

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



про виконання лабораторних та практичних робіт блоку № 4

Звіт

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

Виконала:

Студентка групи ШІ-11

Андрусишин Соломія Володимирівна

Тема роботи: Ознайомлення з одновимірними та двовимірними масивами, їх оголошенням, ініціалізацією та використанням у програмах C++. Вивчення взаємодії між масивами та вказівниками, особливостей їх використання в контексті функцій, а також принципів роботи з посиланнями. Розгляд різниці між статичними та динамічними масивами, а також алгоритмів роботи з динамічною пам'яттю, використання операторів new та delete. Вивчення структур даних, включаючи вкладені структури та об'єднання (union), їх оголошення та практичне застосування. Аналіз алгоритмів обробки масивів і структур, їх інтеграції в більш складні алгоритми та застосування у вирішенні різних завдань.

Мета роботи: : Опанувати методи створення та ініціалізації одновимірних і двовимірних масивів, виконання основних операцій, таких як індексація, обхід і маніпуляції з даними за допомогою циклів та функцій. Навчитися використовувати вказівники для доступу до елементів масивів, розуміти їхню арифметику та зв'язок з посиланнями. Дослідити різницю між вказівниками та посиланнями, особливо в контексті передачі параметрів у функції. Засвоїти динамічне виділення пам'яті та управління нею за допомогою операторів new і delete, створювати та використовувати динамічні масиви. Зрозуміти основи роботи зі структурами даних, навчитися оголошувати та використовувати вкладені структури та об'єднання, моделювати складні об'єкти. Ознайомитися з практичними прикладами алгоритмів пошуку та сортування в масивах, навчитися їх застосовувати для обробки даних у різних структурах. Розвинути вміння створювати, аналізувати та оптимізувати алгоритми для роботи з масивами та структурами даних у контексті вирішення практичних задач

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.

Індивідуальний план опрацювання теорії:

- ***Класи пам'яті у C++***

Джерела:

<https://youtu.be/NyOjKd5Qruk?si=Y0e0cRchCiaSzWfI>

- Висновок: Отже, з цих теоретичних відомостей я дізналась, які є класи пам'яті у C++. Зрозуміла яка різниця між різними типами пам'яті:
 - *Stack*- це область пам'яті, де зберігаються локальні змінні, виклики функцій та їх параметри. Вона працює за принципом (останній прийшов — перший пішов).
 - *Static memory*- це область пам'яті, виділена на весь час роботи програми. Здебільшого використовується для глобальних змінних.

- *Dynamic memory* - це область пам'яті, яка виділяється під час виконання програми, а не заздалегідь. Програміст сам вирішує, коли виділяти та звільняти цю пам'ять.

2. Вступ до Масивів і Вказівників:

- Джерела:

<https://youtu.be/ULdbOaMBPYc?si=dV57GhauXPedfki1>
https://youtu.be/zopWRIYOWw?si=RL3PZttWeA_gbM6l
<https://studfile.net/preview/5319824/page:10/>

- Висновок:

- *Що таке масиви?* - це набір однотипних елементів. Кожен елемент має свій номер (індекс). Всі елементи масиву упорядковані за своїм індексом.

Основними операціями при роботі з масивами є:

- -виділення пам'яті для збереження елементів масиву;
 - -вивільнення пам'яті;
 - -сортування елементів масиву;
 - -пошук елемента масиву, що відповідає заданому критерію пошуку.
-
- *Статичні/Динамічними масиви*
 - **статичні**, коли кількість елементів масиву відома на етапі розробки програми(зберігаються у **стеку**).
 - **динамічні**, коли кількість елементів масиву стає відомою під час виконання програми. Розташування в пам'яті: виділяються в кучі (**heap**).
 - *Що таке вказівники?* - це спеціальні змінні, які зберігають **адресу іншої змінної** в пам'яті комп'ютера. Замість збереження звичайного значення (наприклад, числа чи символу), вони "вказують", де саме знаходиться це значення.

3. Одновимірні Масиви/Двовимірні масиви:

- Джерела:
https://youtu.be/ULdbOaMBPYc?si=43ehUg7K6_fuDE02
- Висновок:
 - *Одновимірний масив* — це структурований тип даних, який складається з фіксованого числа однотипних елементів, що мають спільне ім'я.
 - *Двовимірний масив* — це масив, що містить елементи, в яких номер складається з двох індексів.

4 Вказівники та Посилання:

- Джерела:
<https://acode.com.ua/urok-84-vkazivnyky/>
<https://youtu.be/btNOGePn778?si=GBWbnS55om5w9dG1>
- Висновок: Різниця між вказівниками та посиланнями в контексті функцій.
 - Вказівник — це "адреса змінної", яку можна змінити.
 - Посилання — це "інше ім'я змінної", яке змінити не можна.
 - Оператор адреси & дозволяє дізнатися, яку адресу в пам'яті присвоєно певній змінній.

5. Структури Даних:

- Джерела:
<https://www.youtube.com/watch?v=sZiA1fFVb4g&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=62>
- Висновок :
 - Структура дозволяє згрупувати змінні різних типів даних в єдине ціле.
 - Масиви можна використовувати для зберігання колекцій даних одного типу в структурі.
 - Вказівники у структурах використовуються для:
 - Посилання на інші структури (зв'язні списки, дерева).
 - Динамічного виділення пам'яті.

6. Вкладені Структури Даних:

- Джерела:
https://youtu.be/999IE-6b7_s?si=LtpQ6sehVb4hTkFu

Висновок :

- Звичайна структура зберігає прості дані (наприклад, числа чи рядки).
- Вкладена структура зберігає одну структуру всередині іншої. Це дає змогу згрупувати пов'язані дані. Вкладені структури корисні, коли потрібно описати складні об'єкти, які складаються з дрібніших частин.

Виконання роботи:

Завдання №1

(VNS Lab4 Task1) -

6.

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

Завдання №2

(VNS Lab5 Task1)

Елемент матриці є сідловою точкою, якщо він є найменшим у своєму рядку й найбільшим у своєму стовпці (або навпаки: найбільшим у своєму рядку й найменшим у своєму стовпці). Для заданої матриці визначити всі сідлові точки.

Завдання №3

(Algotester Lab2 variant 2) –

У вас є масив `rr` розміром `N`. Також вам дано 3 цілих числа.

Спочатку ви маєте видалити з масиву ці 3 числа, які вам дані.
Після цього перетворити цей масив у масив сум,
розміром $N_{\text{new}} - 1$ (розмір нового масиву після видалення елементів), який буде відображати суми сусідніх елементів нового масиву.

Далі необхідно вивести масив сум на екран.

Input

У першому рядку ціле число NN - кількість чисел

У другому рядку масив tt , який складається з NN цілих чисел

У третьому рядку 3 цілих числа, a, b, c , які треба видалити з масиву

Output

У першому рядку ціле число MM - кількість чисел у масиві, який буде виведено

У наступному рядку MM чисел - новий масив

Завдання №4

(Algotester Lab3 variant 2) –

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.

Input

У першому рядку ціле число NN

у другому рядку NN цілих чисел $a_1..a_n$

У третьому рядку ціле число MM

у четвертому рядку MM цілих чисел $b_1..b_n$

Output

У першому рядку одне ціле число - кількість елементів, які наявні в обох масивах одночасно.

У другому рядку кількість унікальних елементів в обох масивах (тобто кількість унікальних елементів у масиві, який буде об'єднанням двох даних).

Завдання №5

(Class Practice Work)

Перевірка чи слово або число є паліндромом 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»).

Мета Задачі Навчитися користуватися механізмами перевантаження функції та використовувати рекурсію для вирішення задач обчислення.

Вимоги:

1. Визначення функції: а. Реалізуйте рекурсивну функцію `isPalindrome`, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції: а. `bool isPalindrome(const string& str, int start, int end);`
3. Перевантаження функцій: а. Перевантажте функцію `isPalindrome` для роботи з цілими значеннями. б. `bool isPalindrome(ціле число);`
4. Рекурсія: а. Рекурсивна функція для рядків перевірить символи в поточній початковій і кінцевій позиціях. Якщо вони збігаються, він буде рекурсивно перевіряти наступні позиції, поки початок не перевищить кінець, після чого рядок буде визначено як паліндром.
Кроки реалізації • Визначте та реалізуйте рекурсивну функцію `isPalindrome` для рядків. • Визначте та реалізуйте перевантажену функцію `isPalindrome` для цілих чисел. Використати математичний підхід щоб перевірити чи число є паліндромом.

Завдання №6

(Self Practice Work)

Оголосити три масиви. Перші два заповнити випадковими значеннями від 10 до 30. До елементів третього масиву записати суму відповідних елементів перших двох масивів. (У нульовий осередок третього - суму нульових осередків першого і другого масивів і так далі). Потім знайти середнє арифметичне елементів третього масиву, максимальне значення та мінімальне значення, які він зберігає

Завдання №1 (VNS Lab4 Task1)

```
1  #include <iostream>
2  #include <ctime>
3  using namespace std;
4
5  void Initialize(int arr[], int size);
6  void Show(int arr[], int size);
7  void DeleteElement(int*& arr, int& size, int index);
8  bool AddAfterFirstEven(int*& arr, int& size);
9  int main() {
10     srand(static_cast<unsigned>(time(nullptr)));
11
12     int size;
13     cout << "Введіть кількість елементів масиву: ";
14     cin >> size;
15
16     if (size <= 0) {
17         cout << "Неправильно введено. Введіть число більше за 0." << endl;
18         return 1;
19     }
20
21     int* arr = new int[size];
22
23     Initialize(arr, size);
24     cout << "Початковий масив: ";
25     Show(arr, size);
26
27     int indexToDelete;
28     cout << "Введіть індекс елемента для видалення (від 0 до " << size - 1 << "): ";
29     cin >> indexToDelete;
30
31     if (indexToDelete >= 0 && indexToDelete < size) {
32         DeleteElement(arr, size, indexToDelete);
33         cout << "Масив після видалення елемента: ";
34         Show(arr, size);
35     } else {
36         cout << "Неправильно введено. Видалення не виконане." << endl;
37     }
38
39     if (AddAfterFirstEven(arr, size)) {
40         cout << "Масив після додавання нового елемента: ";
41         Show(arr, size);
42     } else {
43         cout << "Парного елемента не знайдено. Додавання не виконане." << endl;
44     }
45 }
```

```

49
50 void Initialize(int arr[], int size) {
51     for (int i = 0; i < size; i++) {
52         arr[i] = rand() % 100;
53     }
54 }
55
56 void Show(int arr[], int size) {
57     for (int i = 0; i < size; i++) {
58         cout << arr[i] << " ";
59     }
60     cout << endl;
61 }
62
63 void DeleteElement(int*& arr, int& size, int index) {
64     for (int i = index; i < size - 1; i++) {
65         arr[i] = arr[i + 1];
66     }
67     size--;
68 }
69
70 bool AddAfterFirstEven(int*& arr, int& size) {
71     for (int i = 0; i < size; i++) {
72         if (arr[i] % 2 == 0) {
73             int newSize = size + 1;
74             int* newArr = new int[newSize];
75
76             for (int j = 0; j <= i; j++) {
77                 newArr[j] = arr[j];
78             }
79
80             newArr[i + 1] = arr[i] + 2;
81
82             for (int j = i + 1; j < size; j++) {
83                 newArr[j + 1] = arr[j];
84             }
85
86             delete[] arr;
87             arr = newArr;
88             size = newSize;
89             return true;
90         }
91     }
92     return false;
93 }
94

```

```

Введіть кількість елементів у масиві: 5
Початковий масив: 84 32 54 65 60
Введіть індекс елемента для видалення (від 0 до 4): 0
Масив після видалення елемента: 32 54 65 60
Масив після додавання нового елемента: 32 34 54 65 60
PS C:\Users\Solomia\Desktop\epic_4>

```

Завдання №2 (VNS Lab5 Task1)

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int rows, cols;
6      cout << "Введіть к-сть рядків: ";
7      cin >> rows;
8      cout << "Введіть к-сть стовпців: ";
9      cin >> cols;
10
11     int matrix[100][100];
12
13     cout << "Введіть всі елементи матриці:" << endl;
14     for (int i = 0; i < rows; i++) {
15         for (int j = 0; j < cols; j++) {
16             cout << "Елемент [" << i << "][" << j << "]: ";
17             cin >> matrix[i][j];
18         }
19     }
20
21     cout << "Загальний вигляд матриці:" << endl;
22     for (int i = 0; i < rows; i++) {
23         for (int j = 0; j < cols; j++) {
24             cout << matrix[i][j] << " ";
25         }
26         cout << endl;
27     }
28
29
30     cout << "Сідлові точки:" << endl;
31     bool foundSaddlePoint = false;
32
33     for (int i = 0; i < rows; i++) {
34         for (int j = 0; j < cols; j++) {
35             int current = matrix[i][j];
36
37
38             bool isMinInRow = true;
39             for (int k = 0; k < cols; k++) {
40                 if (matrix[i][k] < current) {
41                     isMinInRow = false;
42                     break;
43                 }
44             }
45         }
46     }
```

```

45
46         bool isMaxInCol = true;
47         for (int k = 0; k < rows; k++) {
48             if (matrix[k][j] > current) {
49                 isMaxInCol = false;
50                 break;
51             }
52         }
53
54         if (isMinInRow && isMaxInCol) {
55             cout << "Елемент [" << i << "][" << j << "] = " << current << endl;
56             foundSaddlePoint = true;
57         }
58     }
59 }
60
61 if (!foundSaddlePoint) {
62     cout << "Сідлових точок не знайдено." << endl;
63 }
64
65 return 0;
66 }
67

```

Введіть к-сть рядків: 3
Введіть к-сть стовпців: 3
Введіть всі елементи матриці:

Елемент [0][0]: 4

Елемент [0][1]: 6

Елемент [0][2]: 7

Елемент [1][0]: 1

Елемент [1][1]: 3

Елемент [1][2]: 4

Елемент [2][0]: 5

Елемент [2][1]: 7

Елемент [2][2]: 8

Загальний вигляд матриці:

4 6 7

1 3 4

5 7 8

Сідлові точки:

Елемент [2][0] = 5

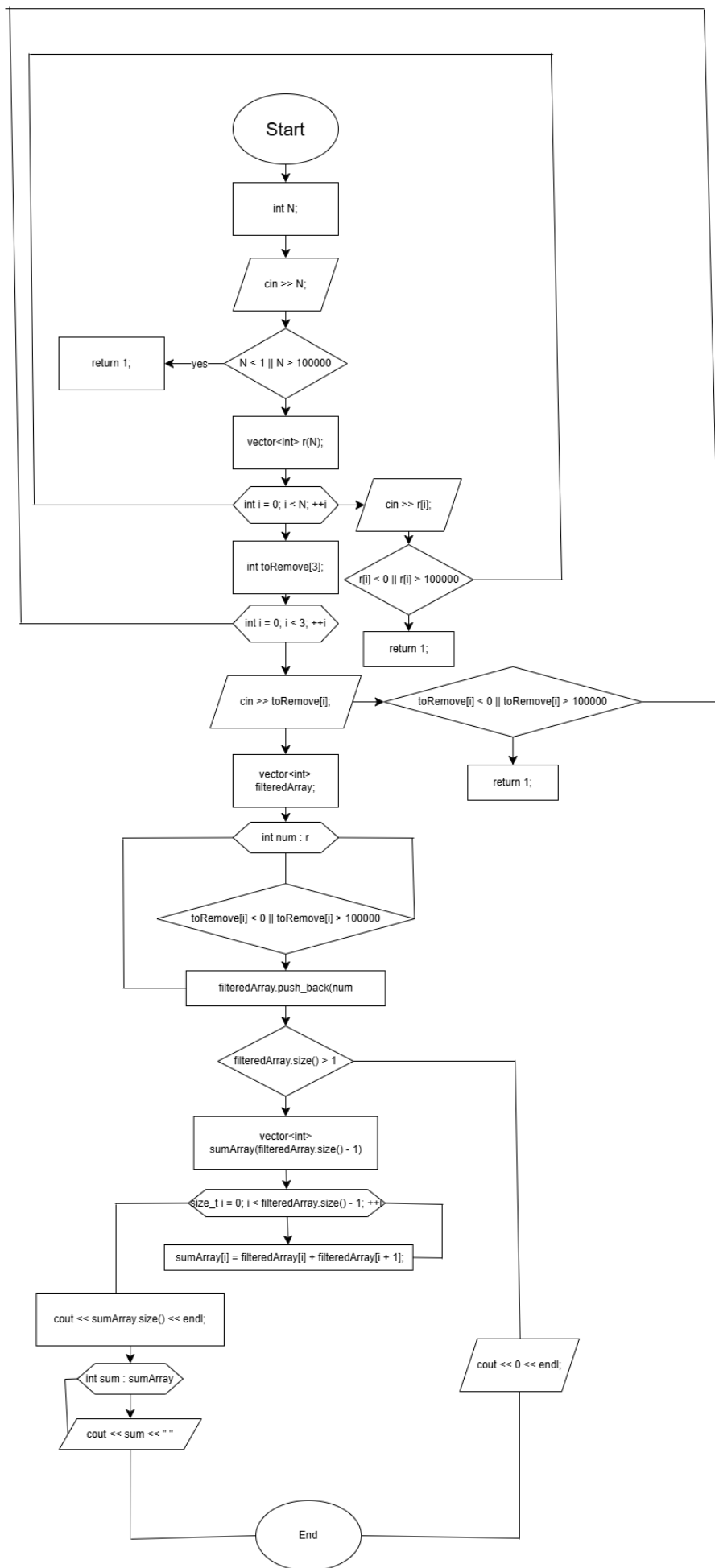
PS C:\Users\Solomia\Desktop\epic_4> █

Завдання №3

(Algotester Lab 2 variant 2)

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6      int N;
7      cin >> N;
8
9      if (N < 1 || N > 100000) {
10         return 1;
11     }
12
13     vector<int> r(N);
14     for (int i = 0; i < N; ++i) {
15         cin >> r[i];
16         if (r[i] < 0 || r[i] > 100000) {
17             return 1;
18         }
19     }
20
21     int toRemove[3];
22     for (int i = 0; i < 3; ++i) {
23         cin >> toRemove[i];
24         if (toRemove[i] < 0 || toRemove[i] > 100000) {
25             return 1;
26         }
27     }
28
29     vector<int> filteredArray;
30     for (int num : r) {
31         if (num != toRemove[0] && num != toRemove[1] && num != toRemove[2]) {
32             filteredArray.push_back(num);
33         }
34     }
35
36     if (filteredArray.size() > 1) {
37         vector<int> sumArray(filteredArray.size() - 1);
38         for (size_t i = 0; i < filteredArray.size() - 1; ++i) {
39             sumArray[i] = filteredArray[i] + filteredArray[i + 1];
40         }
41
42         cout << sumArray.size() << endl;
43         for (int sum : sumArray) {
44             cout << sum << " ";
45         }
46     } else {
47         cout << 0 << endl;
48     }
49
50     return 0;
51 }
```

```
6
1 2 3 4 5 7
4 5 6
3
3 5 10
```



Завдання №4

(Algotester Lab 3 variant 2)

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6      int n, m;
7
8      cin >> n;
9      if (n <= 0 || n > 100) {
10         return 1;
11     }
12
13     vector<int> firstArray(n);
14     for (int i = 0; i < n; ++i) {
15         cin >> firstArray[i];
16         if (firstArray[i] < 0 || firstArray[i] > 100) {
17             return 1;
18         }
19     }
20
21     cin >> m;
22     if (m <= 0 || m > 100) {
23         return 1;
24     }
25
26     vector<int> secondArray(m);
27     for (int i = 0; i < m; ++i) {
28         cin >> secondArray[i];
29         if (secondArray[i] < 0 || secondArray[i] > 100) {
30             return 1;
31         }
32     }
33
34     int commonCount = 0;
35     int totalUnique = 0;
36
37     for (int x : secondArray) {
38         for (int y : firstArray) {
39             if (x == y) {
40                 ++commonCount;
41                 break;
42             }
43         }
44     }
45
46     totalUnique = n + m - commonCount;
47
48     cout << commonCount << endl;
49     cout << totalUnique << endl;
50
51     return 0;
52 }
53
```



```
5
1 2 3 4 5
5
4 5 6 7 8
2
8
```

Завдання №5 (Class Practice Work)

```
1  #include <iostream>
2  using namespace std;
3
4  bool checkPalindrome(const char* word, int length) {
5      int left = 0, right = length - 1;
6
7      while (left < right) {
8          if (word[left] != word[right]) {
9              return false;
10         }
11         left++;
12         right--;
13     }
14
15     return true;
16 }
17
18 bool checkPalindromeNumber(int num) {
19     int reversed = 0, temp = num;
20
21     while (temp > 0) {
22         reversed = reversed * 10 + temp % 10;
23         temp /= 10;
24     }
25
26     return num == reversed;
27 }
28
29 int main() {
30     char type;
31     cout << "Choose type to check for palindrome (W for word, N for number): ";
32     cin >> type;
33
34     if (type == 'W' || type == 'w') {
35         char word[100];
36         cout << "Enter a word: ";
37         cin >> word;
38
39         int length = 0;
40         while (word[length] != '\0') {
41             length++;
42         }
43
44         if (checkPalindrome(word, length)) {
45             cout << "The word is a palindrome!" << endl;
46         } else {
47             cout << "The word is not a palindrome." << endl;
48         }
49     } else if (type == 'N' || type == 'n') {
50         int number;
51         cout << "Enter a number: ";
52         cin >> number;
53
54         if (checkPalindromeNumber(number)) {
55             cout << "The number is a palindrome!" << endl;
56         } else {
57             cout << "The number is not a palindrome." << endl;
58         }
59     } else {
60         cout << "Invalid input. Please enter W for word or N for number." << endl;
61     }
62
63     return 0;
64 }
```

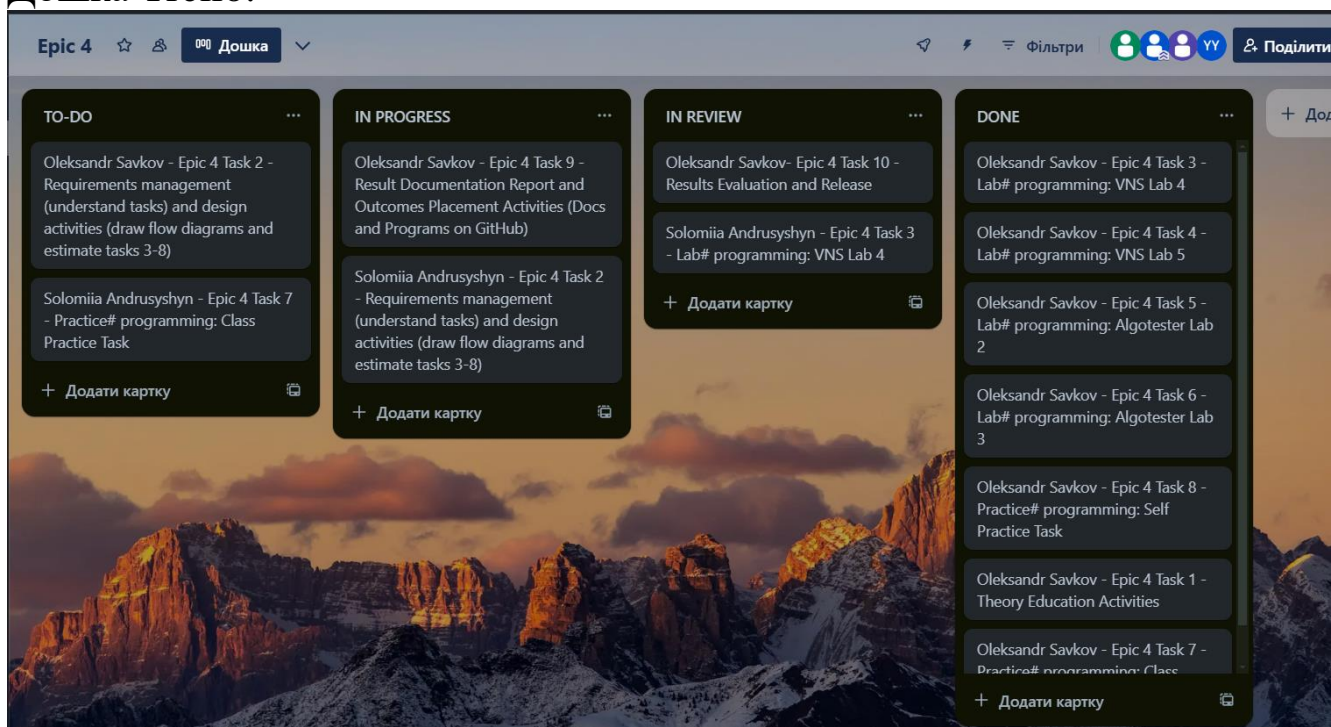
```
Choose type to check for palindrome (W for word, N for number): w
Enter a word: lol
The word is a palindrome!
```

Завдання №6 (Self Practice Work)

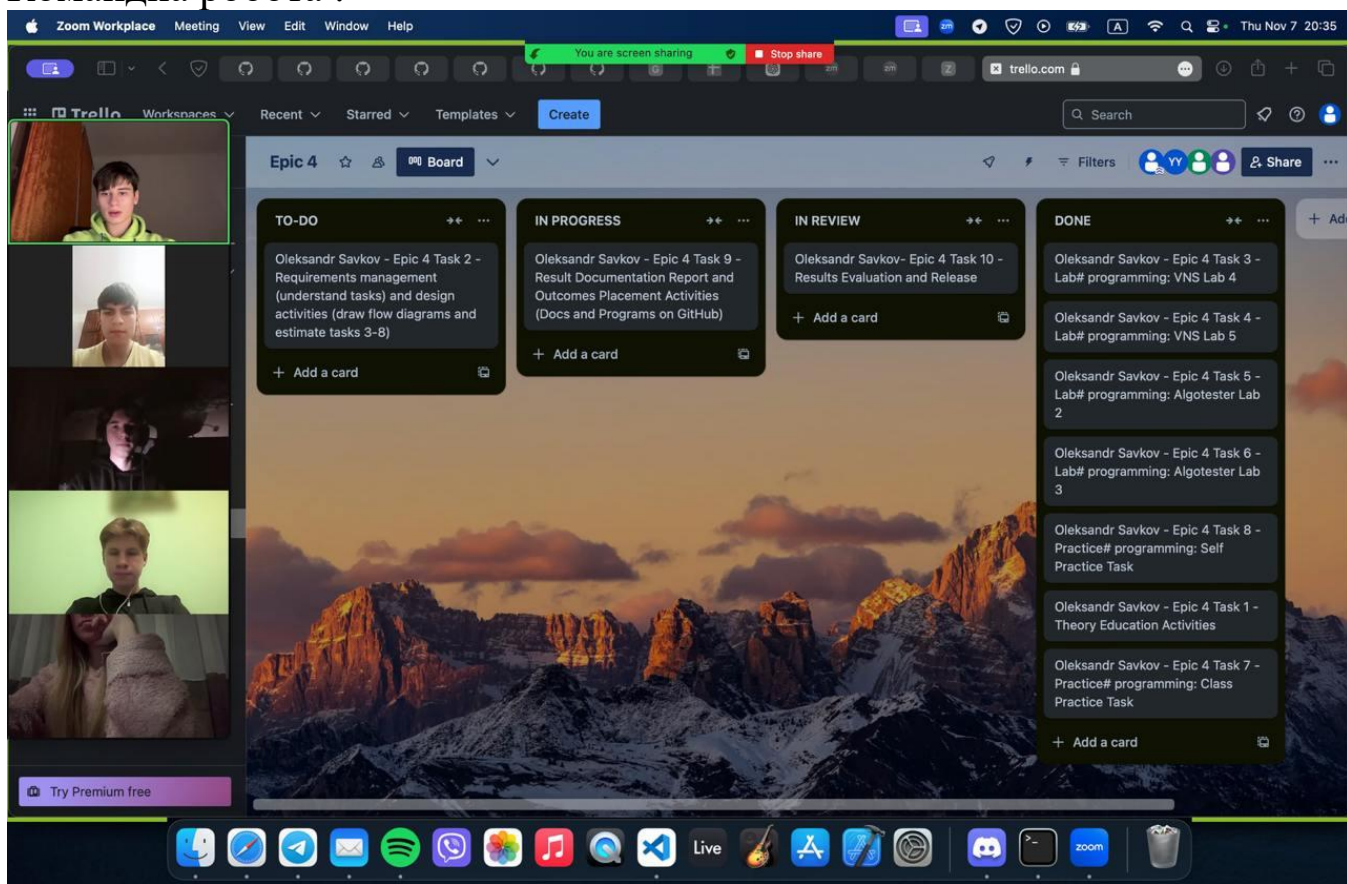
```
1  #include <iostream>
2  #include <ctime>
3  #include <cstdlib>
4  using namespace std;
5
6  int main() {
7      const int SIZE = 10;
8
9      int масив1[SIZE] = {};
10     int масив2[SIZE] = {};
11     int масив3[SIZE] = {};
12
13     srand(time(NULL));
14
15     cout << "Масив 1: ";
16     for (int i = 0; i < SIZE; i++) {
17         масив1[i] = 10 + rand() % 21;
18         cout << масив1[i] << " | ";
19     }
20     cout << endl << endl;
21
22     cout << "Масив 2: ";
23     for (int i = 0; i < SIZE; i++) {
24         масив2[i] = 10 + rand() % 21;
25         cout << масив2[i] << " | ";
26     }
27     cout << endl << endl;
28
29     cout << "Масив 3 (сума масивів 1 та 2): ";
30     for (int i = 0; i < SIZE; i++) {
31         масив3[i] = масив1[i] + масив2[i];
32         cout << масив3[i] << " | ";
33     }
34     cout << endl << endl;
35
36     int середнєЗначення = 0;
37     int сума = 0;
38     int мініЗначення = масив3[0];
39     int максЗначення = масив3[0];
40
41     for (int i = 0; i < SIZE; i++) {
42         сума += масив3[i];
43
44         if (масив3[i] < мініЗначення) {
45             мініЗначення = масив3[i];
46         }
47
48         if (масив3[i] > максЗначення) {
49             максЗначення = масив3[i];
50         }
51     }
52
53     середнєЗначення = сума / SIZE;
54
55     cout << "Середнє арифметичне = " << середнєЗначення << endl;
56     cout << "Мінімальне значення = " << мініЗначення << endl;
57     cout << "Максимальне значення = " << максЗначення << endl;
58     cout << endl << endl;
59
60     return 0;
61 }
62
```

```
Масив 1: 12 | 12 | 27 | 19 | 14 | 24 | 10 | 27 | 13 | 30 |  
Масив 2: 27 | 19 | 14 | 17 | 11 | 21 | 22 | 24 | 15 | 29 |  
Масив 3 (сума масивів 1 і 2): 39 | 31 | 41 | 36 | 25 | 45 | 32 | 51 | 28 | 59 |  
  
Середнє арифметичне = 38  
Мінімальне значення = 25  
Максимальне значення = 59
```

Дошка Trello:



Командна робота :



Висновок: На лабораторній: роботі №4 (еріс 4) , я ознайомилась з поняттями ,що таке Масиви,Структури даних,Вказівники а також класи даних в C++.