

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(ла):

Студент групи ШІ-11

Зубрицький Арсеній Юрійович

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 працювати з одновимірними та двовимірними масивами, вказівниками та посиланнями, динамічними масивами та структурами даних. Застосувати на практиці вивчений теоретичний матеріал.

Теоретичні відомості:

Теоретичні відомості з переліком важливих тем:

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.

1) Індивідуальний план опрацювання теорії:

● Тема №1: Класи пам'яті у C++

- Джерела Інформації
 - <https://acode.com.ua/urok-89-dynamichne-vydilennya-pam-yati/>
 - <https://acode.com.ua/urok-111-stek-i-kupa/>
- Що опрацьовано:
 - Ознайомився з поняттям стеку та купи
 - дізнався як виділяти динамічну пам'ять та вивільняти
- Статус: Ознайомлений
- Початок опрацювання теми: 20.11.2024
- Звершення опрацювання теми: 20.11.2024
- Витрачено часу: 10 хв

● Тема №2: Вступ до Масивів і Вказівників:

- Джерела Інформації:
 - <http://cpp.dp.ua/vykorystannya-masyviv/>

- <https://acode.com.ua/urok-84-vkazivnyky/>
 - <https://acode.com.ua/urok-86-vkazivnyky-i-masyvy/>
 - <https://acode.com.ua/urok-92-posylannya/>
- Що опрацьовано:
 - Ознайомився з масивами та вказівниками, їх застосування з вказівниками
 - Ознайомився з застосуванням вказівників та посилань
- Статус: Ознайомлений
- Початок опрацювання теми: 20.11.2024
- Звершення опрацювання теми: 20.11.2024
- Витрачено часу: 15 хв
- **Тема №3:** Одновимірні Масиви:
 - Джерела Інформації:
 - <https://acode.com.ua/urok-77-masyvy/>
 - <https://acode.com.ua/urok-79-masyvy-i-tsykly/>
 - <https://acode.com.ua/urok-80-sortuvannya-masyviv-metodom-vyboru/>
 - Що опрацьовано:
 - Ознайомився з створенням масивів з їхньою індексацією присвоєнням значень та читання їх.
 - Дізнався про обхід масивів
 - Ознайомився з прикладами алгоритмів сортуванням масивів
 - Статус: Ознайомлений
 - Початок опрацювання теми: 20.11.2024
 - Звершення опрацювання теми: 20.11.2024
 - Витрачено часу: 20 хв
 - **Тема №4:** Вказівники та Посилання:
 - Джерела Інформації:
 - <https://acode.com.ua/urok-86-vkazivnyky-i-masyvy/>
 - <https://acode.com.ua/urok-87-adresna-aryfmetryka-i-indeksatsiya-masyviv/>
 - <https://acode.com.ua/urok-92-posylannya/>
 - <https://acode.com.ua/urok-89-dynamichne-vydilennya-pam-yati/>
 -
 - Що опрацьовано:
 - Ознайомився з використанням вказівників для доступу до елементів масиву, з арифметикою вказівників.
 - Дізнався про різницю між вказівниками та посиланнями в контексті функцій.
 - Ознайомився з динамічним виділенням пам'яті з використанням вказівників.
 - Статус: Ознайомлений
 - Початок опрацювання теми: 20.11.2024

- Звершення опрацювання теми: 20.11.2024
- Витрачено часу: 20 хв
- **Тема №5: Двовимірні Масиви:**
 - Джерела Інформації:
 - <https://acode.com.ua/urok-81-bagatovymirni-masyvy/>
 - https://cherto4ka.xyz/2020/02/08/func_matrix/
 - Що опрацьовано:
 - Опрацював оголошення та ініціалізація двовимірних масивів.
 - Вкладені цикли для обходу двовимірних масивів.
 - Практичні приклади використання двовимірних масивів.
 - Передачу двовимірних масивів у функції.
 - Застосування двовимірних масивів для розв'язання задач.
 - Статус: Ознайомлений
 - Початок опрацювання теми: 20.11.2024
 - Звершення опрацювання теми: 20.11.2024
 - Витрачено часу: 15 хв
- **Тема №6: Динамічні Масиви:**
 - Джерела Інформації:
 - <https://acode.com.ua/urok-89-dynamichne-vydilennya-pam-yati/>
 - <https://acode.com.ua/urok-90-dynamichni-masyvy/>
 - Що опрацьовано:
 - Ознайомився з динамічним виділенням пам'яті
 - Опрацював створення та управління динамічних масивів, Використання операторів new та delete для управління пам'яттю, Реалізація змінної розмірності масивів, Передача динамічних масивів у функції.
 - Статус: Ознайомлений
 - Початок опрацювання теми: 20.11.2024
 - Звершення опрацювання теми: 20.11.2024
 - Витрачено часу: 20 хв
- **Тема №7: Структури Даних:**
 - Джерела Інформації:
 - <https://acode.com.ua/urok-64-struktury/>
 - <https://youtu.be/6IsGVvNbyOo>
 - <https://youtu.be/5ZFgxSUKG1s>
 - Що опрацьовано:
 - Опрацював 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)

- Статус: Ознайомлений
- Початок опрацювання теми: 20.11.2024
- Звершення опрацювання теми: 20.11.2024
- Витрачено часу: 30 хв
- **Тема №8:** Вкладені Структури:
 - Джерела Інформації:
 - <https://acode.com.ua/urok-64-struktury/>
 - <https://youtu.be/dB2V9f0R9uk>
 - Що опрацьовано:
 - Опрацював поняття вкладених структур та їх оголошення.
 - Взаємодію з вкладеними структурами.
 - Використання вкладених структур для моделювання складних даних.
 - Передачу вкладених структур у функції.
 - Приклади реального використання вкладених структур.
 - Статус: Ознайомлений
 - Початок опрацювання теми: 20.11.2024
 - Звершення опрацювання теми: 20.11.2024
 - Витрачено часу: 20 хв
- **Тема №9:** Використання структур
 - Джерела Інформації:
 - <https://acode.com.ua/urok-138-perevantazhennya-operatoriv/>
 - <https://acode.com.ua/urok-141-perevantazhennya-operatoriv-vvodu-i-vyvodu/>
 - Що опрацьовано:
 - Опрацював перевантаження операторів у структурі.
 - Вивід/ввід структури (operator<<);
 - Арифметичні операції з структурами (operator+, operator-);
 - Практичні задачі на виведення структур та операції з ними
 - Статус: Ознайомлений
 - Початок опрацювання теми: 20.11.2024
 - Звершення опрацювання теми: 20.11.2024

Витрачено часу: 40 хв

- **Тема №10:** Алгоритми обробки та робота з масивами та структурами:
 - Джерела Інформації:
 - <https://www.geeksforgeeks.org/introduction-to-sorting-algorithm/>
 - <https://acode.com.ua/urok-79-masyvy-i-tsykly/>
 - Що опрацьовано:
 - Опрацював алгоритми пошуку та сортування в масивах.
 - Обробку 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.

- Статус: Ознайомлений
- Початок опрацювання теми: 20.11.2024
- Звершення опрацювання теми: 20.11.2024

Витрачено часу: 1 год

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

Завдання №1 Vns Lab_4_task_1_25

25.

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

3. Методичні вказівки

1) При виконанні роботи використовуються статичні масиви. Для організації статичних масивів із псевдозмінними межами необхідно оголосити масив досить великої довжини, наприклад, 100 елементів:

```
int N=100;
```

```
int a[N];
```

Потім користувач вводить реальну довжину масиву (не більше N) і працює з масивом тієї довжини, що він сам вказав. Інші елементи (хоча пам'ять під них і буде виділена) не розглядаються.

2) При зменшенні або збільшенні довжини масиву необхідно змінювати його реальну довжину.

Завдання №2 VNS Lab_5_task_1_25

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

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 Lab2v3

Lab 2v3

Обмеження: 1 сек., 256 MiB

Вам дано масив цілих чисел розміром N , 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 1 перелетить у клітинку з індексом a_1 , 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_i та a_{i+1} - виведіть **Stopped**

Врахуйте, що перевіряти треба також до взльоту.

Вхідні дані

У першому рядку ціле число N - розмір масиву

У другому рядку N цілих чисел - елементи масиву

Вихідні дані

У першому рядку фінальна позиція першого та другого дрона.

У другому рядку одне зі слів:

Collision

Miss

Stopped

Завдання №4 Algotester Lab3v3

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 $S_{compressed}$

Завдання №5 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.

Завдання №6 Self practice Algotester

Аркуш

Limits: 2 sec., 256 MiB

Чомусь Міші ніяк не давав спокою квадратний аркуш паперу в клітинку.

Він швидко порахував кількість маленьких квадратиків 1×1 на сторонах аркуша, і їх виявилось однакова кількість — n .

Але Міші цього було замало, і він узявся рахувати, скільки ж усього квадратів утворюють лінії розмітки аркуша.

Input

У єдиному рядку задано одне ціле число n — кількість одиничних квадратиків на сторонах аркуша (його ширина та довжина).

Output

У єдиному рядку виведіть одне ціле число — загальну кількість квадратів, зображених на аркуші.

2. Дизайн та планована оцінка часу виконання завдань:

Програма №1 Vns Lab_4_task_1_25

- Орієнтований час на реалізацію: 1 година
-

Програма №2 VNS Lab_5_task_1_25

- Орієнтований час на реалізацію: 1 година
-

Програма №3 Algotester Lab2v3

- Запланований час на реалізацію: 30 хв
-

Програма №4 Algotester Lab3v3

- Запланований час на реалізацію: 30 хв
-

Програма №5 Перевірка чи слово або число є паліндромом

- Запланований час на реалізацію: 30хв

Програма №6 Self practice Algotester

- Запланований час на реалізацію: 10 хв

4. Код програм з посиланням на зовнішні ресурси:

Завдання №1

```

1  #include <iostream>
2  #include <algorithm> // Для std::sort i std::greater
3  #include <functional> // Для std::greater
4  using namespace std;
5
6  const int N = 100; // Максимальна кількість елементів масиву
7  int a[N];          // Масив для зберігання елементів
8  int n;             // Реальна довжина масиву
9
10 // Функція для друку масиву з індексу K до K+1
11 void printArray(int startIndex, int endIndex) {
12     for (int i = startIndex; i <= endIndex && i < n; ++i) {
13         cout << a[i] << " ";
14     }
15     cout << endl;
16 }
17
18 void printArrayReverse(int startIndex) {
19     for (int i = startIndex; i >= 0; --i) {
20         cout << a[i] << " ";
21     }
22     cout << endl;
23 }
24
25 // Функція для видалення непарних елементів
26 void removeOddNumbers() {
27     int newIndex = 0; // Новий індекс для збереження парних елементів
28     for (int i = 0; i < n; ++i) {
29         if (a[i] % 2 == 0) {
30             a[newIndex++] = a[i]; // Копіюємо парні елементи
31         }
32     }
33     n = newIndex; // Зменшуємо довжину масиву до кількості парних елементів
34 }
35
36 int main() {
37     // Введення довжини масиву
38     cout << "Enter the number of elements (up to 100): ";
39     cin >> n;
40
41     // Перевірка на коректність введеної довжини
42     if (n <= 0 || n > N) {
43         cout << "Invalid size!" << endl;
44         return 1;
45     }
46
47     // Введення елементів масиву
48     cout << "Enter " << n << " elements:" << endl;
49     for (int i = 0; i < n; ++i) {
50         cin >> a[i];
51     }

```

```

53 // 1) Перегляд масиву з права наліво (можна переглядати з кінця)
54 cout << "Array in reverse order: ";
55 printArrayReverse(n - 1);
56
57 // 2) Виведення масиву, починаючи з K-ого елемента до K+1
58 int k;
59 cout << "Enter the index K to start printing from: ";
60 cin >> k;
61
62 // Перевірка коректності введеного індексу
63 if (k < 0 || k >= n) {
64     cout << "Invalid index K!" << endl;
65     return 1;
66 }
67
68 cout << "Array from index " << k << " to " << k + 1 << ": ";
69 printArray(k, k + 1);
70
71 // 3) Впорядкування елементів по спаданню
72 sort(a, a + n, greater<int>());
73 cout << "Array sorted in descending order: ";
74 printArray(0, n - 1);
75
76 // 4) Знищення непарних елементів
77 removeOddNumbers();
78 cout << "Array after removing odd elements: ";
79 printArray(0, n - 1);
80
81 // 5) Друк масиву з K-ого елемента до K+1 після видалення непарних елементів
82 if (k >= n) {
83     cout << "Index K is out of range after removing odd elements!" << endl;
84 } else {
85     cout << "Array from index " << k << " to " << k + 1 << " after removing odd elements: ";
86     printArray(k, k + 1);
87 }
88
89 return 0;
90 }
91

```

Посилання на pull-request:

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/672/files#diff-2d31306a1947fc772782663e29f0d9acdbffdeef3399a0410f14a67a0f6bde28

Завдання №2

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  // Функція для відновлення двовимірного масиву з одновимірного
8  void restoreTwoDimensionalArray(const vector<int>& oneDimArray, vector<vector<int>>& twoDimArray, int rows, int cols) {
9      int index = 0;
10
11      // Відновлення рядків
12      for (int i = 0; i < rows; ++i) {
13          for (int j = 0; j < cols; ++j) {
14              twoDimArray[i][j] = oneDimArray[index++];
15          }
16      }
17
18      // Відновлення стовпців
19      for (int j = 0; j < cols; ++j) {
20          for (int i = 0; i < rows; ++i) {
21              twoDimArray[i][j] = oneDimArray[index++];
22          }
23      }
24  }
25
26  // Функція для виведення двовимірного масиву
27  void printTwoDimensionalArray(const vector<vector<int>>& arr) {
28      for (const auto& row : arr) {
29          for (int val : row) {
30              cout << val << " ";
31          }
32          cout << endl;
33      }
34  }
35
36  int main() {
37      int rows, cols;
38
39      // Введення розмірності двовимірного масиву
40      cout << "Enter the number of rows: ";
41      cin >> rows;
42      cout << "Enter the number of columns: ";
43      cin >> cols;
44
45      // Одновимірний масив, який містить елементи з двовимірного масиву
46      vector<int> oneDimArray(rows * cols * 2); // Має бути вдвічі більше, оскільки спочатку йдуть рядки, потім стовпці
47
48      cout << "Enter the elements of the one-dimensional array (rows first, then columns):" << endl;
49      for (int i = 0; i < oneDimArray.size(); ++i) {
50          cin >> oneDimArray[i];
51      }
52
53      // Створення двовимірного масиву для відновлення
54      vector<vector<int>> twoDimArray(rows, vector<int>(cols));
55
56      // Відновлення двовимірного масиву
57      restoreTwoDimensionalArray(oneDimArray, twoDimArray, rows, cols);
58
59      // Виведення відновленого двовимірного масиву
60      cout << "The restored two-dimensional array is:" << endl;
61      printTwoDimensionalArray(twoDimArray);
62
63      return 0;
64  }

```

Посилання на pull-request:

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/672/files#diff-069e490efebb7bf9238ff652db352c7057df2c0fd69f1d3ec5f5c034a552872b

Завдання №3

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  // Функція для симуляції руху дронів
6  pair<pair<int, int>, string> simulateDrones(const vector<int>& arr, int n) {
7      int left = 1;
8      int right = n;
9
10     while (true) {
11         // Перевірка до взльоту
12         if (left == right) return {{left, right}, "Collision"};
13         if (left + 1 == right) return {{left, right}, "Stopped"};
14         if (left > right) return {{left, right}, "Miss"};
15
16         // Взліт дронів
17         left += arr[left - 1]; // Швидкість лівого дрона
18         right -= arr[right - 1]; // Швидкість правого дрона
19
20         // Перевірка після взльоту
21         if (left == right) return {{left, right}, "Collision"};
22         if (left + 1 == right) return {{left, right}, "Stopped"};
23         if (left > right) return {{left, right}, "Miss"};
24     }
25 }
26
27 int main() {
28     int n;
29     cin >> n;
30     vector<int> arr(n);
31     for (int i = 0; i < n; i++) {
32         cin >> arr[i];
33     }
34
35     auto result = simulateDrones(arr, n);
36
37     cout << result.first.first << " " << result.first.second << endl;
38     cout << result.second << endl;
39
40     return 0;
41 }
42
```

Посилання на pull-request:

https://github.com/artificial-intelligence-department/ai_programming_playground_2024

[/pull/672/files#diff-262b2f45650f4e61386ff9f1a0e906ce49ecab0bb1c433a64cc5a3120b493581](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/672/files#diff-262b2f45650f4e61386ff9f1a0e906ce49ecab0bb1c433a64cc5a3120b493581)

Завдання №4

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  string compressString(const string& s) {
6      string compressed = "";
7      int count = 1;
8
9      for (size_t i = 1; i <= s.length(); i++) {
10         // Якщо поточний символ відрізняється від попереднього або ми на останньому символі
11         if (i == s.length() || s[i] != s[i - 1]) {
12             compressed += s[i - 1]; // Додаємо символ
13             if (count > 1) {         // Додаємо кількість, якщо більше 1
14                 compressed += to_string(count);
15             }
16             count = 1;
17         } else {
18             count++;
19         }
20     }
21
22     return compressed;
23 }
24
25 int main() {
26     string s;
27     cin >> s;
28
29     string compressed = compressString(s);
30
31     cout << compressed << endl;
32
33     return 0;
34 }
35
```

Посилання на pull-request:

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/672/files#diff-ea977296a6e85ac7a90d83b57facb5523af6f19cc2fffe78a23adbb2fce32af6

Завдання №5


```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  // Рекурсивна функція для перевірки, чи є слово паліндромом
6  bool isPalindrome(string str, int start, int end) {
7      // Базовий випадок: якщо початкова позиція більша або рівна кінцевій
8      if (start >= end) {
9          return true;
10     }
11
12     // Якщо символи не співпадають, то це не паліндром
13     if (str[start] != str[end]) {
14         return false;
15     }
16
17     // Перевіряємо наступні символи рекурсивно
18     return isPalindrome(str, start + 1, end - 1);
19 }
20
21 // Перевантажена функція для перевірки числа на паліндром
22 bool isPalindrome(int num) {
23     int reversed = 0, original = num;
24
25     // Якщо число від'ємне, то це не може бути паліндромом
26     if (num < 0) {
27         return false;
28     }
29
30     // Зворотне число для порівняння
31     while (num > 0) {
32         int digit = num % 10; // остання цифра
33         reversed = reversed * 10 + digit;
34         num /= 10; // відкидаємо останню цифру
35     }
36
37     // Якщо число і його зворотне значення співпадають, то це паліндром
38     return original == reversed;
39 }
40
41 int main() {
42     // Перевірка на паліндром для слова
43     string word;
44     cout << "Enter a word: ";
45     cin >> word;
46
47     if (isPalindrome(word, 0, word.length() - 1)) {
48         cout << "\"" << word << "\" is a palindrome." << endl;
49     } else {
50         cout << "\"" << word << "\" is not a palindrome." << endl;
51     }
52
53     // Перевірка на паліндром для числа
54     int number;
55     cout << "Enter a number: ";
56     cin >> number;
57
58     if (isPalindrome(number)) {
59         cout << number << " is a palindrome." << endl;
60     } else {
61         cout << number << " is not a palindrome." << endl;
62     }
63
64     return 0;
65 }

```

Посилання на pull-request:

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/672/files#diff-0a6680fdada9a2c6468a52de92000004d04feec8d2bf596034d2f2bdbf6171a0

Завдання №6

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n;
6      cin >> n;
7
8      int totalSquares = 0;
9
10     for (int k = 1; k <= n; k++) {
11         totalSquares += (n - k + 1) * (n - k + 1);
12     }
13
14     cout << totalSquares << endl;
15
16     return 0;
17 }
```

Посилання на pull-request:

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/672/files#diff-1049d0f832e4331e92bf24756965bab1638618b95c4e71079b7f69e26fbdf778

5. Результати виконання завдань, тестування та фактично затрачений час:

Завдання №1

```

Enter the number of elements (up to 100): 7
Enter 7 elements:
2 3 4 9 8 6 5
Array in reverse order: 5 6 8 9 4 3 2
Enter the index K to start printing from: 4
Array from index 4 to 5: 8 6
Array sorted in descending order: 9 8 6 5 4 3 2
Array after removing odd elements: 8 6 4 2
Index K is out of range after removing odd elements!

```

Час затрачений на виконання завдання: 2 год

Завдання №2

```

Enter the number of rows: 3
Enter the number of columns: 3
Enter the elements of the one-dimensional array (rows first, then columns):
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
The restored two-dimensional array is:
1 4 7
2 5 8
3 6 9

```

Час затрачений на виконання завдання: 2 год

Завдання №3

```

5
1 2 3 7 4
2 1
Miss

```

Час затрачений на виконання завдання: 1 год

Завдання №4

```

AAAQWETYSSSIIIAA
A3QWETYS3I3A2

```

Час затрачений на виконання завдання: 1 год

Завдання №5

```
Enter a word: ОКЖКО
"ОКЖКО" is a palindrome.
Enter a number: 102939201
102939201 is a palindrome.
```

Час затрачений на виконання завдання: 2 год

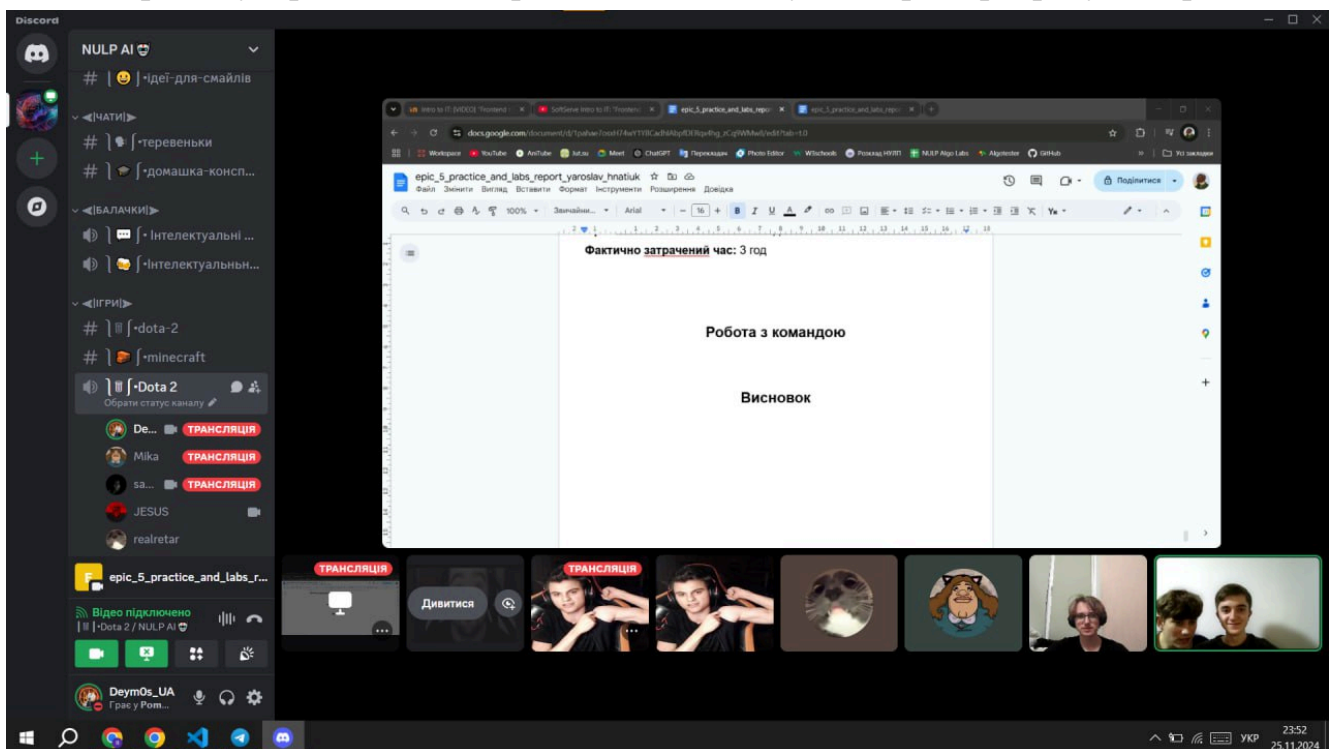
Завдання №6

```
4
30
```

Час затрачений на виконання завдання: 20хв

6. Кооперація з командою:

- Скрін з зустрічі по обговоренню задач Епіку та Скрін прогресу по Тrello



Висновки:

Під час роботи над 4 епіком я навчився працювати з одновимірними масивами: сортували, фільтрували, та виводили їх у різних форматах. Також навчився працювати з двовимірними масивами, вказівниками та посиланнями, динамічними

масивами та структурами даних. Та застосував на практиці весь вивчений теоретичний матеріал.