" << j + 1 << "]: ";
43             cin >> matrix[i][j];
44         }
45     }
46
47     if (canSortAntiDiagonal(matrix)) {
48         cout << "Можливо розташувати елементи на бічній діагоналі у порядку зростання." << endl;
49     } else {
50         cout << "Неможливо розташувати елементи на бічній діагоналі у порядку зростання." << endl;
51     }
52
53     return 0;
54 }
```

Рисунок 3.3. Код до програми № 2

```

Введіть розмір матриці N: 3
Введіть елемент [1][1]: 1
Введіть елемент [1][2]: 2
Введіть елемент [1][3]: 3
Введіть елемент [2][1]: 5
Введіть елемент [2][2]: 6
Введіть елемент [2][3]: 7
Введіть елемент [3][1]: 2
Введіть елемент [3][2]: 1
Введіть елемент [3][3]: 1
Можливо розташувати елементи на бічній діагоналі у порядку зростання.
PS C:\Users\razzie>

```

Рисунок 3.4. Приклад виконання програми № 2

Фактично затрачений час 2 години.

Завдання №3

```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6      int N;
7      cin >> N;
8      vector<int> a(N);
9
10     for (int i = 0; i < N; ++i) {
11         cin >> a[i];
12     }
13
14     int left = 0;
15     int right = N - 1;
16
17     while (true) {
18         if (left == right) {
19             cout << left + 1 << " " << right + 1 << endl;
20             cout << "Collision" << endl;
21             break;
22         }
23         else if (left + 1 == right) {
24             cout << left + 1 << " " << right + 1 << endl;
25             cout << "Stopped" << endl;
26             break;
27         }
28         else if (left > right) {
29             cout << left + 1 << " " << right + 1 << endl;
30             cout << "Miss" << endl;
31             break;
32         }
33
34         left += a[left];
35         right -= a[right];
36     }
37
38     return 0;
39 }

```

Рисунок 3.5. Код до програми №3

```

10
1 3 1 1 5 1 1 2 1 2
5 6
Stopped
PS C:\Users\razzie>

```

Рисунок 3.6. Приклад виконання програми №3

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
2 дні тому	C++ 23	Зараховано	0.003	1.273	Перегляд

Рисунок 3.7. Статус задачі на алготестері

Фактично затрачений час 3.5 години.

Завдання №4

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  string compressString(const string& s) {
6      string compressed;
7      int n = s.length();
8
9      for (int i = 0; i < n; ) {
10         char currentChar = s[i];
11         int count = 0;
12
13         while (i < n && s[i] == currentChar) {
14             count++;
15             i++;
16         }
17
18         compressed += currentChar;
19         if (count > 1) {
20             compressed += to_string(count);
21         }
22     }
23
24     return compressed;
25 }
26
27 int main() {
28     string s;
29     cin >> s;
30
31     string compressed = compressString(s);
32     cout << compressed << endl;
33
34     return 0;
35 }

```

Рисунок 3.8. Код до програми №4

```

AAAAABBBCCQQQ
A4B3CQ4
PS C:\Users\razzie>

```

Рисунок 3.9. Приклад виконання програми №4

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
2 дні тому	C++ 23	Зараховано	0.003	1.309	Перегляд

Рисунок 3.10. Статус задачі на алготестері

Фактично затрачений час 2.5 години.

Завдання №5

```

1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  bool isPalindrome(const string& str, int start, int end) {
7      if (start >= end) return true;
8      if (str[start] != str[end]) return false;
9      return isPalindrome(str, start + 1, end - 1);
10 }
11
12 bool isPalindrome(int number) {
13     if (number < 0) return false;
14     string str = to_string(number);
15     return isPalindrome(str, 0, str.size() - 1);
16 }
17
18 int main() {
19     string str;
20     cout << "Введіть слово для перевірки: ";
21     cin >> str;
22     if (isPalindrome(str, 0, str.size() - 1)) {
23         cout << str << " є паліндромом." << endl;
24     } else {
25         cout << str << " не є паліндромом." << endl;
26     }
27
28     int number;
29     cout << "Введіть число для перевірки: ";
30     cin >> number;
31     if (isPalindrome(number)) {
32         cout << number << " є паліндромом." << endl;
33     } else {
34         cout << number << " не є паліндромом." << endl;
35     }
36
37     return 0;
38 }

```

Рисунок 3.11. Код до програми №5

```

Введіть слово для перевірки: level
level є паліндромом.
Введіть число для перевірки: 54345
54345 є паліндромом.
PS C:\Users\razzie>

```

Рисунок 3.12. Приклади виконання програми №5

Фактично затрачений час 2.5 години.

Завдання №6

```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main()
6  {
7      short n;
8      cin >> n;
9      cin.ignore();
10     vector<string> arr(n);
11     for (auto &el : arr)
12     {
13         getline(cin, el);
14     }
15
16     long long count = 0;
17     for (int j = 0; j < n; j++)
18     {
19         for (size_t i = 0; i < arr[j].length(); i++)
20             [1]
21             if (i == arr[j].length() - 1 && arr[j][i] == '#')
22             {
23                 count++;
24             }
25             else if (arr[j][i - 1] == '#' && arr[j][i] != '#')
26             {
27                 count++;
28             }
29             [2]
30     }
31
32
33     cout << count;
34
35     return 0;
36 }

```

Рисунок 3.13. Код до програми №6

```

8
d#rkness b#yond twilight
crimson beyond blood that flows
buried in the stream of time is where your power grow
i pledge myself to conquer all the foes who stand
before the mighty gift bestowed in my unworthy hand
let the fools who stand before me be destroyed
by the power you and i possess
dr#gon sl#ve
4
PS C:\Users\razzie> █

```

Рисунок 3.14. Приклад виконання програми №6

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
3 години тому	C++ 23	Зараховано	0.018	1.125	Перегляд

Рисунок 3.15. Статус задачі на алготестері

Фактично затрачений час 2.5 години.

Посилання на файл у пулл реквесті

4. Робота з командою:

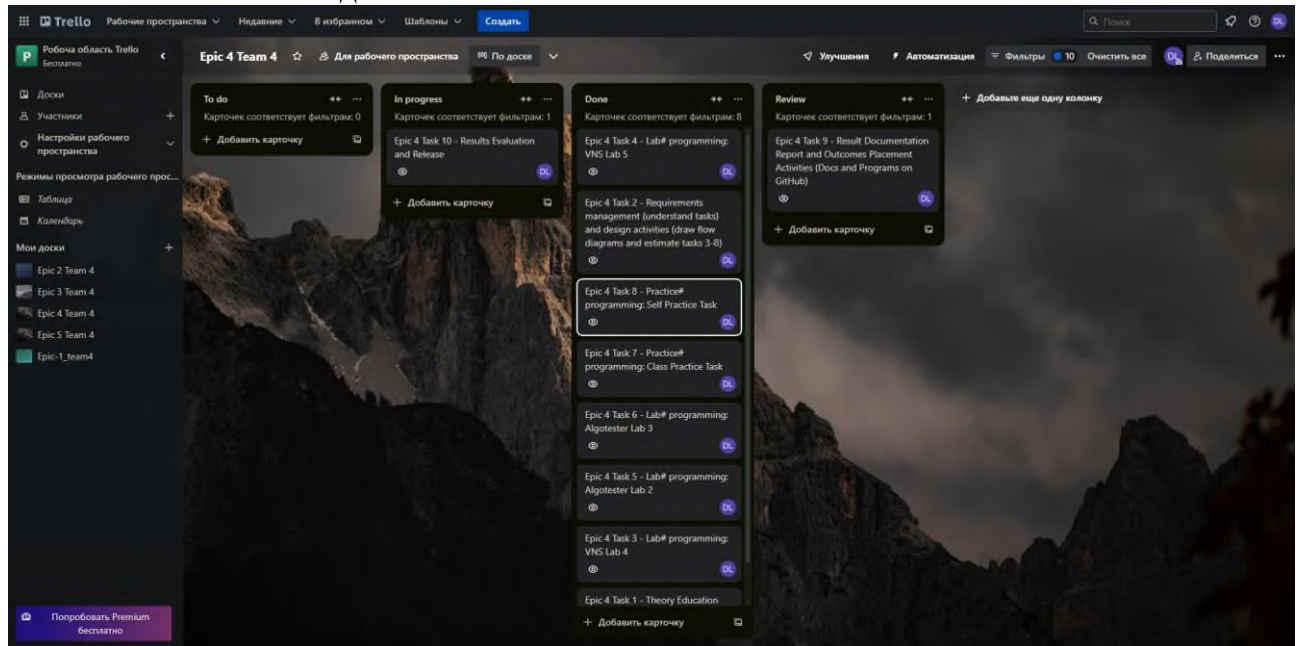


Рисунок 4.1. Командная доска в Trello

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.