

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 ШІ – 12

Лящук Соломія

EPIC 4

“Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.”

Tasks - (Задачі 4 ітерації в Trello або Any Tasks Manager):

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

Sources:

<http://cpp.dp.ua/klasy-pam-yati-u-c-builder/>

<https://www.youtube.com/watch?v=KHK5bMp6vbo>

<https://acode.com.ua/urok-86-vkazivnyky-i-masyvy/>

<https://acode.com.ua/urok-81-bagatovymirni-masyvy/>

<https://acode.com.ua/urok-64-struktury/>

<https://acode.com.ua/urok-84-vkazivnyky/>

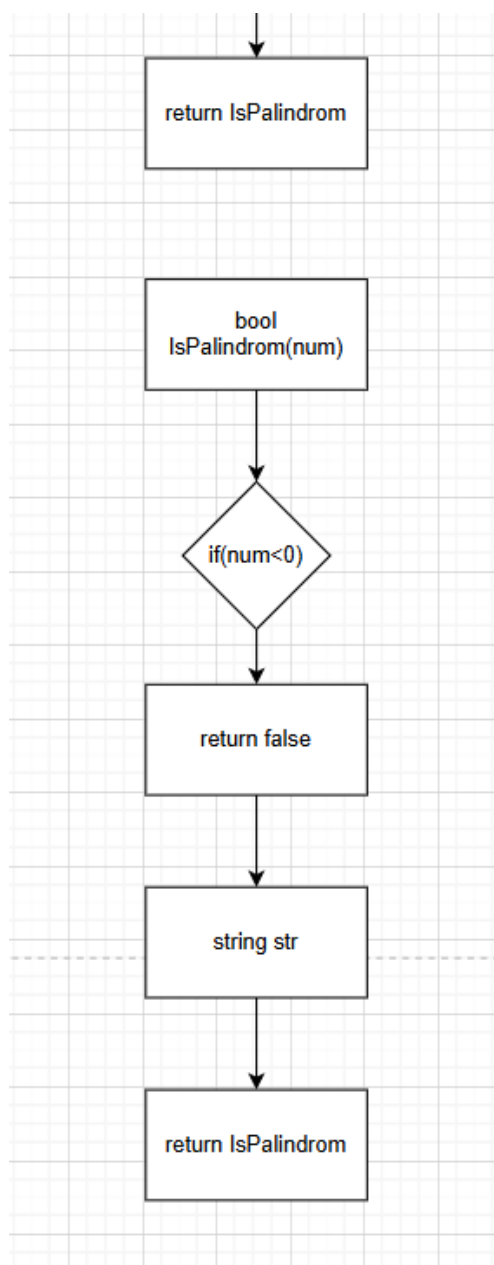
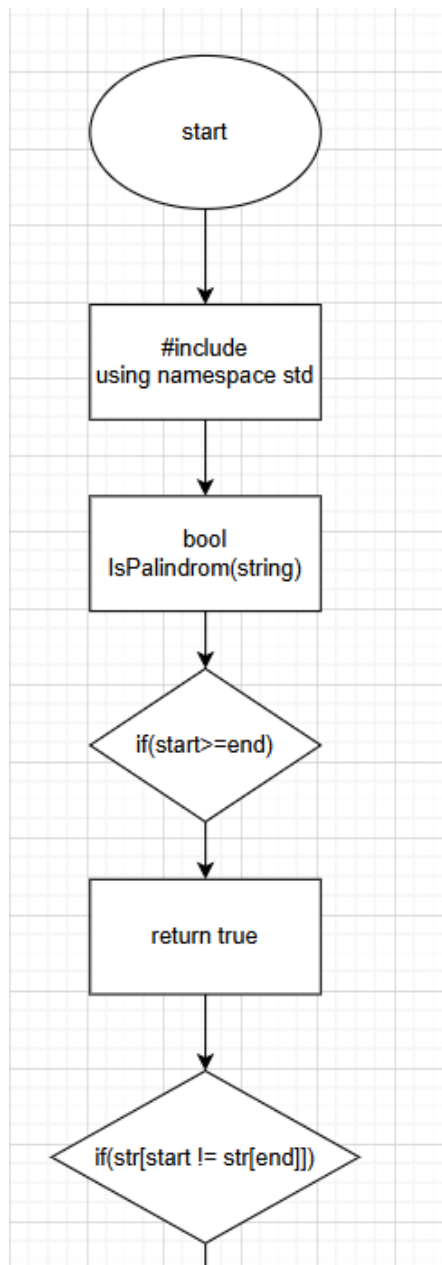
<https://acode.com.ua/urok-92-posylannya/>

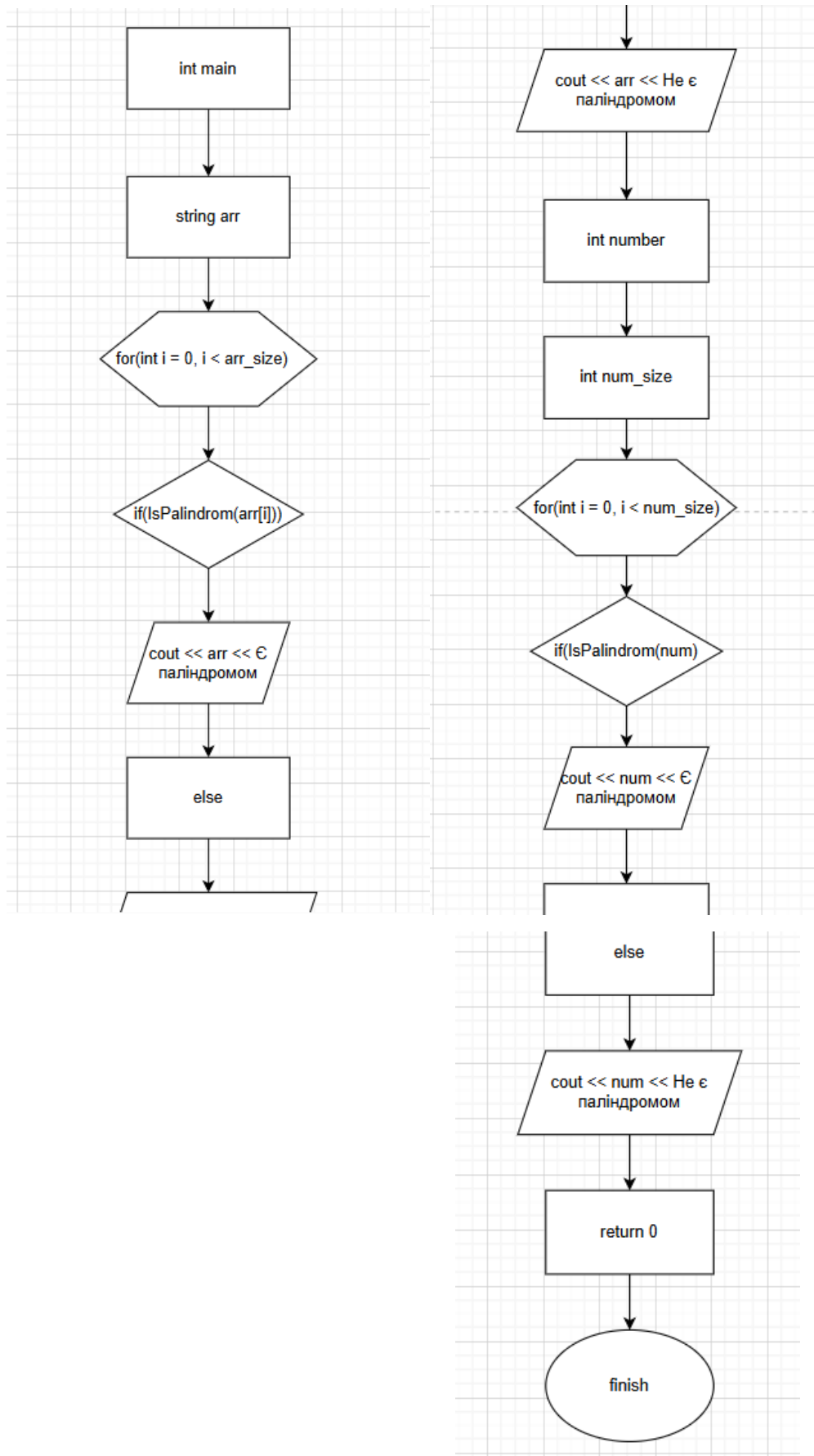
Task 1 - Theory Education Activities

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.

Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-8)





Task 3 - VNS Lab 4

V – 21

- 1) Реалізувати з використанням масиву двонаправлене кільце (перегляд можливий в обидва боки, від останнього елемента можна перейти до першого).
- 2) Роздрукувати отриманий масив, починаючи з K-ого елемента і до K-1 (по кільцю вліво).
- 3) Додати в кільце перший і 3 останніх елементи.
- 4) Роздрукувати отриманий масив, починаючи з K-ого елемента (і до K+1 по кільцю вправо).

```
> cpp > vns_lab_4_task_1_variant_1_solomia_liashchuk.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      const int size = 10;
7
8      int k;
9
10     cout << "Введіть значення для k(від 0 до 9): ";
11     cin >> k;
12
13     int arr[size] = {1, 42, 11, 9, 37, 7, 23, 12, 4, 3};
14
15     cout << "Цикл вліво: ";
16     for (int i = 0; i < size; i++)
17     {
18         int index = (k - i + size) % size;
19         cout << arr[index] << " ";
20     }
21     cout << endl;
22
23     int newSize = size + 4;
24     int newArr[newSize];
25
26     newArr[0] = arr[0];
27     for(int i = 0; i < newSize; i++)
28     {
29         newArr[i + 1] = arr[i];
30     }
31
32     newArr[size + 1] = arr[size - 3];
33     newArr[size + 2] = arr[size - 2];
34     newArr[size + 3] = arr[size - 1];
35
36     int n;
```

```

37     cout << "Введіть значення для k(від 0 до " << newSize - 1 << "): ";
38     cin >> n;
39
40     cout << "Цикл вправо: ";
41     for (int i = 0; i < newSize; i++)
42     {
43         int index = (n + i) % newSize;
44         cout << newArr[index] << " ";
45     }
46     cout << endl;
47
48     return 0;
49 }

```

```

rpreter=mi'
• Введіть значення для k(від 0 до 9): 7
  Цикл вліво: 12 23 7 37 9 11 42 1 3 4
  Введіть значення для k(від 0 до 13): 4
  Цикл вправо: 9 37 7 23 12 4 3 12 4 3 1 1 42 11
○ PS C:\Users\olesi>

```

В цьому завданні я використала масиви для збереження значень і цикли щоб виконати умову задачі

Task 4 - VNS Lab 5

V – 21

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

Знайти максимальний з повторюваних елементів двовимірного масиву

```

1  #include <iostream>
2  using namespace std;
3
4  int countOccurrences(int arr[5][3], int rows, int cols, int num)
5  {
6      int count = 0;
7      for (int i = 0; i < rows; i++)
8      {
9          for (int j = 0; j < cols; j++)
10         {
11             if(arr[i][j] == num)
12             {
13                 count++;
14             }
15         }
16     }
17     return count;
18 }
19
20 int FindMax(int arr[5][3], int rows, int cols)
21 {
22     int maxRepeated = -1;
23     bool foundMax = false;
24
25     for (int i = 0; i < rows; i++)
26     {
27         for (int j = 0; j < cols; j++)
28         {
29             int currentElement = arr[i][j];
30
31             if (countOccurrences(arr, rows, cols, currentElement) > 1)
32             {

```

```

31             if (countOccurrences(arr, rows, cols, currentElement) > 1)
32             {
33                 if (!foundMax || currentElement > maxRepeated)
34                 {
35                     maxRepeated = currentElement;
36                     foundMax = true;
37                 }
38             }
39         }
40     }
41
42     if(!foundMax)
43     {
44         return -1;
45     }
46     return maxRepeated;
47 }
48
49 int main()
50 {
51     int rows = 5;
52     int cols = 3;
53
54     int arr[5][3] =
55     {
56         {1, 2, 7},
57         {5, 11, 6},
58         {8, 27, 12},
59         {9, 45, 6},

```



```

54     int arr[5][3] =
55     {
56         {1, 2, 7},
57         {5, 11, 6},
58         {8, 27, 12},
59         {9, 45, 6},
60         {33, 27, 8}
61     };
62
63     int result = FindMax(arr, rows, cols);
64
65     if (result != -1)
66     {
67         cout << "Максимальний повторюваний елемент: " << result << endl;
68     }
69     else
70     {
71         cout << "Повторюваних елементів не знайдено." << endl;
72     }
73
74     return 0;
75 }
76

```

Максимальний повторюваний елемент: 27

Використовувала двовимірний масив та функції для розв'язання задачі.

Task 5 - Algotester Lab 2

V - 3

```

1  #include <iostream>
2
3  int main()
4  {
5      int N;
6      std::cin >> N;
7      int arr[N];
8
9      for(int i = 0; i < N; i++)
10         std::cin >> arr[i];
11     int left_pos = 0, right_pos = N - 1;
12
13     for(int i = 0; i < N; i++)
14     {
15         if(left_pos == right_pos)
16         {
17             std::cout << left_pos + 1 << " " << right_pos + 1 << std::endl;
18             std::cout << "Collision";
19             break;
20         }
21         if(left_pos > right_pos)
22         {
23             std::cout << left_pos + 1 << " " << right_pos + 1 << std::endl;
24             std::cout << "Miss";
25             break;
26         }
27         if((right_pos - left_pos) == 1)
28         {
29             std::cout << left_pos + 1 << " " << right_pos + 1 << std::endl;
30             std::cout << "Stopped";
31             break;
32         }
33
34         left_pos += arr[left_pos];
35         right_pos -= arr[right_pos];
36     }

```

```

33
34         left_pos += arr[left_pos];
35         right_pos -= arr[right_pos];
36     }
37
38     return 0;
39 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS



```

• PS C:\Users\olesi> & 'c:\Users\olesi\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAd
-stdout=Microsoft-MIEngine-Out-50bez1lk.e3f' '--stderr=Microsoft-MIEngine-Error-jlsyx1lk.tqe' '--pid=
rpreter=mi'
6
4
3
1
2
3
1
5 5
Collision

```

Task 6 - Algotester Lab 3

V – 3

C: > cpp >  algotester_lab_3_task_solomia_liashchuk.cpp >  main()

```
1  #include <iostream>
2  #include <string>
3
4  int main()
5  {
6      std::string str;
7      std::cin >> str;
8
9      int counter = 1;
10     int n = str.length();
11
12     for (int i = 0; i < n; i++)
13     {
14         if (i + 1 < n && str[i] == str[i + 1])
15         {
16             counter++;
17         } else
18         {
19             std::cout << str[i];
20             if (counter > 1)
21             {
22                 std::cout << counter;
23             }
24             counter = 1;
25         }
26     }
27
28     return 0;
29 }
30
```

options compilation execution

SSSKKHNNRQQ
S3K2HN3RQ2

Task 7 - Class Practice Task

Задача

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

```
C: > cpp > practice_work_team_tasks_solomia_lashchuk.cpp > main()
1  #include <iostream>
2  #include <string>
3  #include <sstream>
4  #include <algorithm>
5
6  using namespace std;
7
8  bool IsPalindrome(const string& str, int start, int end)
9  {
10     if (start >= end)
11         return true;
12
13     if (str[start] != str[end])
14         return false;
15
16     return IsPalindrome(str, start + 1, end - 1);
17 }
18
19 bool IsPalindrom(int num)
20 {
21     if (num < 0)
22         return false;
23
24     string str = to_string(num);
25
26     return IsPalindrome(str, 0, str.size() - 1);
27 }
28
29 int main()
30 {
31     string arr[] = {"programming", "1254521", "winter", "coca-cola", "radar", "Lviv"};
32
33     int arr_size = sizeof(arr) / sizeof(arr[0]);
34
35     for (int i = 0; i < arr_size; ++i)
36     {
37         if (IsPalindrom(arr[i]) || arr[i].size() < 2)
```

```

36     {
37         if (IsPalindrome(arr[i], 0, arr[i].size() - 1))
38         {
39             cout << "\"" << arr[i] << "\" є паліндромом." << endl;
40         } else
41         {
42             cout << "\"" << arr[i] << "\" не є паліндромом." << endl;
43         }
44     }
45
46     int numbers[] = {121, 12321, 12345, 111, -121};
47
48     int num_size = sizeof(numbers) / sizeof(numbers[0]);
49
50     for (int i = 0; i < num_size; ++i)
51     {
52         if (IsPalindrom(numbers[i]))
53         {
54             cout << numbers[i] << " є паліндромом." << endl;
55         } else
56         {
57             cout << numbers[i] << " не є паліндромом." << endl;
58         }
59     }
60
61     return 0;
62 }
63

```

```

"programming" не є паліндромом.
"1254521" є паліндромом.
"winter" не є паліндромом.
"coca-cola" не є паліндромом.
"radar" є паліндромом.
"Lviv" не є паліндромом.
121 є паліндромом.
12321 є паліндромом.
12345 не є паліндромом.
111 є паліндромом.
-121 не є паліндромом.

```

Task 8 - Self Practice Task

Odd Array

```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main()
6  {
7      int n;
8      cin >> n;
9      vector<int> arr(n);
10     vector<int> oddNums, evenNums;
11
12     for(int i = 0; i < n; i++)
13     {
14         cin >> arr[i];
15         if(arr[i] % 2 == 0)
16         {
17             evenNums.push_back(arr[i]);
18         }
19         else
20         {
21             oddNums.push_back(arr[i]);
22         }
23     }
24
25     if(abs((int)evenNums.size() - (int)oddNums.size()) > 1)
26     {
27         cout << -1 << "\n";
28         return 0;
29     }
30     vector<int> f;
31
32     if(evenNums.size() >= oddNums.size())
33     {
34         for(int i = 0; i < evenNums.size(); i++)
35         {

```

```

31
32     if(evenNums.size() >= oddNums.size())
33     {
34         for(int i = 0; i < evenNums.size(); i++)
35         {
36             f.push_back(evenNums[i]);
37             if(i < oddNums.size())
38                 f.push_back(oddNums[i]);
39         }
40     }
41
42     else
43     {
44         for(int i = 0; i < oddNums.size(); i++)
45         {
46             f.push_back(oddNums[i]);
47             if(i < evenNums.size())
48                 f.push_back(evenNums[i]);
49         }
50     }
51
52
53     for(int num : f)
54     {
55         cout << num << " ";
56         cout << "\n";
57     }
58
59     return 0;
60 }

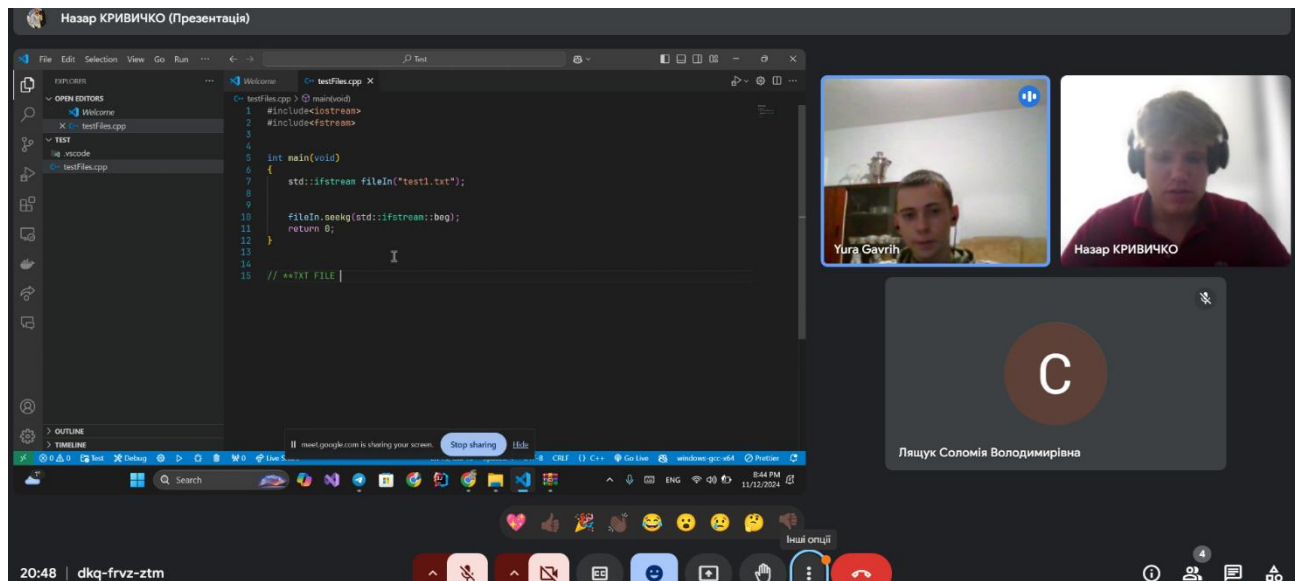
```

```

PS C:\Users\olesi> g++ c:\Users\olesi\vscode\src\main.cpp -stdout=Microsoft-MIEngine-Out-lhojaevq.tj -preprocessor=mi'
4
7
8
4
9
8
7
4
9
PS C:\Users\olesi>

```

Робота в команді:



Висновок: у цій лабораторній роботі я ознайомила з основами роботи з масивами, вказівниками, посиланнями, динамічними масивами та структурами даних у C++. Навчилася використовувати одновимірні й двовимірні масиви для зберігання та обробки даних, працювати з вказівниками для управління пам'яттю та створювати динамічні масиви невідомого наперед розміру. Робота зі структурами та вкладеними структурами допомогла зрозуміти, як організовувати складніші дані, а застосування алгоритмів обробки — як ефективно працювати з ними.

