

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

ро 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
Михальчук Антон Євгенійович

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:

Метою даної роботи є вивчення та закріплення теоретичних і практичних знань щодо використання та маніпулювання одновимірними та двовимірними масивами, а також розуміння принципів роботи з вказівниками, посиланнями та динамічними масивами. Окрім того, передбачається розгляд та застосування основних структур даних, включаючи вкладені структури, а також реалізація алгоритмів обробки масивів та структур для ефективного вирішення практичних задач програмування.

Теоретичні відомості:

- 1) Теоретичні відомості з переліком важливих тем:
 - Тема №*.1: C++ Arrays, Data structures
- 2) Індивідуальний план опрацювання теорії:
 - Тема №*.1: C++ Basics
 - o Джерела Інформації
 - Відео. <https://www.youtube.com/watch?v=2UDMGCCcRCjo>
 - Стаття. <https://www.w3schools.com/cpp/>
 -
 - o Що опрацьовано:
 - Вивчив базовий синтаксис та семантику мови C++.
 - Особливу увагу приділяв таким темам, як масиви
 - o Статус: Ознайомлений
 - o Початок опрацювання теми: 15.09.2024
 - o Звершення опрацювання теми: 11.11.2024

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

Завдання №1 VNS Lab 4 Варіант: 9

- Деталі завдання:
 - o Сформулювати одновимірний масив цілих чисел, використовуючи генератор
 - o випадкових чисел.
 - o Роздрукувати отриманий масив.
 - o Знищити перший елемент із заданим значенням.
 - o Зсунути масив циклічно на K елементів вправо.
 - o Роздрукувати отриманий масив.
- Важливі деталі для врахування в імплементації програми:

- При виконанні роботи використовуються статичні масиви. Для організації статичних масивів із псевдозмінними межами необхідно оголосити масив досить великої довжини, наприклад, 100 елементів.
- При зменшенні або збільшенні довжини масиву необхідно змінювати його реальну довжину.

Завдання №2 VNS Lab 5 Варіант: 9

- Деталі завдання:
 - Написати функцію для обчислення суми елементів квадратної матриці, які розташовані нижче головної діагоналі. З її допомогою знайти максимальне значення такої суми в n матрицях.

Завдання №3 Algotester Lab 2 Варіант: 3

- Деталі завдання:

Lab 2v3

Обмеження: 1 сек., 256 МБ

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 опиняться 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} - введіть **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

Обмеження

$$1 \leq N \leq 1000$$

$$1 \leq a_i \leq 5$$

Гарантується, що дрони не можуть вилетіти за межі $[1, N]$ (тобто не існує масиву, в якому дрони влітають за межі до того, як станеться одна з 3х вищезазначених ситуацій) та їх швидкість завжди більша за нуль.

Завдання №4 Algotester Lab 3 Варіант: 2

- Деталі завдання:

Lab 3v2

Обмеження: 1 сек., 256 МБ

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 $a_1 \dots a_n$

У третьому рядку ціле число M

у четвертому рядку M цілих чисел $b_1 \dots b_n$

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 Task

- 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»).

Вимоги:

- Визначення функції:
 - Реалізуйте рекурсивну функцію `isPalindrome`, яка перевіряє, чи заданий рядок є паліндромом.
- Приклад визначення функції:
 - `bool isPalindrome(const string& str, int start, int end);`
- Перевантаження функцій:
 - Перевантажте функцію `isPalindrome` для роботи з цілими значеннями.
- `bool isPalindrome(ціле число);`
- Рекурсія:
 - 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 Task Непарний масив

- Деталі завдання:

Непарний масив

Обмеження: 2 сек., 256 МБ

Зенік і Марічка мають масив з n цілих чисел a_i . Вони хочуть перепорядкувати елементи масиву так, щоб сума будь-яких двох послідовних елементів була непарною.

Будь ласка, допоможіть їм зробити це або скажіть, що це неможливо.

Вхідні дані

Перший рядок містить ціле число n — розмір масиву.

Другий рядок містить n цілих чисел a_i — елементи масиву.

Вихідні дані

У єдиному рядку виведіть n цілих чисел — перепорядкований масив такій, що сума будь-яких двох послідовних елементів є непарною, або **-1**, якщо так зробити неможливо.

Обмеження

$$1 \leq n \leq 10^5,$$

$$1 \leq a_i \leq 10^9.$$

2. Дизайн та планована оцінка часу виконання завдань:

Програма №1 VNS Lab 4 Варіант: 9

- Планований час на реалізацію: 20 хв

Програма №2 VNS Lab 5 Варіант: 9

- Планований час на реалізацію: 30 хв.

Програма №3 Algotester Lab 3 Варіант: 2

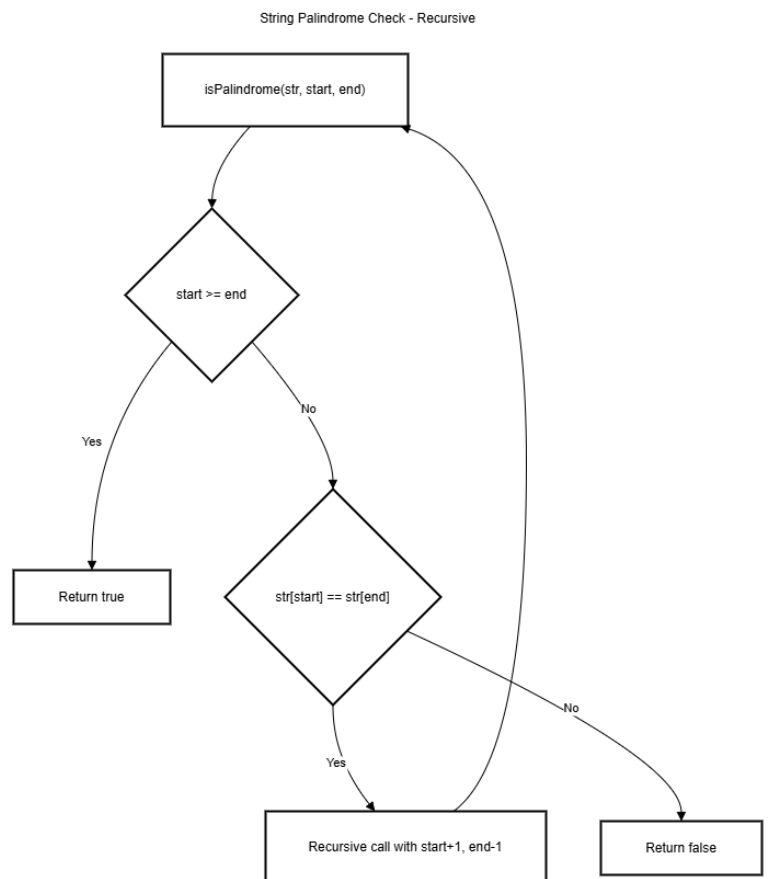
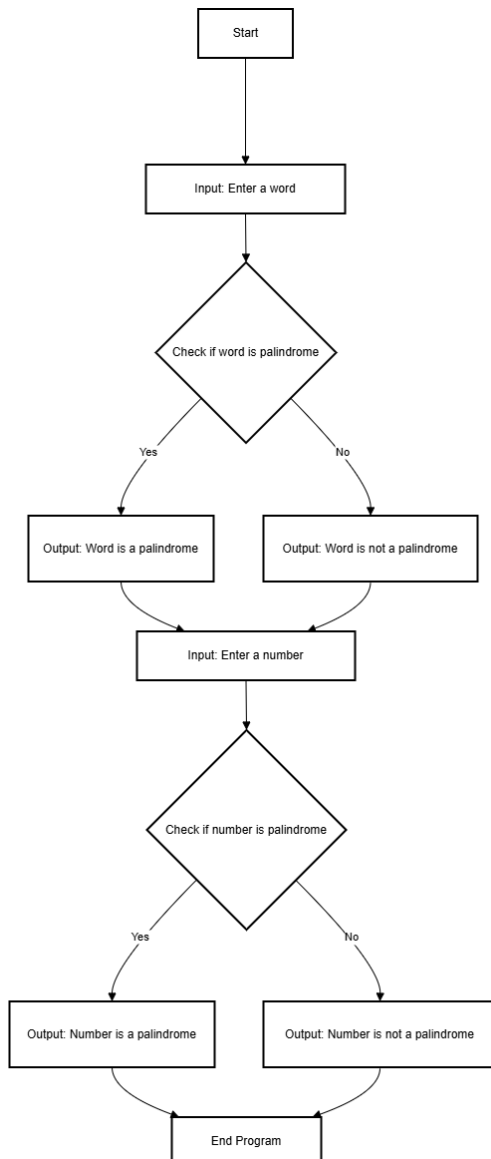
- Планований час на реалізацію: 20 хв.

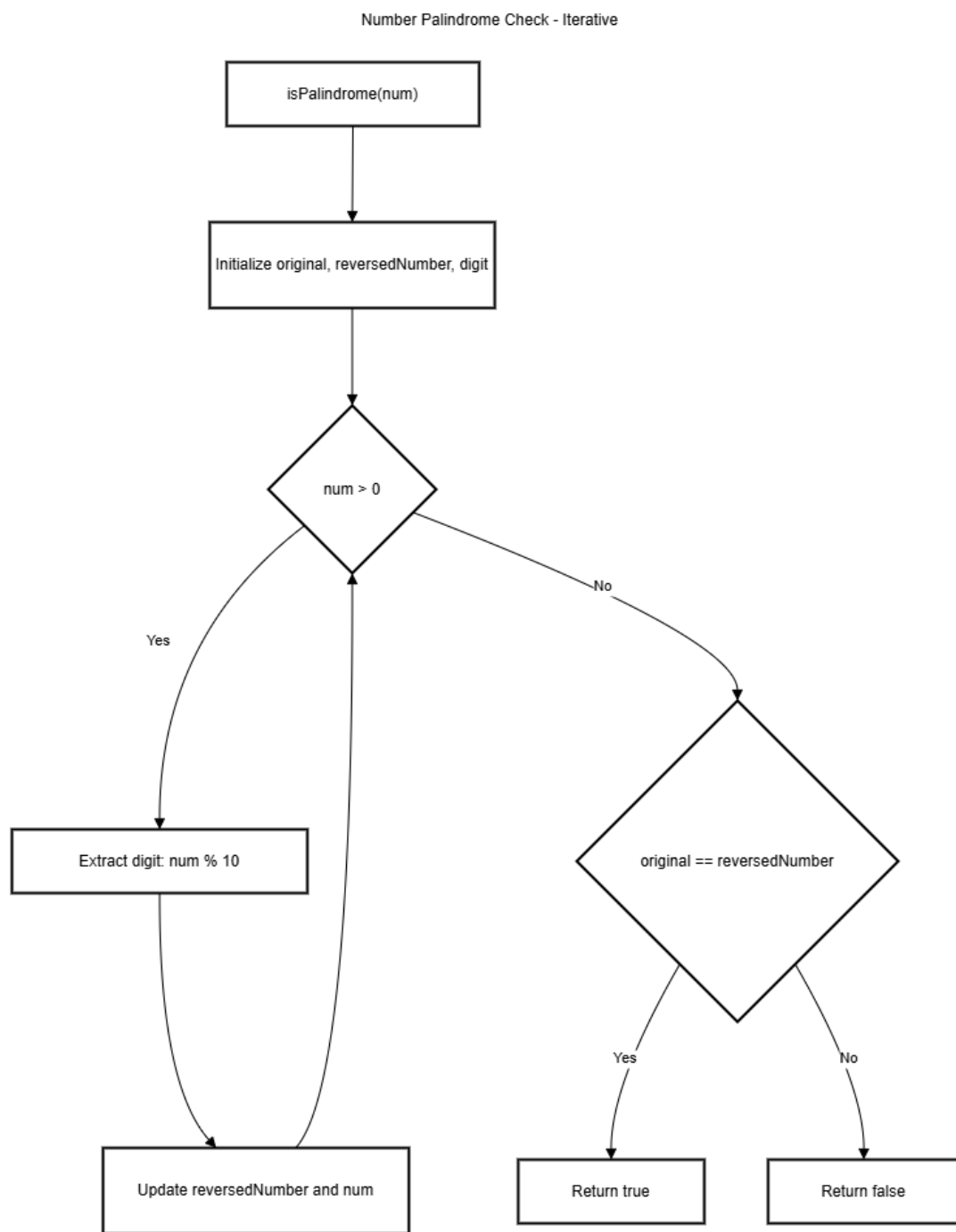
Програма №4 VNS Algotester Lab 2 Варіант: 3

- Планований час на реалізацію: 30 хв.

Програма №5 Class Practice Task

- Блок-схема





- Планований час на реалізацію: 50 хв.

Програма №6 Self Practice Task Непарний масив

- Планований час на реалізацію: 50 хв

4. Код програм з посиланням на зовнішні ресурси:

Завдання №1

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/266/files#diff-92afdec32338c21e40ccb123a692a4d858248aa262b93e76a98f8e4a1ecb7a9c

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

int main()
{
    int elementToDelete, indexOfElementToDelete = -1, k, size;
    int arr[100], newArr[100];

    cout << "Enter the size of array: ";
    cin >> size;
    random_device rd;
    mt19937 gen(rd());
    uniform_int_distribution<int> distr(1, 100);

    for (int i = 0; i < size; ++i)
    {
        arr[i] = distr(gen);
    }

    for (int i = 0; i < size; ++i)
    {
        cout << arr[i] << " ";
    }
    cout << endl;

    cout << "Enter the value you want to delete: ";
    cin >> elementToDelete;
    for (int i = 0; i < size; ++i)
    {
        if (arr[i] == elementToDelete)
            indexOfElementToDelete = i;
    }

    if (indexOfElementToDelete != -1)
    {
        for (int i = indexOfElementToDelete; i < size; ++i)
        {
            arr[i] = arr[i + 1];
        }
    }
}
```



```

        --size;
    }

    else
    {
        cout << "No value to delete!" << endl;
    }

    cout << "How many elements shall the array be shifted to the right: ";
    cin >> k;
    int newIndex;
    for (int i = 0; i < size; ++i)
    {
        newIndex = (i + k) % size;
        if (newIndex < 0)
        {
            newIndex = size + newIndex;
        }
        newArr[newIndex] = arr[i];
    }

    for (int i = 0; i < size; ++i)
    {
        cout << newArr[i] << " ";
    }
    cout << endl;

    return 0;
}

```

Завдання №2

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/266/files#diff-d83e20e4ef30c0f1fd7f0c74bbf46dbb56310b423dcddd76f4715906f89f86d1

```

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

int sumOfBelowNumbers(int matrix[5][5])
{
    int sum = 0;
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < i; j++)
        {
            sum += matrix[i][j];
        }
    }
}

```

```

    }
}

return sum;
}

int main()
{
    int matrices[10][5][5];
    int sums[10];
    random_device rd;
    mt19937 gen(rd());
    uniform_int_distribution<int> distr(-100, 100);

    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            for (int k = 0; k < 5; k++)
            {
                matrices[i][j][k] = distr(gen);
            }
        }
        sums[i] = sumOfBelowNumbers(matrices[i]);
    }

    int max = sums[0];
    for (int i = 1; i < 10; i++)
    {
        if (sums[i] > max)
            max = sums[i];
    }
    cout << max;
    return 0;
}

```

Завдання №3

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/266/files#diff-cf6eded9566285684642f3dd4f355a3148e36137ff589f0d872c47ed0f145509

```

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

```

```

int main()
{
    int n;
    cin >> n;
    vector<int> numbers(n);

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

    int indexLeft = 0, indexRight = n - 1;

    while (true)
    {
        if (indexRight - indexLeft == 1)
        {
            cout << indexLeft + 1 << ' ' << indexRight + 1 << endl;
            cout << "Stopped" << endl;
            break;
        }
        else if (indexRight - indexLeft == 0)
        {
            cout << indexLeft + 1 << ' ' << indexRight + 1 << endl;
            cout << "Collision" << endl;
            break;
        }
        else if (indexLeft > indexRight)
        {
            cout << indexLeft + 1 << ' ' << indexRight + 1 << endl;
            cout << "Miss" << endl;
            break;
        }

        indexLeft += numbers[indexLeft];
        indexRight -= numbers[indexRight];
    }

    return 0;
}

```

Завдання №4

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/266/files#diff-555a142b7741ef9ecfef5ddfd85b93585ee9328e7c980c9f60df31447a5c82bb

```
#include <iostream>
```

```

#include <vector>
using namespace std;

int main()
{
    int n, m;
    cin >> n;
    vector<int> a(n);

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

    cin >> m;
    vector<int> b(m);

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

    int intersection = 0, symmetricDifference = 0;

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            if (a[i] == b[j])
            {
                ++intersection;
            }
        }
    }

    for (int i = 0; i < n; i++)
    {
        bool notFound = true;
        for (int j = 0; j < m; j++)
        {
            if (a[i] == b[j])
            {
                notFound = false;
                b[j] = -1;
            }
        }

        if (notFound)

```

```

        {
            ++symmetricDifference;
        }
    }

    for (int i = 0; i < m; i++)
    {
        bool notFound = true;
        for (int j = 0; j < n; j++)
        {
            if (b[i] == a[j])
            {
                notFound = false;
            }
        }

        if (notFound)
        {
            ++symmetricDifference;
        }
    }

    cout << intersection << endl
         << symmetricDifference;

    return 0;
}

```

Завдання №5

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/266/files#diff-84167e79a2609ef992bc476033248f7366f932435119a75b3ac550405ef0164a

```

#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)
{
    int original = num, reversedNumber = 0, digit;
    while (num > 0)
    {
        digit = num % 10;
        reversedNumber = reversedNumber * 10 + digit;
        num = (num - digit) / 10;
    }

    return original == reversedNumber;
}

int main()
{
    string word;
    cout << "Enter a word: ";
    cin >> word;

    if (isPalindrome(word, 0, word.length() - 1))
    {
        cout << word << " is a palindrome." << endl;
    }
    else
    {
        cout << word << " is not a palindrome." << endl;
    }

    int number;
    cout << "Enter a number: ";
    cin >> number;

    if (isPalindrome(number))
    {
        cout << number << " is a palindrome." << endl;
    }
    else
    {
        cout << number << " is not a palindrome." << endl;
    }

    return 0;
}

```

```
}
```

Завдання №6

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/266/files#diff-89b746c0b22a94d4a6cec53f873c9c8cd4e7c0da325cb2ac7af4fd906a84ce52

```
#include <iostream>
#include <vector>
#include <cmath>

using namespace std;

int main()
{
    int n;
    cin >> n;
    vector<int> oddNumbers, evenNumbers;

    int t;
    for (int i = 0; i < n; i++)
    {
        cin >> t;
        if (t % 2 == 0)
        {
            evenNumbers.push_back(t);
        }
        else
        {
            oddNumbers.push_back(t);
        }
    }

    int sizeOdd = oddNumbers.size(), sizeEven = evenNumbers.size();

    if (sizeOdd == 0 || sizeEven == 0)
    {
        cout << -1;
    }
    else if (abs(sizeOdd - sizeEven) <= 1)
    {
        if (sizeOdd >= sizeEven)
        {
            for (int i = 0; i < sizeEven; i++)
```

```

    {
        cout << oddNumbers[i] << ' ' << evenNumbers[i] << ' ';
    }

    if (sizeOdd > sizeEven)
        cout << oddNumbers[sizeOdd - 1];
}
else
{
    for (int i = 0; i < sizeOdd; i++)
    {
        cout << evenNumbers[i] << ' ' << oddNumbers[i] << ' ';
    }

    if (sizeEven > sizeOdd)
        cout << evenNumbers[sizeEven - 1];
}
}
else
{
    cout << -1;
}

return 0;
}

```

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

Завдання №1

Створив псевдодинамічний масив за допомогою оголошення його розмірності, як 100.
Та утворив псевдозмінні межі.

```

Enter the size of array: 10
92 68 62 43 70 97 15 56 21 46
Enter the value you want to delete: 62
How many elements shall the array be shifted to the right: 3
56 21 46 92 68 43 70 97 15

```

Також додав вивід, коли елемент для видалення не виявлено:


```

Enter the size of array: 10
87 49 58 2 80 52 64 96 97 89
Enter the value you want to delete: 4
No value to delete!
How many elements shall the array be shifted to the right: 8
58 2 80 52 64 96 97 89 87 49

```

Додав можливість зсування вліво за допомогою від'ємних значень

```

Enter the size of array: 10
15 27 31 37 49 38 67 1 27 21
Enter the value you want to delete: 31
How many elements shall the array be shifted to the right: -5
67 1 27 21 15 27 37 49 38

```

Час затрачений на виконання завдання: 35 хв.

Завдання №2

Помістив 10 матриць 5*5 в один масив.

Заповнив їх випадковими значеннями за допомогою псевдовипадкових чисел.

```

PS C:\Users\Lenovo\ai_programming_playground_2024> & 'c:\Users\Lenovo\.vscode\extensions
'--stdin=Microsoft-MIEngine-In-mgq1dqde.vi5' '--stdout=Microsoft-MIEngine-Out-rsu5glhj.ni
f1d1zgze.n2z' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
264
PS C:\Users\Lenovo\ai_programming_playground_2024>

```

Час затрачений на виконання завдання: 40 хв.

Завдання №3

4 години тому	Lab 2v3 - Lab 2v3	C++ 23	Зараховано	0.003	1.215	1861777
---------------	-------------------	--------	------------	-------	-------	---------

Завдання №4

3 години тому	Lab 3v2 - Lab 3v2	C++ 23	Зараховано	0.003	1.180	1861787
---------------	-------------------	--------	------------	-------	-------	---------

Завдання №5

Ввід спершу слова, а потім числа.

Функція для рядків є рекурсивною, що постійно звужує діапазон рядка та перевіряє, чи є символи на кінцях однаковими.

Перевантажена функція для цілочисельних значень використовує спосіб ділення на остачу та цілочисельне ділення для пошуку протилежного числа. Після чого звіряємо, чи є число паліндромом.

```
Enter a word: pylyp
pylyp is a palindrome.
Enter a number: 23544342
23544342 is not a palindrome.
```

Завдання №6

Розбиваємо масив на два підмасиви: один для парних чисел, інший для непарних.

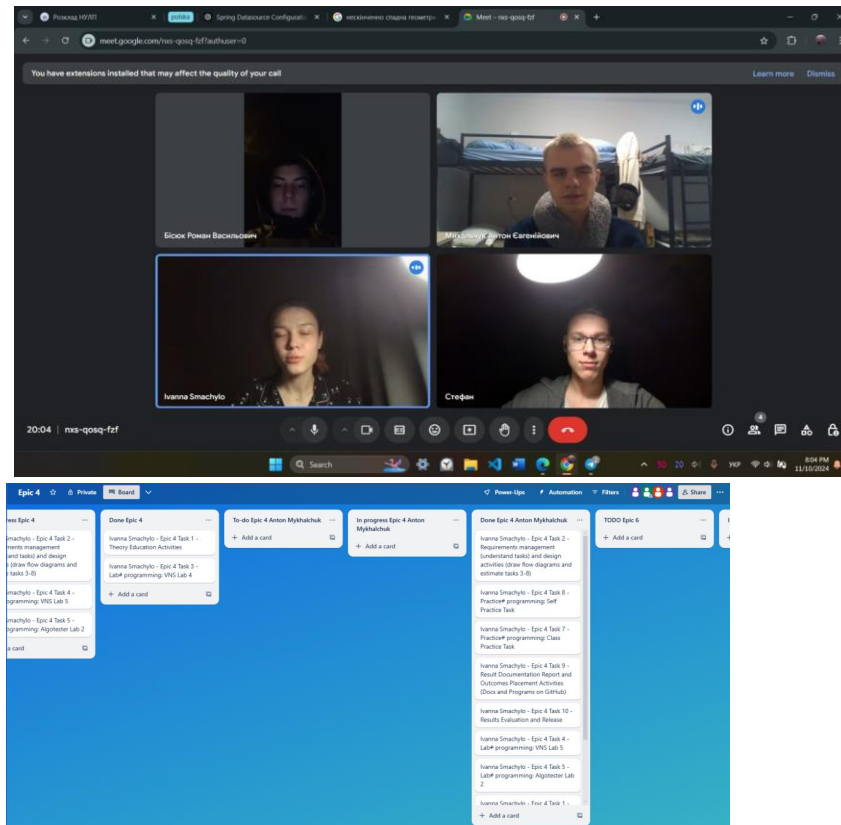
Якщо кількість парних або непарних чисел дорівнює нулю, виводимо -1.

Якщо кількість елементів кожного типу відрізняється не більш ніж на 1, то можемо чергувати їх для досягнення бажаного результату.

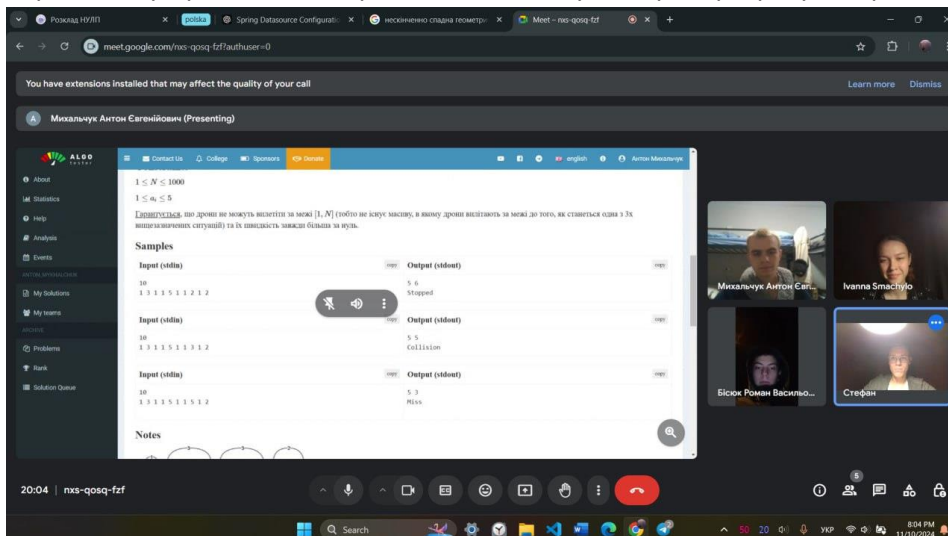
2 години тому	1172 - Непарний масив	C++ 23	Зараховано	0.061	2.008	1861849
---------------	-----------------------	--------	------------	-------	-------	---------

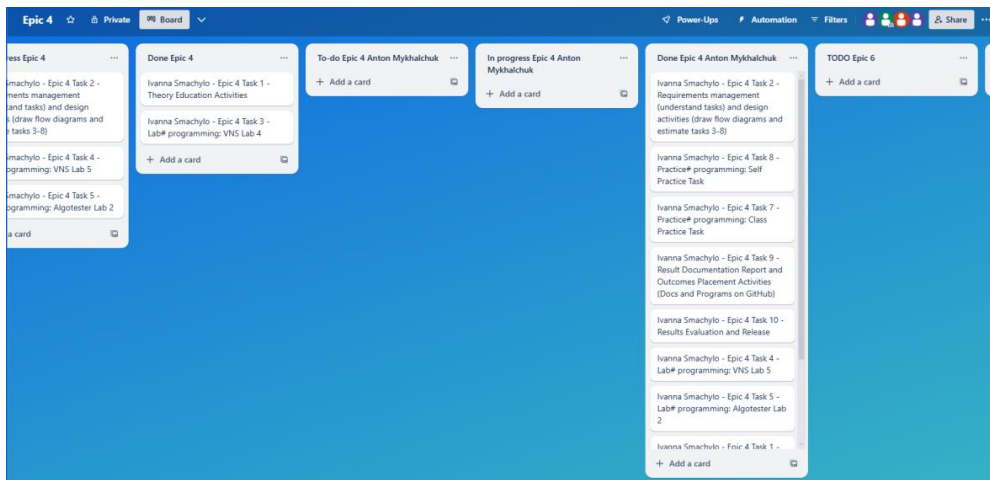
6. Кооперація з командою:

- Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло

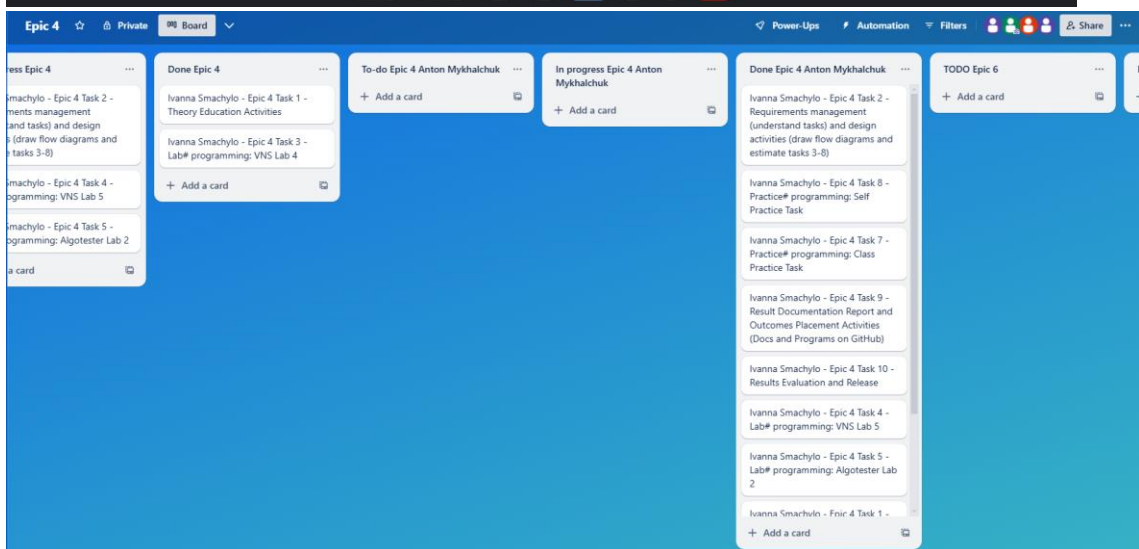
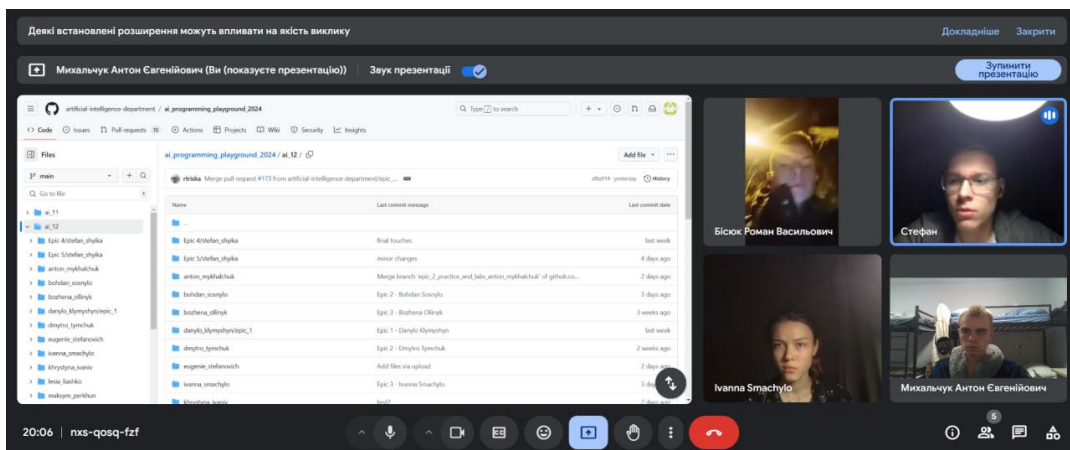


- Скрін з 2-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло





- Скрін з 3-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло (опційно)



Висновки:

У ході роботи було розглянуто та опрацьовано основні концепції програмування в мові C++, пов'язані з циклами та функціями. Особливу увагу приділено вкладеним циклам, умовам завершення їх виконання, а також реалізації різноманітних завдань за допомогою циклічних структур.

Засвоєно принципи роботи з функціями, включаючи базові виклики та перевантаження для підтримки аргументів різного типу. Досліджено використання функцій зі змінною кількістю параметрів, рекурсивні виклики та вбудовані функції для оптимізації коду. Виконані завдання дозволили поглибити розуміння застосування теоретичних знань на практиці, включаючи реалізацію математичних розрахунків через ряди, обчислення з точністю до ϵ , а також створення програм із використанням структурованих та перевантажених функцій для специфічних сценаріїв.

Загалом, робота дала змогу оволодіти важливими аспектами циклів і функцій, що є основою для створення ефективних програм на C++.