

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
Корнілов Артем Сергійович

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:

Закріпити на практиці вивчений теоретичний матеріал, зокрема робота з масивами, двовимірними та одновимірними, а також з вказівниками та посиланнями. Також розібратися з динамічними масивами та динамічною пам'яттю в C++.

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++:
 - Джерела інформації:
 - Статті.
<http://cpp.dp.ua/klasy-pam-yati-u-c-builder/>
 - Що опрацьовано:
 - Статична пам'ять.
 - Динамічна пам'ять.
 - Поняття стеку.
 - Виділення та вивільнення пам'яті.
- Тема №2: Вступ до Масивів і Вказівників:
 - Джерела інформації:
 - Статті.
<https://www.youtube.com/watch?v=ULdbOaMBPYc&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=40>
<https://www.youtube.com/watch?v=zopWRlYOXWw&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=57>
 - Що опрацьовано:
 - 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.

Запланований час на вивчення 1 година.
Витрачений час 1 година.

- Тема №3: Одновимірні масиви:

- 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=PLiPRE8VmJzOpn](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=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 1-25

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) Впорядкувати елементи по спаданню. 4) Знищити з кільця непарні елементи. 5) Роздрукувати отриманий масив, починаючи з K -ого елемента і до $K+1$.

Завдання №2 VNS lab 5 – task 1-25

Із двовимірного масиву в одновимірний записали спочатку рядки в довільному порядку, потім стовпці в довільному порядку. Написати програму, яка відновлює вихідний двовимірний масив маючи одновимірний, якщо відомо розмірність двовимірного масиву й елементи в ньому не повторюються..

Завдання №3 Algotester lab 2-2

Limits: 1 sec., 256 MiB

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?

Input

У першому рядку ціле число N - кількість чисел

У другому рядку масив r , який складається з N цілих чисел

Output

Єдине ціле число m - мінімальна втома, яку можна отримати

Constraints

$$1 \leq N \leq 10^5$$

$$0 \leq r_i \leq 10^5$$

Завдання №4 Algotester lab 3-3

Lab 3v3

Обмеження: 1 сек., 256 MiB

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.

Вхідні дані

У першому рядку стрічка S

Вихідні дані

Стрічка $S_{compressed}$

Обмеження

$1 \leq |S| \leq 10^5$

Завдання №5 Class practice work

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

Вимоги

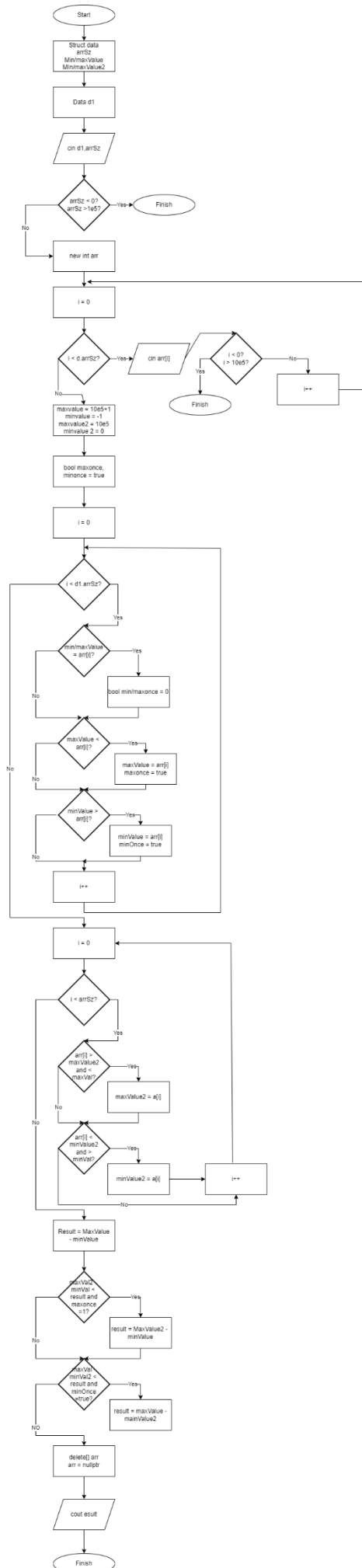
1. Визначення функції:
 1. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:
 1. *bool isPalindrome(const string& str, int start, int end);*
3. Перевантаження функцій:
 1. Перевантажте функцію *isPalindrome* для роботи з цілими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

Це програма яка зберігає числа введені користувачем та знаходить середнє значення.

Дизайн та планована оцінка часу виконання завдань:

Блок схема до завдання №3 Algotester lab 3-1



Код програм з посиланням на зовнішні ресурси

Завдання №1

```
pic4 > vns_lab_4_task_1_variant_25_artem_kornilov.cpp > print_l_to_r(int *, int &, const int)
1  #include <stdio.h>
2  #include <iostream>
3
4  using namespace std;
5
6  void print_l_to_r (int* arrey, int& index, const int size) {
7      int K = index % size;
8
9      for (int i = K; i != K + 1; i--) {
10         if (arrey[i] == arrey[size-1]) {
11             i = size - 1;
12         }
13         if (arrey[i] != 0) {
14             cout << arrey[i] << " ";
15         }
16     }
17     cout << endl;
18 }
19
20
21 int main () {
22
23     const int arrsz = 10;
24
25     int arr[arrsz] = {0, 4, 6, 3, 8, 5, 2, 1, 7, 9};
26
27     int *p = &arr[arrsz-1];
28     arr[0] = *p;
29
30     print_l_to_r(arr, 2, arrsz);
31
32     arr[0] = 0;
33
34     bool swapped;
35
36     do {
37         swapped = false;
38
39         for (int i = 0; i < arrsz; i++) {
40             for (int j = 1; j < arrsz - 1 - i; j++) {
41                 if (arr[j] < arr[j+1]) {
42                     int temp = arr[j];
43                     arr[j] = arr[j + 1];
44                     arr[j + 1] = temp;
45                     swapped = true;
46                 }
47             }
48         }
49     } while (swapped);
50
51     for (int i = 1; i < arrsz - 1; i++) {
52         if (arr[i] % 2 == 1) {
53             arr[i] = 0;
54         }
55     }
56
57     for (int j = arrsz - 1; j > 1; j--) {
58         if (arr[j] == 0) {
59             arr[j] = arr[j-1];
60             arr[j - 1] = 0;
61         }
62     }
63
64
65     p = &arr[arrsz-1];
66     arr[0] = *p;
67
68     print_l_to_r(arr, 2, arrsz);
69
70     return 0;
71 }
```


Завдання №2

```
1  #include <iostream>
2
3  using namespace std;
4
5  void ruinMatrix (const int matrixSize, int* array, int matrix[][3], int a_1, int a_2, int a_3, int b_1, int b_2, int b_3) {
6      if (a_1 >= 3 || a_1 < 0 || b_1 >= 3 || b_1 < 0 || a_2 >= 3 || a_2 < 0 || b_2 >= 3 || b_2 < 0 || a_3 >= 3 || a_3 < 0 || b_3 >= 3 || b_3 < 0) {
7          cout << "wrong input!\n";
8      }
9
10     for (int j = 0; j < matrixSize; j++) {
11         array[0+j] = matrix[a_1][j];
12     }
13
14     for (int j = matrixSize; j < matrixSize*2; j++) {
15         array[0+j] = matrix[a_2][j - matrixSize];
16     }
17
18     for (int j = matrixSize * 2; j < matrixSize * 3; j++) {
19         array[j] = matrix[a_3][j - matrixSize * 2];
20     }
21
22     for (int j = matrixSize*3; j < matrixSize*4; j++) {
23         array[0+j] = matrix[j - matrixSize*3][b_1];
24     }
25
26     for (int j = matrixSize*4; j < matrixSize*5; j++) {
27         array[0+j] = matrix[j - matrixSize*4][b_2];
28     }
29
30     for (int j = matrixSize*5; j < matrixSize*6; j++) {
31         array[0+j] = matrix[j - matrixSize*5][b_3];
32     }
33
34
35     return;
36 }
37
38 void restoreMatrix (const int matrixSize, int* array, int matrix[][3]) {
39
40     int** tempMatrix1 = new int*[matrixSize];
41     int** tempMatrix2 = new int*[matrixSize];
42     for (int i = 0; i < matrixSize; i++) {
43         tempMatrix1[i] = new int[matrixSize];
44         tempMatrix2[i] = new int[matrixSize];
45     }
46
47     for (int i = 0; i < matrixSize; i++) {
48         for (int j = matrixSize*i; j < matrixSize + matrixSize*i; j++) {
```

```

38 void restoreMatrix (const int matrixSize, int* array, int matrix[][3]) {
39     for (int i = 0; i < matrixSize; i++) {
40         for (int j = matrixSize*i; j < matrixSize + matrixSize*i; j++) {
41             tempMatrix1[i][j - matrixSize*i] = array[j];
42         }
43     }
44
45     for (int i = 0; i < matrixSize; i++) {
46         for (int j = matrixSize*i; j < matrixSize + matrixSize*i; j++) {
47             tempMatrix2[i][j - matrixSize*i] = array[j + matrixSize*3];
48         }
49     }
50
51     for (int i = 0; i < matrixSize; i++) {
52         for (int j = 0; j < matrixSize; j++) {
53             for (int i1 = 0; i1 < matrixSize; i1++) {
54                 for (int j1 = 0; j1 < matrixSize; j1++) {
55                     if (tempMatrix1[i][j] == tempMatrix2[i1][j1]) {
56                         matrix[j1][j] = tempMatrix1[i][j];
57                     }
58                 }
59             }
60         }
61     }
62
63     for (int i = 0; i < matrixSize; i++) {
64         delete[] tempMatrix1[i];
65         delete[] tempMatrix2[i];
66     }
67     delete[] tempMatrix1;
68     delete[] tempMatrix2;
69
70     return;
71 }
72
73 void print (const int size, int* array) {
74     for (int i = 0; i < size; i++) {
75         cout << array[i] << " ";
76     }
77
78     cout << endl << endl;
79
80     return;
81 }
82
83 void print (const int size, int matrix[][3]) {

```

```

92
93 void print (const int size, int matrix[][3]) {
94     for (int i = 0; i < size; i++) {
95         for (int j = 0; j < size; j++) {
96             cout << matrix[i][j] << " ";
97         }
98
99         cout << endl;
100     }
101
102     cout << endl;
103
104     return;
105 }
106
107
108
109 int main () {
110     const int matrixSize = 3;
111     const int arrSz = matrixSize*matrixSize*2;
112     int originalMatrix[matrixSize][matrixSize] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
113
114     print (matrixSize, originalMatrix);
115
116     int ruinedArray[arrSz];
117
118     ruinMatrix (matrixSize, ruinedArray, originalMatrix, 2, 1, 0, 2, 0, 1);
119
120     print (arrSz, ruinedArray);
121
122     int restoredMatrix[matrixSize][matrixSize];
123
124     restoreMatrix (matrixSize, ruinedArray, restoredMatrix);
125
126     print (matrixSize, restoredMatrix);
127
128
129     return 0;
130 }

```

Завдання №3

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main () {
6
7      struct Data
8      {
9          int arrSz;
10         int maxValue;
11         int minValue;
12         int maxValue2;
13         int minValue2;
14     };
15
16     Data d1;
17
18     cin >> d1.arrSz;
19
20     if (d1.arrSz > 100000 || d1.arrSz < 1){
21         return 0;
22     }
23
24     int* arr = new int[d1.arrSz];
25
26     for (int i = 0; i < d1.arrSz; i++){
27         cin >> arr[i];
28         if (arr[i] < 0 || arr[i] > 100000) {
29             return 0;
30         }
31     }
32
33     d1.maxValue = -1;
34     d1.minValue = 100001;
35     d1.maxValue2 = 0;
36     d1.minValue2 = 100000;
37     bool minOnce = true;
38     bool maxOnce = true;
39
40     for (int i = 0; i < d1.arrSz; i++) {
41
42         if (d1.minValue == arr[i]) {
43             minOnce = false;
44         }
45
46         if (d1.maxValue == arr[i]) {
47             maxOnce = false;
48         }

```

```

48     }
49
50     if (d1.maxValue < arr[i]) {
51         d1.maxValue = arr[i];
52         maxOnce = true;
53     }
54
55     if (d1.minValue > arr[i]) {
56         d1.minValue = arr[i];
57         minOnce = true;
58     }
59
60 }
61
62 for (int i = 0; i < d1.arrSz; i++) {
63
64     if (arr[i] > d1.maxValue2 && arr[i] < d1.maxValue) {
65         d1.maxValue2 = arr[i];
66     }
67
68     if (arr[i] < d1.minValue2 && arr[i] > d1.minValue) {
69         d1.minValue2 = arr[i];
70     }
71
72 }
73
74 unsigned int result = d1.maxValue - d1.minValue;
75
76 if (d1.maxValue - d1.minValue2 < result && minOnce) {
77     result = d1.maxValue - d1.minValue2;
78 }
79
80 if (d1.maxValue2 - d1.minValue < result && maxOnce) {
81     result = d1.maxValue2 - d1.minValue;
82 }
83
84 delete[] arr;
85
86 arr = nullptr;
87
88 cout << result;
89
90 return 0;
91 }

```

Завдання №4

```

1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  string compressString(const string& s) {
7      string compressed = "";
8      int n = s.size();
9
10     int i = 0;
11     while (i < n) {
12         char currentChar = s[i];
13         int count = 1;
14
15         while (i + 1 < n && s[i] == s[i + 1]) {
16             count++;
17             i++;
18         }
19
20         compressed += currentChar;
21         if (count > 1) {
22             compressed += to_string(count);
23         }
24
25         i++;
26     }
27
28     return compressed;
29 }
30
31 int main() {
32     string s;
33
34     cin >> s;
35
36     cout << compressString(s) << endl;
37
38     return 0;
39 }

```

Завдання №5

```

pic4 > practice_work_task_1_kornilov_artem.cpp > ...
1  #include<iostream>
2  #include<string>
3  using namespace std;
4
5  bool isPalindrome(string& text, int& start, int end){
6
7      if(start >= end)
8          return true;
9
10     if (text[start] != text[end])
11         return false;
12
13     return isPalindrome(text, ++start, --end);
14
15 }
16
17 bool isPalindrome(int& number) {
18
19     string text = to_string(number);
20
21     int start = 0;
22     int end = text.length() - 1;
23
24     return isPalindrome (text, start, end);
25 }
26
27 int main () {
28
29     string text1;
30     getline(cin, text1);
31
32     int start = 0;
33     int end = text1.length() - 1;
34
35     if(isPalindrome(text1, start, end)) {
36         cout << "string \" " << text1 << "\" is a palindrome\n";
37     }
38     else {
39         cout << "string \" " << text1 << "\" is NOT a palindrome\n";
40     }
41
42     int text2;
43     cin >> text2;
44
45     (isPalindrome(text2)) ? cout << "string \" " << text2 << "\" is a palindrome\n" : cout << "string \" " << text2 << "\" is NOT a palindrome\n";
46
47     return 0;
48 }

```

Завдання №6

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int n;
7      cout << "Enter the amount of numbers: ";
8      cin >> n;
9
10     double* numbers = new double[n];
11
12     cout << "Enter " << n << " numbers:" << endl;
13     for (int i = 0; i < n; ++i) {
14         cout << "Number " << (i + 1) << ": ";
15         cin >> numbers[i];
16     }
17
18     double sum = 0;
19     for (int i = 0; i < n; ++i) {
20         sum += numbers[i];
21     }
22     double average = sum / n;
23
24     cout << "Avarage value: " << average << endl;
25
26     delete[] numbers;
27
28     return 0;
29 }

```

**Результати виконання завдань, тестування та фактично
затрачений час:**

Завдання №1 VNS lab 4 – task 1-25

```

6 4 9 7 1 2 5 8
1 2 4 6 8

```

Фактично затрачений час: 2год

Завдання №2 VNS lab 5 – task 1-25

```

1 2 3
4 5 6
7 8 9

7 8 9 4 5 6 1 2 3 3 6 9 1 4 7 2 5 8

1 2 3
4 5 6
7 8 9

```

Фактично затрачений час: 2год

Завдання №3 Algotester lab 2-1

```

4
1 2 2 4
1

```


Фактично затрачений час: 45хв

Завдання №4 Algotester lab 3-3

```
AAAbbVCCCCC  
A3b2BC5
```

Фактично затрачений час: 30хв

Завдання №5 Class practice work

```
abdcba  
string "abdcba" is a palindrome  
123331  
string 123331 is NOT a palindrome
```

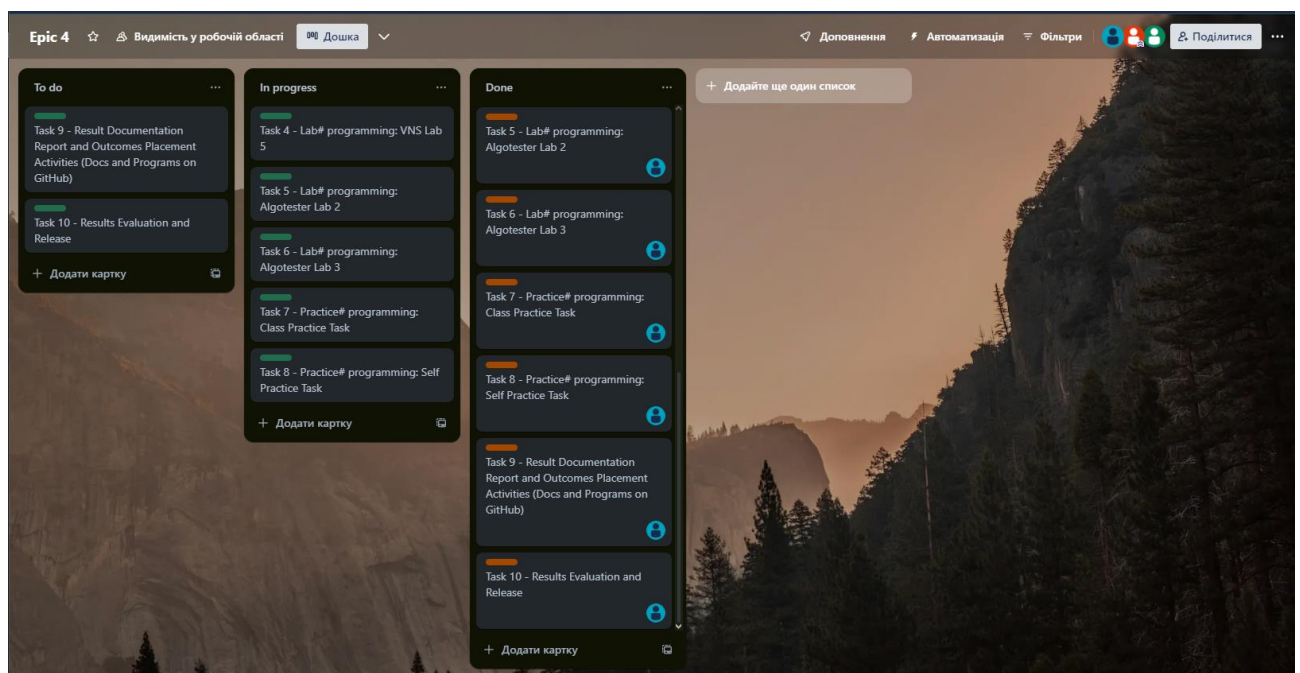
Фактично затрачений час: 1год

Завдання №6 Self-practice work

```
Enter the amount of numbers: 3  
Enter 3 numbers:  
Number 1: 2  
Number 2: 5  
Number 3: 7  
Average value: 4.66667
```

Фактично затрачений час: 30хв

Кооперація з командою:



Висновок:

Під час виконання практичних і лабораторних робіт блоку №4 я засвоїв чимало нового матеріалу, такого як: одновимірні та двовимірні масиви, вказівники і посилання, динамічні масиви та структури даних. Завдяки практичному застосуванню цих знань я краще зрозумів, як все це працює і як реалізується. Також створив блок-схему для найскладнішого завдання, що допомогло мені краще розібратися в роботі програми. Окрім цього, створив дошку в Trello для командної роботи.