

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 ІІІ-11
Вербицький Юрій Віталійович

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: Навчитися ефективно працювати з одновимірними та двовимірними масивами, використовувати вказівники та посилання для оптимізації роботи з динамічними масивами, розібратися зі структурами даних і вкладеними структурами для зберігання складних даних, а також освоїти алгоритми обробки та маніпуляції масивами та структурами.

Теоретичні відомості:

1)Перелік тем:

1. Класи пам'яті у C++
2. Вступ до Масивів і Вказівників
3. Одновимірні Масиви
4. Вказівники та Посилання
5. Двовимірні Масиви
6. Динамічні Масиви
7. Структури Даних
8. Вкладені Структури
9. Використання структур
10. Алгоритми обробки та робота з масивами та структурами

2)Індивідуальний план опрацювання теорії:

1. Класи пам'яті у C++

- [C++ Storage Classes](#)
- [Memory Classes in C++](#)

2. Вступ до Масивів і Вказівників

- [Introduction to Arrays in C++](#)

- [Pointers in C++](#)

3. Одновимірні Масиви

- [C++ One-Dimensional Arrays](#)
- [Array Basics in C++](#)

4. Вказівники та Посилання

- [Pointers and References in C++](#)
- [References in C++](#)

5. Двовимірні Масиви

- [Two-Dimensional Arrays in C++](#)
- [2D Arrays in C++](#)

6. Динамічні Масиви

- [Dynamic Arrays in C++](#)
- [C++ Dynamic Memory](#)

7. Структури Даних

- [Structures in C++](#)
- [Introduction to Data Structures](#)

8. Вкладені Структури

- [Nested Structures in C++](#)
- [C++ Nested Structures](#)

9. Використання структур

- [C++ Structs and their Use](#)
- [Uses of Structures in C++](#)

10. Алгоритми обробки та робота з масивами та структурами

- [Algorithms for Arrays in C++](#)
- [Working with Arrays and Structures in C++](#)

Виконання роботи:

1)Перелік завдань:

- John Black - Epic 4 Task 1 - Theory Education Activities
- John Black - Epic 4 Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-8)
- John Black - Epic 4 Task 3 - Lab# programming: VNS Lab 4(варіант 4)
- John Black - Epic 4 Task 4 - Lab# programming: VNS Lab 5(варіант 4)
- John Black - Epic 4 Task 5 - Lab# programming: Algotester Lab 2(варіант 1)
- John Black - Epic 4 Task 6 - Lab# programming: Algotester Lab 3(варіант 3)
- John Black - Epic 4 Task 7 - Practice# programming: Class Practice Task
- John Black - Epic 4 Task 8 - Practice# programming: Self Practice Task
- John Black - Epic 4 Task 9 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)
- John Black - Epic 4 Task 10 - Results Evaluation and Release

2)Умови завдань:

Task 3:

Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.

Роздрукувати отриманий масив.

Видалити останній елемент, який дорівнює 0.

Додати після елемента масиву із заданим індексом елемент зі значенням 100.

Роздрукувати отриманий масив.

Task4: Використовуючи функції, розв'язати зазначене у варіанті завдання. Масив повинен передаватися у функцію як параметр.

Дано двовимірний масив розміром $N \times M$. Знайти в ньому підмасив розміром 3×3 , сума елементів якого максимальна. N і M можуть бути не кратними трьом.

Task5:

Lab 2v2

Limits: 1 sec., 256 MiB

У вас є масив r 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

N - кількість чисел

У другому рядку масив r , який складається з N

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

Output

У першому рядку ціле число

M - кількість чисел у масиві, який буде виведено

У наступному рядку M чисел - новий масив

Task6: Lab 3v3

Limits: 1 sec., 256 MiB

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 N . У другому рядку N цілих чисел $a_1..a_n$. У третьому рядку ціле число M у четвертому рядку M 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).

Task7: 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»).

Мета Задачі

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

Вимоги:

1. Визначення функції:
 - a. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:
 - a. *bool isPalindrome(const string& str, int start, int end);*
3. Перевантаження функцій:
 - a. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
 - b. *bool isPalindrome(ціле число);*
4. Рекурсія:
 - a. 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.

Task 8:

Lab 2v2

Обмеження: 1 сек., 256 MiB

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!

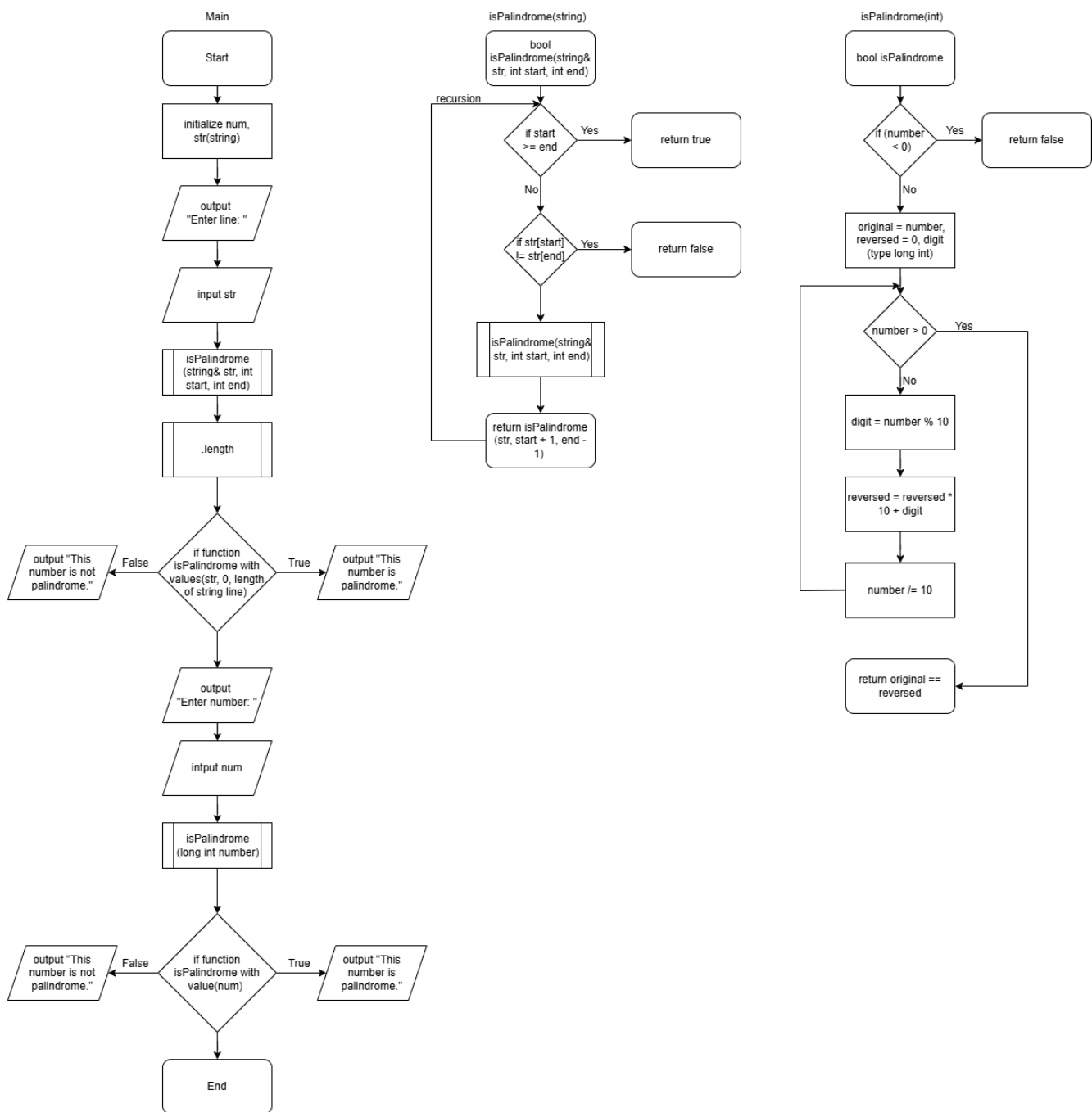
Input У першому рядку N - кількість рядків у заклятті. В наступних N рядках - набір слів w_1, \dots, w_M , розділених пробілами, де кожне слово може містити малі латинські літери та символ $\#$, який позначає пошкоджену букву.

Output

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

3)Дизайн та планова оцінка часу виконання завдань:

Task 7 - Practice# programming: Class Practice Task



Орієнтовний час виконання: 50хв

4)Код програм з посиланням на зовнішні ресурси:

Task 3 - Lab# programming: VNS Lab 4(варіант 7)

Посилання на файл програми:


```

1  #include <iostream>
2  #include <stdio.h>
3  #include <cstdlib> //random
4
5  using namespace std;
6
7  int main(){
8
9      int size; // = 9;
10     cin >> size;
11     //int array[size] = {5, 7, 98, 65, 74, 34, 67, 12, 0};
12     int array[size];
13     for(int i = 0; i < size; i++){
14         array[i] = rand() % 100;
15     }
16
17     for(int i = 0; i < size; i++){
18         cout << array[i] << " ";
19     } cout << endl;
20
21
22     for(int i = size - 1; i >= 0; i--){
23         if(array[i] == 0){
24             for(int j = i; j < size-1; j++){
25                 array[j] = array[j+1];
26             }
27             size --;
28             break;
29         }
30     }
31
32     for(int i = 0; i < size; i++){
33         cout << array[i] << " ";
34     } cout << endl;
35
36
37     int k;
38     cin >> k;
39     for(int i = size; i > k; i--){
40         array[i] = array[i-1];
41     }
42     array[k+1] = 100;
43     size ++;
44
45     for(int i = 0; i < size; i++){
46         cout << array[i] << " ";
47     }
48
49     return 0;
50 }
51

```

Task 4 - Lab# programming: VNS Lab 5(варіант 17)

Посилання на файл програми:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6
7      int N, M;
8      cin >> N >> M;
9      int array[N][M];
10
11     for(int i = 0; i < N; i++){
12         for(int j = 0; j < M; j++){
13             cin >> array[i][j];
14         }
15     }
16
17     int maxSum = INT_MIN;
18     int Row = 0, Col = 0;
19
20     for (int i = 0; i <= N - 3; i++) {
21         for (int j = 0; j <= M - 3; j++) {
22             int prevsum = 0;
23             for (int k = i; k < i + 3; k++) {
24                 for (int l = j; l < j + 3; l++) {
25                     prevsum += array[k][l];
26                 }
27             }
28             if (prevsum > maxSum) {
29                 maxSum = prevsum;
30                 Row = i;
31                 Col = j;
32             }
33         }
34     }
35     cout << endl;
36
37     cout << "MainMatrix:" << endl;
38     for(int i = 0; i < N; i++){
39         for(int j = 0; j < M; j++){
40             cout << array[i][j] << " ";
41         }
42         cout << endl;
43     }
44     cout << endl;
45
46     cout << "MaxMatrix 3*3: " << endl;
47     for (int i = Row; i < Row + 3; i++) {
48         for (int j = Col; j < Col + 3; j++) {
49             cout << array[i][j] << " ";
50         }
51         cout << endl;
52     }
53     cout << endl;
54
55     cout << "MaxSum of Matrix 3*3:" << endl;
56     cout << maxSum;
57
58
59
60     return 0;
61 }

```

Task 5 - Lab# programming: Algotester Lab 2(варіант 2)

Посилання на файл програми:

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6  int N;
7  cin >> N;
8
9  vector<int> r(N);
10 for (int i = 0; i < N; ++i) {
11     cin >> r[i];
12 }
13
14 int a, b, c;
15 cin >> a >> b >> c;
16
17 vector<int> newmass;
18 for (int i = 0; i < N; ++i) {
19     if (r[i] != a && r[i] != b && r[i] != c) {
20         newmass.push_back(r[i]);
21     }
22 }
23
24 int M = newmass.size() - 1;
25 if (M <= 0) {
26     cout << 0 << endl;
27     return 0;
28 }
29
30 vector<int> sums(M);
31 for (int i = 0; i < M; ++i) {
32     sums[i] = newmass[i] + newmass[i + 1];
33 }
34
35
36 cout << M << endl;
37 for (int sum : sums) {
38     cout << sum << " ";
39 }
40
41 cout << endl;
42
43     return 0;
44 }
```

Task 6 - Lab# programming: Algotester Lab 3(варіант 3)

Посилання на файл програми:

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  vector<int> universal_set(vector<int>& a, vector<int>& b){
8      vector<int> uniq = a;
9      for (int num : b){
10         if (find(uniq.begin(), uniq.end(), num) == uniq.end()){
11             uniq.push_back(num);
12         }
13     }
14     return uniq;
15 }
16
17
18 int main(){
19     int N, M;
20     cin >> N;
21
22     vector<int> a(N);
23     for(int i = 0; i < N; i++){
24         cin >> a[i];
25     }
26     cin >> M;
27     vector<int> b(M);
28     for(int i = 0; i < M; i++){
29         cin >> b[i];
30     }
31
32
33     int same = 0;
34     for(int i = 0; i < N; i++){
35         for(int j = 0; j < M; j++){
36             if(a[i] == b[j]){
37                 same++;
38             }
39         }
40     }
41
42
43     vector<int> uniq = universal_set(a, b);
44     int unicquantity = uniq.size();
45
46     cout << endl;
47     cout << same << endl;
48     cout << unicquantity << endl;
49     return 0;
50 }

```

Task 7 - Practice# programming: Class Practice Task

Посилання на файл програми:

```

1  #include <iostream>
2  #include <string>
3  #include <algorithm>
4
5  using namespace std;
6
7  bool isPalindrome(string& str, int start, int end){
8      if (start >= end) {
9          return true;
10     }
11     if (str[start] != str[end]) {
12         return false;
13     }
14     return isPalindrome(str, start + 1, end - 1);
15 }
16
17 bool isPalindrome(long int number) {
18     if (number < 0) {
19         return false;
20     }
21
22     long int original = number;
23     long int reversed = 0;
24
25     while (number > 0) {
26         long int digit = number % 10;
27         reversed = reversed * 10 + digit;
28         number /= 10;
29     }
30
31     return original == reversed;
32 }
33
34 int main() {
35
36     string str;
37     cout << "Enter line: ";
38     cin >> str;
39
40     if (isPalindrome(str, 0, str.length() - 1)) {
41         cout << "This line is palindrome." << endl;
42     } else {
43         cout << "This line is not palindrome." << endl;
44     }
45
46     int num;
47     cout << "Enter number: ";
48     cin >> num;
49
50     if (isPalindrome(num)) {
51         cout << "This number is palindrome." << endl;
52     } else {
53         cout << "This number is not palindrome." << endl;
54     }
55
56     return 0;
57 }

```

Task 8 - Practice# programming: Self Practice Task

Посилання на файл програми:

```

1  #include <iostream>
2  #include <vector>
3  #include <string>
4
5  using namespace std;
6
7  int main() {
8      short int N;
9      cin >> N;
10     cin.ignore();
11
12     vector<string> w(N);
13     int M = 0;
14
15     for (int i = 0; i < N; i++) {
16         getline(cin, w[i]);
17         bool suborder = false;
18
19         for (char c : w[i]) {
20             if (c == '#') {
21                 if (!suborder) {
22                     M++;
23                     suborder = true;
24                 }
25             } else {
26                 suborder = false;
27             }
28         }
29     }
30
31     cout << M << endl;
32     return 0;
33 }
34

```

5) Результати виконання завдань та фактично затрачений час

Task 3 - Lab# programming: VNS Lab 4(варіант 7)

```

5
41 67 34 0 69
41 67 34 69
1
41 67 100 34 69

```

Фактично затрачений час: 40хв

Task 4 - Lab# programming: VNS Lab 5(варіант 17)

```

5
5
1 2 3 4 5
8 7 4 2 5
3 4 7 8 9
1 0 5 4 7
5 0 3 9 7

MainMatrix:
1 2 3 4 5
8 7 4 2 5
3 4 7 8 9
1 0 5 4 7
5 0 3 9 7

MaxMatrix 3*3:
7 8 9
5 4 7
3 9 7

MaxSum of Matrix 3*3:
59

```

Фактичний час затрачений на виконання: 50хв

Task 5 - Lab# programming: Algotester Lab 2(варіант 2)

```

7
1 2 7 9 6 5 4
1 7 6
3
11 14 9

```

Фактичний час затрачений на виконання: 35хв

Task 6 - Lab# programming: Algotester Lab 3(варіант 2)

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

1
7
```

Фактичний час затрачений на виконання: 20хв

Task 7 - Practice# programming: Class Practice Task

```
Enter line: level
This line is palindrome.
Enter number: 7667
This number is palindrome.
PS C:\Users\Yura\Desktop\c++> ^C
PS C:\Users\Yura\Desktop\c++>
PS C:\Users\Yura\Desktop\c++> & 'c:\
ncher.exe' '--stdin=Microsoft-MIEngi
r-hk2tao2t.cuj' '--pid=Microsoft-MIE
Enter line: okay
This line is not palindrome.
Enter number: 1535
This number is not palindrome.
PS C:\Users\Yura\Desktop\c++> █
```

Фактичний час виконання: 30хв

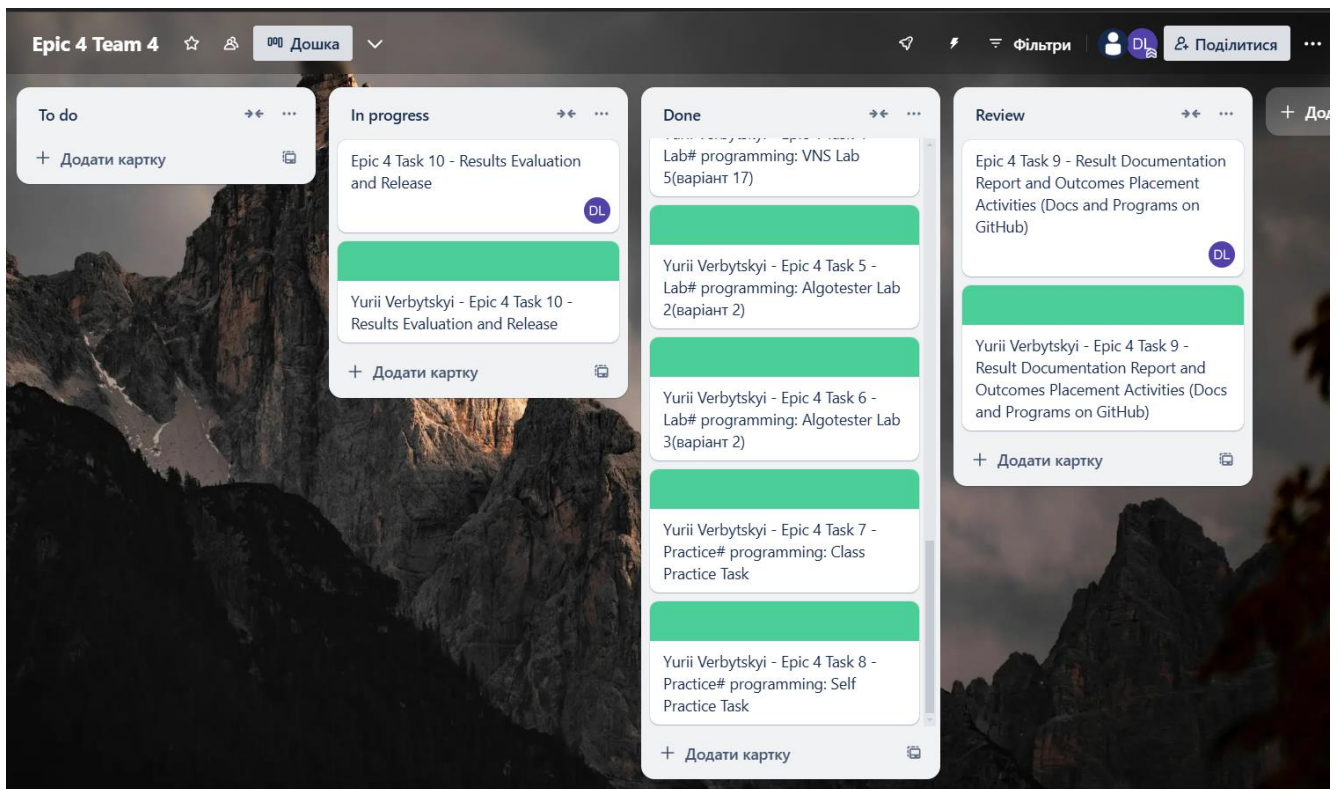
Task 8 - Practice# programming: Self Practice Task

```
3
I`m g##ing to pass #xam
I #o#e it would b# okay
Hope#ully I have time
6
```

Фактичний час виконання: 25хв

6)Робота з комадою

Trello:



Висновок: Опановуючи роботу з масивами різних типів, вказівниками, посиланнями, динамічними структурами даних та алгоритмами їх обробки, ми набуваємо необхідних знань і навичок для ефективного управління даними та їхньої оптимізації. Це дозволяє будувати більш гнучкі та продуктивні програми, здатні працювати з великими обсягами даних, зберігаючи при цьому структурованість і логічну цілісність інформації.

Посилання на пул реквест: