

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

Виконала:

Студентка групи ІІІ-13
Осінна Єлизавета Сергіївна

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:

Одержання навичок обробки одновимірних масивів. Організувати обробку масивів з використанням функцій, навчитися передавати масиви як параметри функцій.

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.

1) Індивідуальний план опрацювання теорії:

- [C++ • Теорія • Урок 40 • Одновимірні масиви](#)
- [Структури в C++ / Уроки по C++ / aCode](#)
- [Масиви в C++ / Уроки по C++ / aCode](#)
- [Двовимірні масиви в C++ / aCode](#)
- [Динамічне виділення пам'яті в C++ / Уроки по C++ / aCode](#)
-

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

Завдання № 1 VNS Lab 4

- Варіант 5
- Деталі завдання
- Важливі деталі для врахування в імплементації програми

Завдання № 2 VNS Lab 5

- Варіант 5
- Деталі завдання^

Написати функцію для знищення рядка із двовимірною масиву. Рядки, що залишилися, повинні бути розташовані щільно, елементи яких бракує замінюються 0. За допомогою розроблених функцій знищити з масиву рядки з номерами від А до В.

Завдання № 3 Algotester Lab 2

- Варіант 2
- Деталі завдання:

У вас є масив g 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.

Вхідні дані

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

Вихідні дані

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

У наступному рядку M чисел - новий масив

Обмеження

$$1 \leq N \leq 105$$

$$0 \leq r_i \leq 105$$

$$0 \leq a, b, c \leq 105$$

Завдання № 4 Algotester Lab 3

- Варіант 2

- Деталі завдання:

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.

Вхідні дані

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 $b_1..b_n$

Вихідні дані

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).

Обмеження

$$0 \leq N \leq 100$$

$$0 \leq a_i \leq 100$$

$$0 \leq M \leq 100$$

$$0 \leq b_i \leq 100$$

Завдання № 5 Class Practice Work

- Деталі завдання

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 Work

- Деталі завдання

Lab 3v3

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.

Input

У першому рядку стрічка S

Output

Стрічка Scompressed

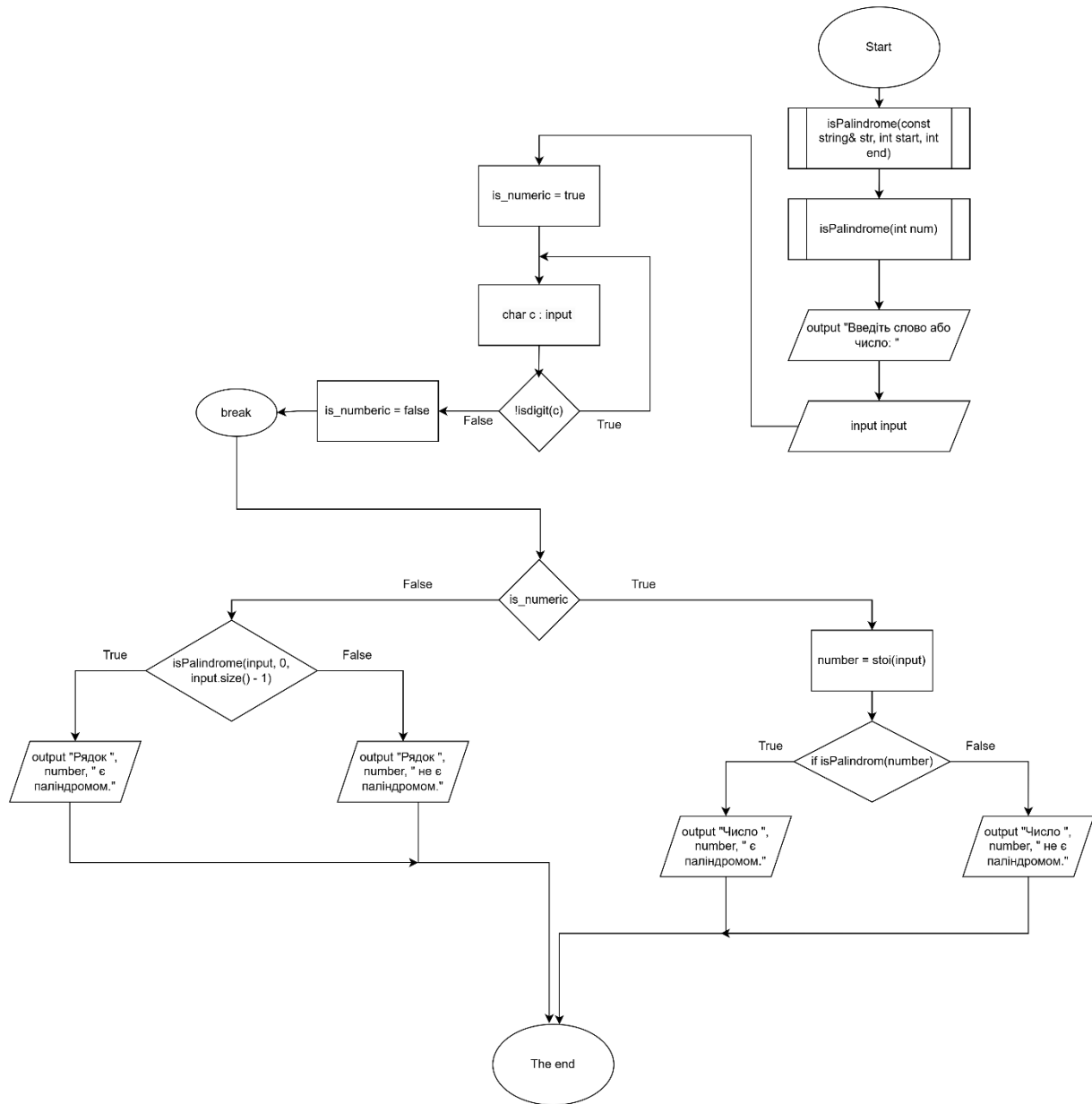
Constraints

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

2. Дизайн та планована оцінка часу виконання завдань:

Програма № 5 Class Practice Work

- Блок-схема



- Планований час на реалізацію кожної програми: 45 хв

4. Код програм з посиланням на зовнішні ресурси:

Завдання № 1 VNS Lab 4

```
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;
```

```

int main() {
    // Ініціалізуємо генератор випадкових чисел
    srand(time(0));

    // 1. Створення одновимірної динамічної масиви цілих чисел
    const int SIZE = 20; // Початковий розмір масиву
    int* array = new int[SIZE];
    int current_size = SIZE;

    // Заповнення масиву випадковими числами від 0 до 99
    for (int i = 0; i < current_size; ++i) {
        array[i] = rand() % 100;
    }

    // 2. Роздрукувати отриманий масив
    cout << "Початковий масив:" << endl;
    for (int i = 0; i < current_size; ++i) {
        cout << array[i] << " ";
    }
    cout << endl;

    // 3. Видалення елементів, кратних 7
    // Підрахунок кількості елементів, які залишаться
    int new_size = 0;
    for (int i = 0; i < current_size; ++i) {
        if (array[i] % 7 != 0) {
            ++new_size;
        }
    }

    // Створюємо новий масив для збереження елементів, які не є кратними 7
    int* filtered_array = new int[new_size];
    int index = 0;
    for (int i = 0; i < current_size; ++i) {
        if (array[i] % 7 != 0) {
            filtered_array[index++] = array[i];
        }
    }

    // Оновлення розміру і заміна масиву
    delete[] array;
    array = filtered_array;
    current_size = new_size;

    // 4. Додавання 0 після кожного непарного елемента
    // Підрахунок нового розміру масиву з урахуванням вставок
    new_size = current_size;
    for (int i = 0; i < current_size; ++i) {
        if (array[i] % 2 != 0) {

```

```

        ++new_size; // Кожен непарний елемент потребує додаткового місця для 0
    }
}

// Створення нового масиву для зберігання результату
int* modified_array = new int[new_size];
index = 0;
for (int i = 0; i < current_size; ++i) {
    modified_array[index++] = array[i];
    if (array[i] % 2 != 0) {
        modified_array[index++] = 0; // Додаємо 0 після непарного елемента
    }
}

// Оновлення розміру і заміна масиву
delete[] array;
array = modified_array;
current_size = new_size;

// 5. Роздрукувати отриманий масив
cout << "Змінений масив:" << endl;
for (int i = 0; i < current_size; ++i) {
    cout << array[i] << " ";
}
cout << endl;

// Звільнення пам'яті
delete[] array;

return 0;
}

```

Результат:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4> & 'c:\Users\ser
ebugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-de3hmyoq.f4t'
rosoft-MIEngine-Error-k5wmoyrw.3w4' '--pid=Microsoft-MIEngine-Pid-scurmhko.wul' '--dbgEx
Початковий масив:
33 51 60 94 93 45 16 50 30 95 98 80 57 91 96 82 59 17 22 92
Змінений масив:
33 0 51 0 60 94 93 0 45 0 16 50 30 95 0 80 57 0 96 82 59 0 17 0 22 92
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4>

```

Час затрачений на виконання завдання: 1 год

Завдання № 2 VNS Lab 5

```
#include <iostream>
```



```

using namespace std;

// Функція для видалення рядків з індексів A до B
void deleteRows(int** matrix, int& rows, int cols, int A, int B) {
    int shift = B - A + 1; // Кількість рядків для видалення

    // Зсуваємо рядки, які залишились, вгору
    for (int i = A; i + shift < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            matrix[i][j] = matrix[i + shift][j];
        }
    }

    // Заповнюємо останні рядки нулями
    for (int i = rows - shift; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            matrix[i][j] = 0;
        }
    }

    // Оновлюємо кількість рядків
    rows -= shift;
}

int main() {
    int rows = 6, cols = 5; // Початковий розмір матриці
    int A = 1, B = 3;       // Індeksi рядків для видалення

    // Створення двовимірного динамічного масиву
    int** matrix = new int*[rows];
    for (int i = 0; i < rows; ++i) {
        matrix[i] = new int[cols];
    }

    // Ініціалізація матриці
    int counter = 1;
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            matrix[i][j] = counter++;
        }
    }

    // Вивід початкової матриці
    cout << "Початкова матриця:" << endl;
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }
}

```

```

// Видалення рядків з A до B
deleteRows(matrix, rows, cols, A, B);

// Вивід оновленої матриці
cout << "\nОновлена матриця після видалення рядків:" << endl;
for (int i = 0; i < rows; ++i) {
    for (int j = 0; j < cols; ++j) {
        cout << matrix[i][j] << " ";
    }
    cout << endl;
}

// Видалення динамічної пам'яті
for (int i = 0; i < rows + (B - A + 1); ++i) {
    delete[] matrix[i];
}
delete[] matrix;

return 0;
}

```

Результат:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

rossoft-MIEngine-Error-dysm2iex.kbv' '--pid=Microsoft-MIEngine-Pid-ecpgluk3.crp' '--db
Початкова матриця:
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
26 27 28 29 30

Оновлена матриця після видалення рядків:
1 2 3 4 5
21 22 23 24 25
26 27 28 29 30
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4>

```

Час затрачений на виконання завдання: 1 год

Завдання № 3 Algotester Lab 2

```

#include <iostream>
using namespace std;

int main() {
    int N;
    cin >> N;

    int* r = new int[N];

```

```

for (int i = 0; i < N; ++i) {
    cin >> r[i];
}

int a, b, c;
cin >> a >> b >> c;

// Масив для фільтрованих елементів (максимальний розмір такий самий, як у `r`)
int* filtered = new int[N];
int new_size = 0;

// Фільтрація елементів без використання set
for (int i = 0; i < N; ++i) {
    if (r[i] != a && r[i] != b && r[i] != c) {
        filtered[new_size++] = r[i];
    }
}

// Масив для сум сусідніх елементів
if (new_size > 1) {
    int* sums = new int[new_size - 1];
    for (int i = 0; i < new_size - 1; ++i) {
        sums[i] = filtered[i] + filtered[i + 1];
    }

    // Вивід результатів
    cout << (new_size - 1) << endl;
    for (int i = 0; i < new_size - 1; ++i) {
        cout << sums[i] << " ";
    }
    cout << endl;

    delete[] sums; // Звільняємо пам'ять для масиву сум
} else {
    // Якщо не вистачає елементів для створення пар
    cout << 0 << endl;
}

// Звільнення пам'яті
delete[] r;
delete[] filtered;

return 0;
}

```

Результат:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4> & 'c:\Users\serhe\.vs
ebugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-krzin0mz.11p' '--st
rosoft-MIEngine-Error-gpxywwcp.u5v' '--pid=Microsoft-MIEngine-Pid-2k4bowo3.24y' '--dbgExe=C:\V
5
3 7 8 9 2
3 2 9
1
15
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4> |
```

Час затрачений на виконання завдання: 1 год

Завдання № 4 Algotester Lab 3

```
#include <iostream>
using namespace std;

int main(){
    int N, M;

    // Зчитування масиву a
    cin >> N;
    int* a = new int[N];
    for (int i = 0; i < N; i++) {
        cin >> a[i];
    }

    // Зчитування масиву b
    cin >> M;
    int* b = new int[M];
    for (int i = 0; i < M; i++) {
        cin >> b[i];
    }

    // Знаходимо кількість спільних елементів
    int common_elements = 0;
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            if (a[i] == b[j]) {
                common_elements++;
                break; // Переходимо до наступного елемента з a
            }
        }
    }

    // Знаходимо кількість унікальних елементів
    int* union_array = new int[N + M];
    int unique_count = 0;
```

```

// Додаємо елементи з a
for (int i = 0; i < N; i++) {
    bool is_unique = true;
    for (int j = 0; j < unique_count; j++) {
        if (union_array[j] == a[i]) {
            is_unique = false;
            break;
        }
    }
    if (is_unique) {
        union_array[unique_count++] = a[i];
    }
}

// Додаємо елементи з b
for (int i = 0; i < M; i++) {
    bool is_unique = true;
    for (int j = 0; j < unique_count; j++) {
        if (union_array[j] == b[i]) {
            is_unique = false;
            break;
        }
    }
    if (is_unique) {
        union_array[unique_count++] = b[i];
    }
}

// Вивід результатів
cout << common_elements << endl;
cout << unique_count << endl;

// Звільнення пам'яті
delete[] a;
delete[] b;
delete[] union_array;

return 0;
}

```

Результат:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

ebugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-wxnmiyuy.n
rosoft-MIEngine-Error-l1l4nxe0.mq0' '--pid=Microsoft-MIEngine-Pid-4n12atil.l2s' '--d
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4> ^C
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4>
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4> & 'c:\Users
ebugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-t3bhnt2m.5
rosoft-MIEngine-Error-dkoopu5n.j1y' '--pid=Microsoft-MIEngine-Pid-pigumjrm.zjb' '--d
8
4 5 8 9 6 2 3 1
3
4 7 10
1
10
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4> |
```

Час затратений на виконання завдання: 1 год

Завдання № 5 Class Practice Work

```
#include <iostream>
#include <string>
using namespace std;

bool isPalindrome(const string& str, int start, int end) {
    if (start >= end) {
        return true;
    }
    if (str[start] != str[end]) {
        return false;
    }
    return isPalindrome(str, start + 1, end - 1);
}

bool isPalindrome(int num) {
    if (num < 0) return false;

    int reversed = 0, original = num;

    while (num > 0) {
        reversed = reversed * 10 + num % 10;
        num /= 10;
    }

    return original == reversed;
}

int main() {
```

```

    cout << "Введіть слово або число: ";
    string input;
    cin >> input;

    bool is_numeric = true;

    for (char c : input) {
        if (!isdigit(c)) {
            is_numeric = false;
            break;
        }
    }

    if (is_numeric) {
        int number = stoi(input);
        if (isPalindrome(number)) {
            cout << "Число " << number << " є паліндромом." << endl;
        } else {
            cout << "Число " << number << " не є паліндромом." << endl;
        }
    } else {
        if (isPalindrome(input, 0, input.size() - 1)) {
            cout << "Рядок \"" << input << "\" є паліндромом." << endl;
        } else {
            cout << "Рядок \"" << input << "\" не є паліндромом." << endl;
        }
    }

    return 0;
}

```

Результат:

```

PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4>
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4> & 'c:\Users\serhe\
ebugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-3vsmychk.ftt' '--s
rosoft-MIEngine-Error-vxlh2bro.ukx' '--pid=Microsoft-MIEngine-Pid-krkzlpes.kih' '--dbgExe=C:
Введіть слово або число: 126367
Число 126367 не є паліндромом.
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4> ^C
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4>
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4> & 'c:\Users\serhe\
ebugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-sab22xpj.fp3' '--s
rosoft-MIEngine-Error-pdw0tix4.z0c' '--pid=Microsoft-MIEngine-Pid-z3od5nuu.g3n' '--dbgExe=C:
Введіть слово або число: radar
Рядок "radar" є паліндромом.
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4>

```

Час затрачений на виконання завдання: 1 год

Завдання № 6 Self Practice Work

```

#include <iostream>
#include <cstring>

```

```

using namespace std;

void compressString(const char* s, char* compressed) {
    int n = strlen(s);
    int pos = 0;

    for (int i = 0; i < n; ++i) {
        char currentChar = s[i];
        int count = 1;

        while (i + 1 < n && s[i + 1] == currentChar) {
            ++count;
            ++i;
        }

        compressed[pos++] = currentChar;

        if (count > 1) {
            char countStr[10];
            sprintf(countStr, "%d", count);

            for (int j = 0; countStr[j] != '\0'; ++j) {
                compressed[pos++] = countStr[j];
            }
        }

        compressed[pos] = '\0';
    }
}

int main() {
    char s[100001];
    cin >> s;

    char compressed[200001];

    compressString(s, compressed);

    cout << compressed << endl;

    return 0;
}

```

Результат:

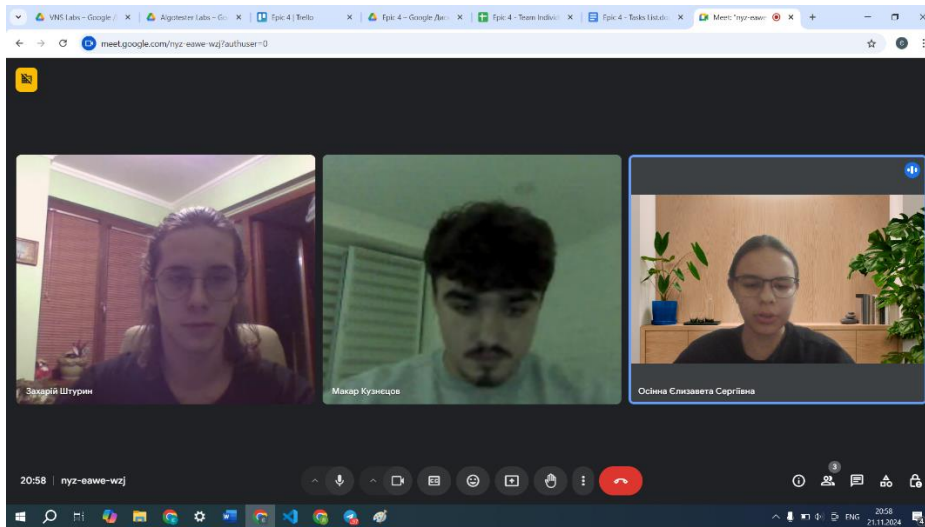
```

PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4>
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4> & 'c:\Users\serhe\.vscode\extensions\ms-vscode
ebugAdapters\bin\windowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-y0qepmfl.fmo' '--stdout=Microsoft-MIEngine-Output
rosoft-MIEngine-Error-awu2s0hr.lcw' '--pid=Microsoft-MIEngine-Pid-kduw4iaw.vls' '--dbgExe=C:\msys64\mingw64\bin\gdb.exe'
vvvvvjjrrrrttt
v6j3r2t4
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_4>

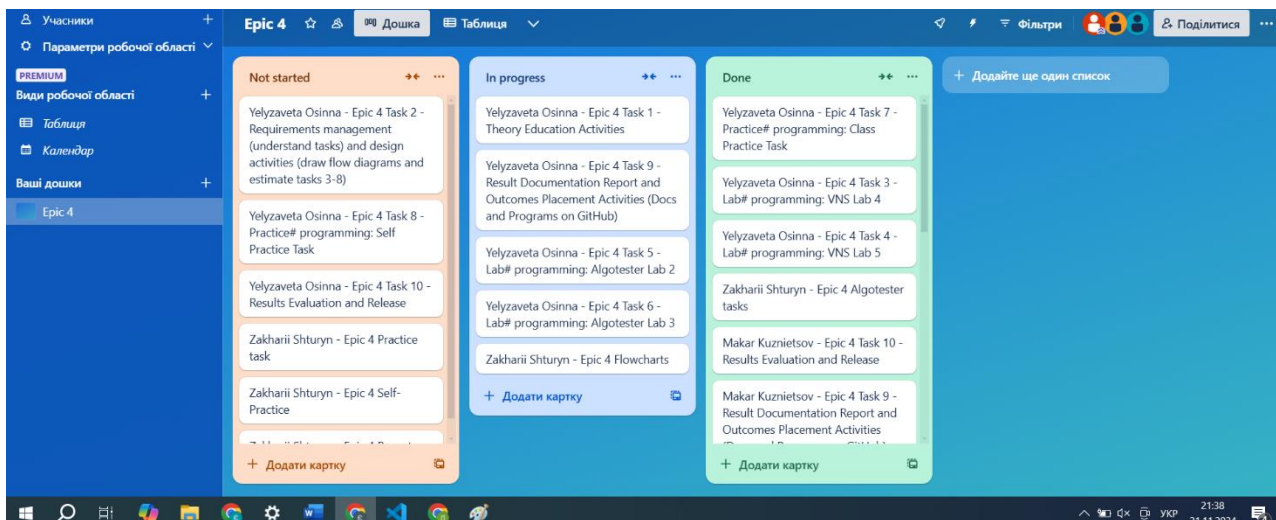
```


Час затрачений на виконання завдання: 1 год

6. Кооперація з командою:



Trello:



Висновки:

Виконуючи цю роботу я одержала навички обробки одновимірних масивів, обробки масивів з використанням функцій, навчилася передавати масиви як параметри функцій.