



## Звіт

**про виконання лабораторних та практичних робіт блоку № 4**  
**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  
Макович Маркіян Володимирович

## **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:**

Дослідження одновимірних і двовимірних масивів для зберігання і впорядкування даних, що забезпечує швидкий доступ і обробку великих обсягів інформації.

Дослідження вказівників та посилань для розуміння адресації пам'яті та оптимізації використання ресурсів, що дозволяє ефективніше працювати з динамічними структурами даних.

Дослідження динамічних масивів для створення програм із змінною кількістю елементів, що підвищує гнучкість і адаптивність коду.

Дослідження структур даних та вкладених структур для організації складних об'єктів, що забезпечує кращу структуру і читабельність програмного коду.

Дослідження алгоритмів обробки масивів і структур для реалізації ефективної обробки даних, що сприяє написанню оптимізованих і масштабованих програм.

## **Теоретичні відомості:**

У даній роботі розглядаються основні принципи роботи з масивами та структурами даних, зокрема одновимірні й двовимірні масиви для організації і зберігання великих обсягів даних. Особливу увагу приділено вказівникам і посиланням як засобам управління пам'яттю та ефективного доступу до даних. Розглянуто динамічні масиви, які забезпечують гнучке управління розміром даних під час виконання програми. Досліджено основи структур даних і вкладених структур для створення складних, логічно організованих об'єктів. Описано алгоритми обробки масивів і структур, що дозволяють ефективно виконувати операції пошуку, сортування і модифікації даних, покращуючи оптимізацію коду.

## **Виконання роботи:**

1. Опрацювання завдання та вимог до програм та середовища:

Завдання №1 VNS Lab 4 - Task 1      Варіант завдання: 19

- 1) Реалізувати з використанням масиву однонаправлене кільце (перегляд можливий зправа наліво, від першого елемента можна перейти до останнього).
- 2) Роздрукувати отриманий масив, починаючи з  $K$ -ого елемента і до  $K+1$ .
- 3) Додати в кільце перший і останній елементи.
- 4) Знищити з кільця непарні елементи.
- 5) Роздрукувати отриманий масив, починаючи з  $K$ -ого елемента й до  $K+1$ .

#### Завдання №2 VNS Lab 5 - Task 1      Варіант завдання: 19

Використовуючи функції, розв'язати зазначене у варіанті завдання. Масив повинен передаватися у функцію як параметр.

Задано рядок з  $N^2$  цифр. Встановити чи можна, розбивши рядок на підстрічки довжиною  $N$ , записати їх у рядки двовимірного масиву  $N \times N$  по одній цифрі в одному елементі так, щоб вони в першому стовпці розташувалися в порядку зростання.

#### Завдання №3 Algotester Lab 2      Варіант завдання: 2

У вас є масив  $tt$  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.

#### Завдання №4 Algotester Lab 3      Варіант завдання: 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.

#### Завдання №5 Class Practice Work

### Перевірка чи слово або число є паліндромом Задача

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 Self Practice Work

### Зуби

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.

## 2. Графічне представлення Завдання за допомогою Draw.io

### 1) Class Practice Work Варіант завдання: 19



4. Код програм з посиланням на зовнішні ресурси:

Завдання №1 VNS Lab 4 - Task 1

Варіант завдання: 19

```
1  using namespace std;
2
3  const int N = 100;
4  int arr[N] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
5  int arrSize = 10;
6  int k = 3;
7
8  int main()
9  {
10     int n = 0;
11     cout << "Array rotated by (10-k) " << arrSize - k << " positions:\n";
12     for (int i = 1; i <= arrSize; i++)
13     {
14         if (k + i >= arrSize)
15         {
16             cout << arr[n] << " ";
17             n++;
18         }
19         else
20         {
21             cout << arr[k + i] << " ";
22         }
23     }
24     cout << endl;
25     int newFirst = 15;
26     int newLast = 20;
27     arrSize += 2;
28     arr[arrSize - 1] = newLast;
29     for (int i = arrSize - 2; i > 0; i--)
30     {
31         arr[i] = arr[i - 1];
32     }
33     arr[0] = newFirst;
34     n = 1;
35     for (int i = 0; i < arrSize; i++)
36     {
37         arr[i] = arr[i + n];
38         n++;
39     }
40     n = 0;
41
42     cout << "Array rotated by " << (arrSize / 2) - k << " positions and with odd numbers deleted:\n";
43     for (int i = 1; i <= arrSize / 2; i++)
44     {
45         if (k + i >= arrSize / 2)
46         {
47             cout << arr[n] << " ";
48             n++;
49         }
50         else
51         {
52             cout << arr[k + i] << " ";
53         }
54     }
55     return 0;
56 }
```

```
4
5  const int N = 4;
6
7  bool matrix(int arr[])
8  {
9      bool pos = false;
10     int matrix[N][N] = {};
11     int count = 0;
12     for (int i = 0; i < N; i++)
13     {
14         for (int j = 0; j < N; j++)
15         {
16             matrix[i][j] = arr[count];
17             count++;
18         }
19     }
20
21     for (int i = 0; i < N - 1; i++)
22     {
23         if (matrix[i + 1][0] > matrix[i][0])
24         {
25             pos = true;
26         }
27         else
28         {
29             pos = false;
30             return pos;
31         }
32     }
33
34     return pos;
35 }
36
37 int main()
38 {
39
40     int arr[N * N] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16};
41     if (matrix(arr) == 1)
42     {
43         cout << "It is possible" << endl;
44     }
45     else
46     {
47         cout << "It is impossible" << endl;
48     }
49
50     return 0;
51 }
```

```
5  int main()
6  {
7      int N;
8      cin >> N;
9
10     int r[N] = {};
11     vector<int> newArr = {};
12     vector<int> sumArr = {};
13
14     for (int i = 0; i < N; i++)
15     {
16         cin >> r[i];
17     }
18     int a, b, c;
19     cin >> a >> b >> c;
20
21     for (int i = 0; i < N; i++)
22     {
23
24         if (r[i] != a && r[i] != b && r[i] != c)
25         {
26             newArr.push_back(r[i]);
27         }
28     }
29     if (newArr.size() == 0 || newArr.size() == 1)
30     {
31         cout << 0;
32         return 0;
33     }
34
35     for (int i = 0; i < newArr.size() - 1; i++)
36     {
37         sumArr.push_back(newArr[i] + newArr[i + 1]);
38     }
39
40     cout << sumArr.size() << endl;
41     for (int i = 0; i < sumArr.size(); i++)
42     {
43         cout << sumArr[i] << " ";
44     }
45 }
```

```
3  using namespace std;
4
5  int main()
6  {
7      int N, M, count1 = 0;
8      cin >> N;
9      int firstArr[N] = {};
10     for (int i = 0; i < N; i++)
11     {
12         cin >> firstArr[i];
13     }
14     cin >> M;
15     int secondArr[M] = {};
16     for (int i = 0; i < M; i++)
17     {
18         cin >> secondArr[i];
19     }
20
21     for (int i = 0; i < N; i++)
22     {
23         for (int j = 0; j < M; j++)
24         {
25             if (firstArr[i] == secondArr[j])
26             {
27                 count1++;
28             }
29         }
30     }
31
32     cout << count1 << endl;
33     cout << (N + M) - count1 << endl;
34
35     return 0;
36 }
```



## Завдання №5 Class Practice Work

```
1  #include <iostream>
2  #include <string>
3  #include <cmath>
4
5  using namespace std;
6
7  bool isPalindrom(string str, int start, int end)
8  {
9
10     if (start > end)
11     {
12         return true;
13     }
14     if (str[start] != str[end])
15         return false;
16
17     return isPalindrom(str, start + 1, end - 1);
18 }
19
20 int getFirstDigit(int number)
21 {
22
23     while (number >= 10)
24     {
25         number /= 10;
26     }
27
28     return number;
29 }
30 bool isPalindrom(int num)
31 {
32     bool isPal = true;
33     int start = getFirstDigit(num);
34     int end = num % 10;
35
36     if (static_cast<int>(to_string(num).length()) <= 1)
37     {
38         return true;
39     }
40     if (start != end)
```

```
41     {
42         if (start != end)
43         {
44             return false;
45         }
46         if (start == end)
47         {
48             num /= 10;
49             num %= static_cast<int>(pow(10, static_cast<int>(log10(num))));
50             return isPalindrom(num);
51         }
52         return true;
53     }
54 }
55 int main()
56 {
57     if (isPalindrom("aavvaavvaa", 0, 9) == 1)
58     {
59         cout << "(aavvaavvaa) is a palindrom" << endl;
60     }
61     if (isPalindrom(12321) == 1)
62     {
63         cout << "(12321) is a palindrom" << endl;
64     }
65
66     return 0;
67 }
```

## Завдання № 6 Self Practice Work

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

Посилання на мою гілку де знаходяться ці програми:

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/194](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/194)

## 5. Результати виконання завдань, тестування:

Завдання №1 VNS Lab 4 - Task 1      Варіант завдання: 19

```
PS C:\Users\Маркіян\Desktop\Epic 4> cd "c:\Users\Маркіян\Desktop\Epic 4"
Array rotated by (10-k) 7 positions:
4 5 6 7 8 9 0 1 2 3
Array rotated by 3 positions and with odd numbers deleted:
8 20 0 2 4 6
PS C:\Users\Маркіян\Desktop\Epic 4> █
```

Завдання №2 VNS Lab 5 - Task 1      Варіант завдання: 19

```
int arr[N * N] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16};
// ...
PS C:\Users\Маркіян\Desktop\Epic 4> cd "c:\Users\Маркіян\Desktop\Epic 4"
It is possible
PS C:\Users\Маркіян\Desktop\Epic 4> █
```

Завдання №3 Algotester Lab 2      Варіант завдання: 2

```
PS C:\Users\Маркіян\Desktop\Epic 4> cd "c:\Users\Маркіян\Desktop\Epic 4"
ych }
6
1 2 3 4 5 7
4 5 6
3
3 5 10
PS C:\Users\Маркіян\Desktop\Epic 4> █
```

Завдання №4 Algotester Lab 3      Варіант завдання: 2

```
PS C:\Users\Маркіян\Desktop\Epic 4> cd "c:\Users\Маркіян\Desktop\Epic 4"
ych }
5
1 2 3 4 5
5
4 5 6 7 8
2
8
PS C:\Users\Маркіян\Desktop\Epic 4> █
```

## Завдання №5 Class Practice Work

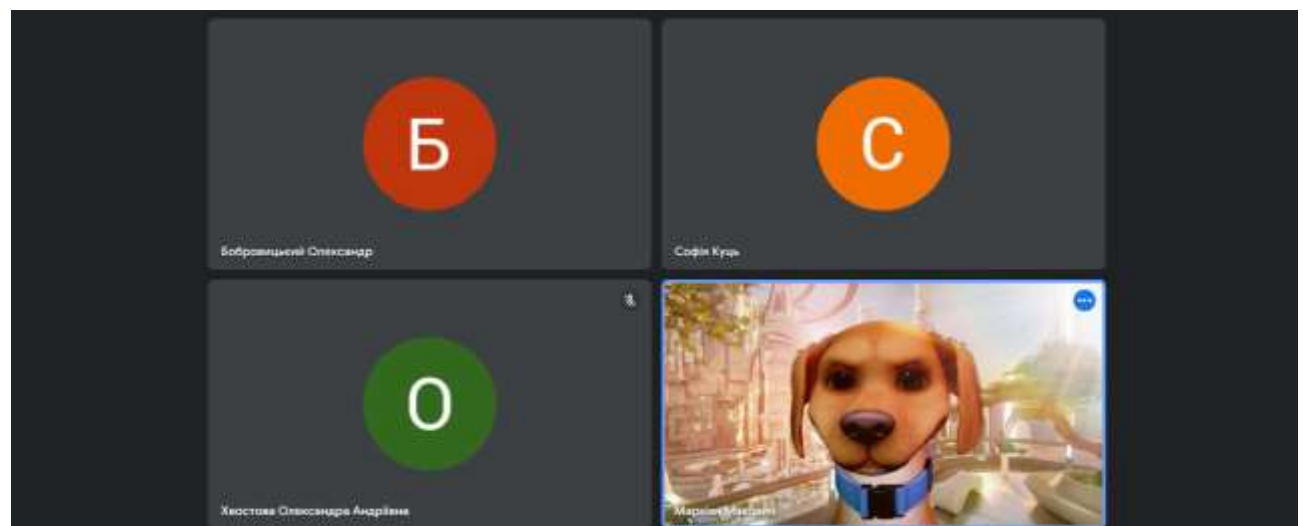
```
PS C:\Users\Маркіян\Desktop\Epic 4> cd
ian_makovych }
(aavvaavvaa) is a palindrom
(12321) is a palindrom
PS C:\Users\Маркіян\Desktop\Epic 4> █
```

## Завдання № 6 Self Practice Work

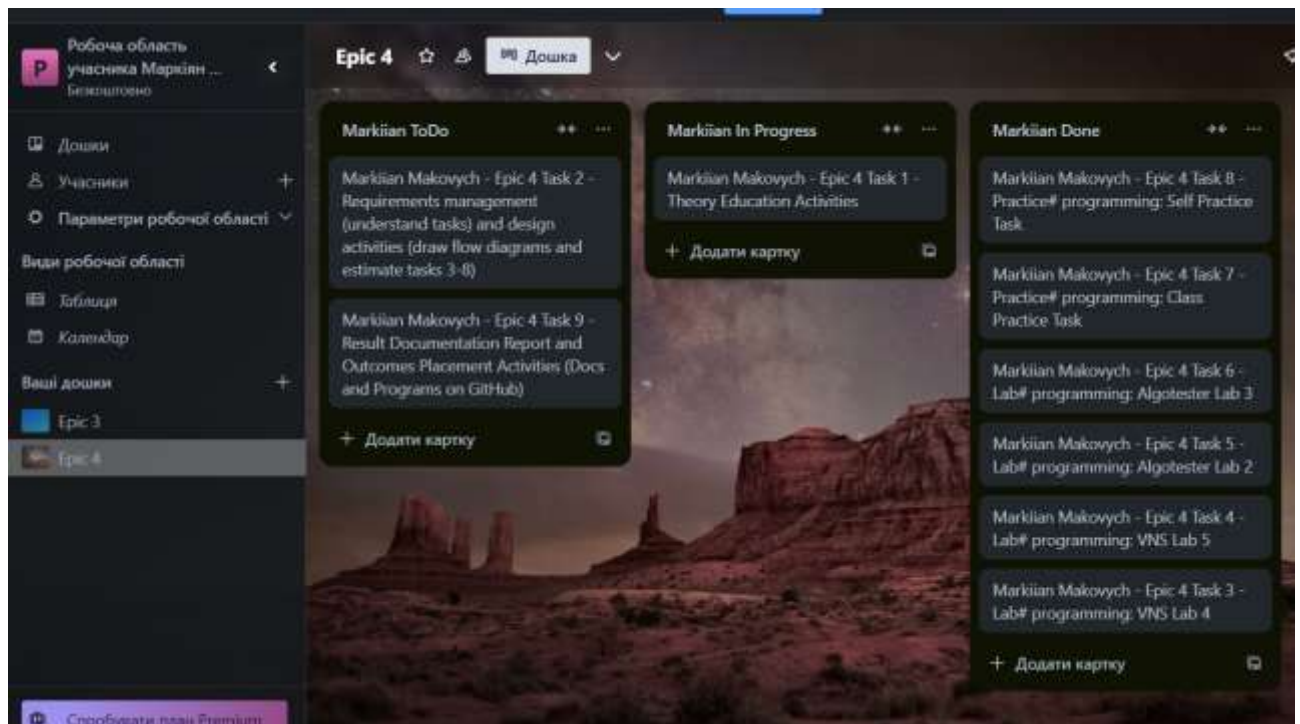
```
(12321) is a palindrom
PS C:\Users\Маркіян\Desktop\Epic 4> cd "c:\Users
_work_self_algotester_tasks_markiiian_makovych }
7 4
7 1 4 7 6 3 4
3
PS C:\Users\Маркіян\Desktop\Epic 4> █
```

## 6. Кооперація з командою:

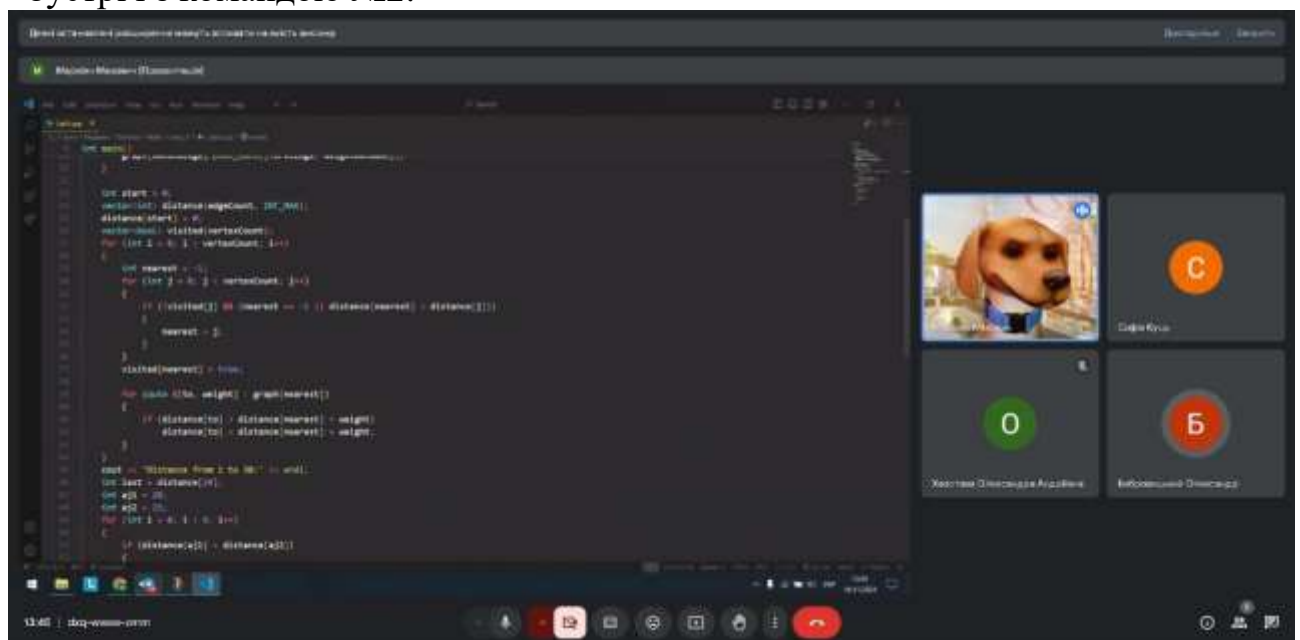
Зустріч з командою №1:



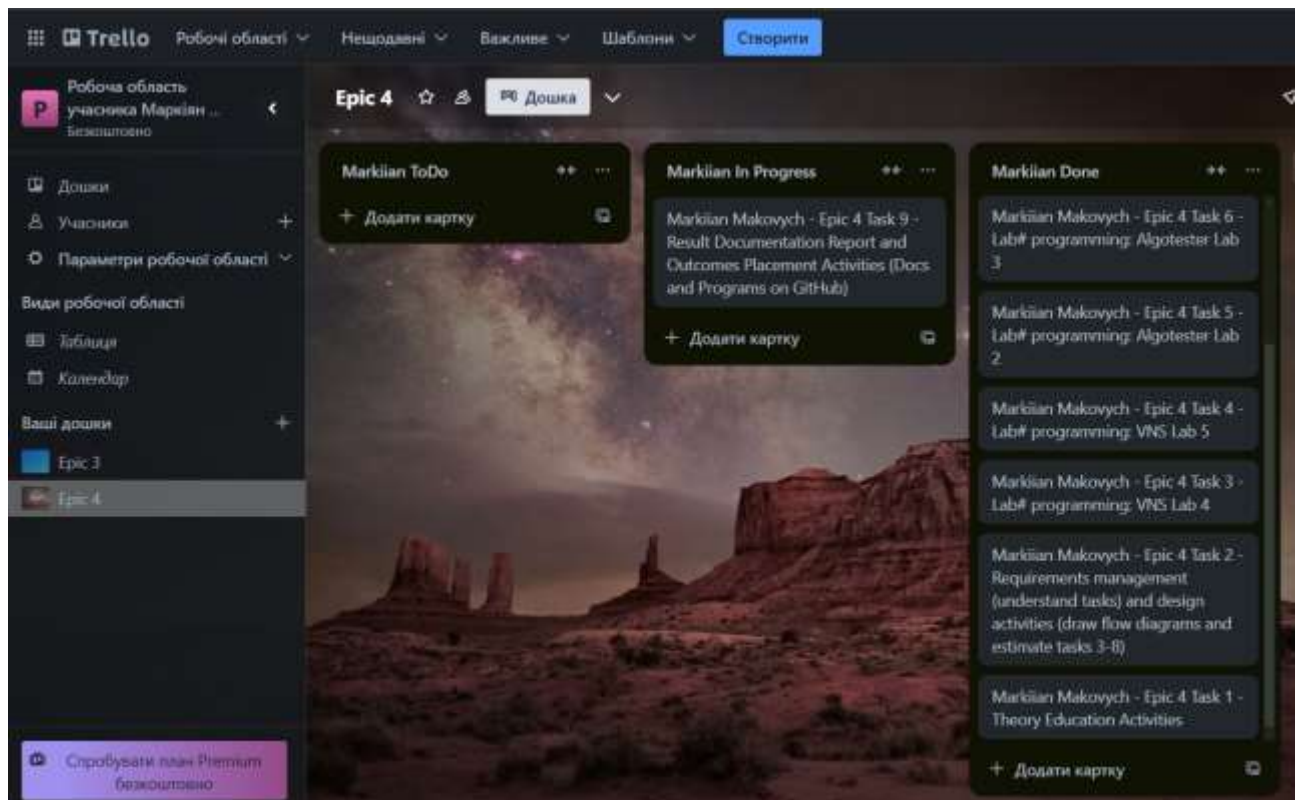
Прогрес Trello:



Зустріч з командою №2:



Прогрес Trello:



**Висновок:** В процесі виконання лабораторної роботи я навчився використовувати одновимірні та двовимірні масиви для зберігання і впорядкування даних, що покращує доступ до великого обсягу інформації. Також я ознайомився з поняттями вказівників і посилань, що дозволяє ефективно управляти пам'яттю і використовувати динамічні масиви. Окрім того, я досліджував структури даних та алгоритми обробки масивів, що сприяє написанню оптимізованих і масштабованих програм.