Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 4**

***з дисципліни:*** «Мови та парадигми програмування»

***з розділу***: «Прості структури даних. Одновимірні масиви. Двовимірні масиви. Алгоритми обробки»

***Виконав:***

студент групи ШІ-11

Винницький Андрій Олегович

# **Тема роботи:**

# Прості структури даних. Одновимірні масиви. Двовимірні масиви. Алгоритми обробки

# **Мета роботи:**

Навчитися працювати з простими структурами даних, такими як одновимірні та двовимірні масиви. У програмуванні розвивати навички роботи з елементами масивів, розробляти алгоритми для обробки даних та вирішення конкретних завдань. Застосовувати отримані знання для практичних завдань, таких як сортування, пошук елементів у масивах, а також оптимізація алгоритмів.. Всі зроблені задачі і звіт запушити на GitHub і зробити pull request.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Прості структури даних
* Тема №2: Одновимірні масиви
* Тема №3: Двовимірні масиви
* Тема №4: Вектори

1. Індивідуальний план опрацювання теорії:

* Тема №1: Прості структури даних
  + Статус: Ознайомлений
  + Початок опрацювання теми: 12.11.2023
  + Звершення опрацювання теми: 18.11.2023
* Тема №2: Одновимірні масиви.
  + Джерела Інформації:
    - <https://www.guru99.com/cpp-dynamic-array.html>
    - <https://www.w3schools.com/cpp/cpp_arrays.asp>
    - <https://cplusplus.com/doc/tutorial/arrays/>
  + Що опрацьовано:
    - Ознайомився з динамічним одновимірним масивом і звичайним масивом і застосував на практиці
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.11.2023
  + Звершення опрацювання теми: 22.11.2023
* Тема №3: Двовимірні масиви
  + Джерела Інформації:
    - <https://www.programiz.com/cpp-programming/multidimensional-arrays>
    - <https://www.w3schools.com/cpp/cpp_arrays_multi.asp>
  + Що опрацьовано:
    - Ознайомився з двовимірним масивом
  + Статус: Ознайомлений
  + Початок опрацювання теми: 23.11.2023
  + Звершення опрацювання теми: 24.11.2023
* Тема №4: Вектори
  + Джерела Інформації:
    - <https://cplusplus.com/reference/vector/vector/>
  + Що опрацьовано:
    - Ознайомився з векторами і застосував їх на практиці
  + Статус: Ознайомлений
  + Початок опрацювання теми: 23.11.2023
  + Звершення опрацювання теми: 24.11.2023

# **Виконання роботи:**

## **1. Опрацювання завдання та вимог до програм та середовища:**

Завдання №1 VNS Lab 4

* Варіант завдання 9
* В завданні необхідно:

1) Сформувати одновимірний масив цілих чисел, використовуючи генератор

випадкових чисел.

2) Роздрукувати отриманий масив.

3) Знищити перший елемент із заданим значенням.

4) Зсунути масив циклічно на К елементів вправо.

5) Роздрукувати отриманий масив.

Завдання №2 VNS Lab 5

* Варіант завдання 9
* В завданні необхідно:

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 завдання 3
* Потрібно реалізувати описану задачу:

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 перелетить у клітинку з індексом a1, тобто його наступна позиція рахується як поточна позиція + число у поточній .Правий робить аналогічно в протилежну сторону. Вони це роблять до моменту, коли трапиться одна з зазначених подій:

Якшо 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 –

Завдання №4 Algotester Lab 3

* Варіант завдання 2
* Потрібно реалізувати описану задачу:

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.

Завдання №5 Class Practice Work

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

Завдання №5 Self Practice Work

* Реалізувати counting sort

## **2. Дизайн та планована оцінка часу виконання завдань:**

Програма №1 VNS Lab 4

* Планований час на реалізацію 30 хв

Програма №2 VNS Lab 5

* Планований час на реалізацію 30 хв

Програма №3 Algotester Lab 2

* Блок-схема

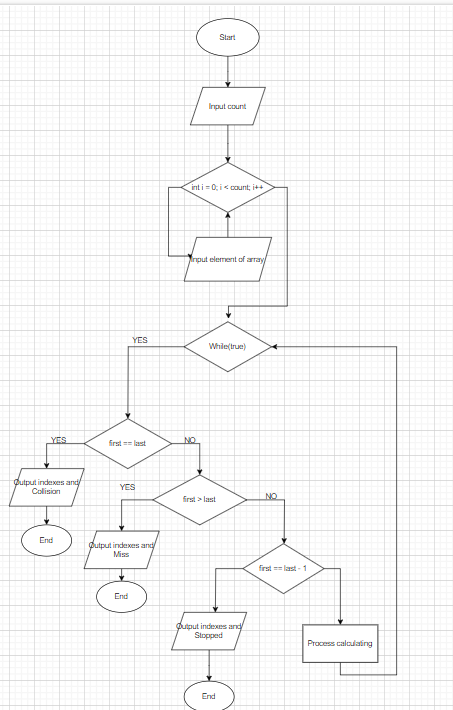


Рисунок : Блок схема до задачі на алготестері

* Планований час на реалізацію 10 хв

Програма №4 Algotester Lab 3

* Планований час на реалізацію 15 хв

Програма №5 Class Practice Work

* Планований час на реалізацію 20 хв

Програма №5 Self Practice Work

Планований час на реалізацію 20 хв

## **3. Код програм з посиланням на зовнішні ресурси:**

Завдання №1 VNS Lab 4

#include <iostream>  
#include<ctime>  
using namespace std;  
  
int\* creation\_of\_array(int size)  
{  
 int\* temp\_arr = new int[size];  
 for(int i = 0; i < size; i++)  
 {  
 temp\_arr[i] = rand() % 100;  
 }  
 return temp\_arr;  
}  
  
void print\_array(int\* arr, int size)  
{  
 for(int i = 0; i < size; i++)  
 {  
 cout<< arr[i]<< " ";  
 }  
}  
  
void deletion\_element(int\*& arr, int& size , int index)  
{  
 int\* temp\_arr = new int[size - 1];  
 for(int i = 0; i < size; i++)  
 {  
 if (i != index)  
 {  
 temp\_arr[i - 1] = arr[i];  
 }  
  
 }  
 delete [] arr;  
 arr = temp\_arr;  
 --size;  
  
}  
  
void cycle\_rotating(int\*& arr ,int& size, int parametr\_of\_rotating)  
{  
 int\* temp\_arr = new int[size];  
  
 for(int i = size - parametr\_of\_rotating , j = 0; i < size && j < parametr\_of\_rotating;i++,j++)  
 {  
 temp\_arr[j] = arr[i];  
 }  
  
 for(int k = size - parametr\_of\_rotating - 1, h = 0 ; k < size && h < size - parametr\_of\_rotating;k++, h++)  
 {  
 temp\_arr[k] = arr[h];  
 }  
 delete [] arr;  
 arr = temp\_arr;  
  
  
}  
  
int main() {  
 srand(time(0));  
 cout << "Enter size of array: ";  
 int size\_arr;  
 cin>> size\_arr;  
 int\* array = creation\_of\_array(size\_arr);  
 cout << "Generated array: ";  
 print\_array(array, size\_arr);  
 deletion\_element(array, size\_arr, 0);  
 cycle\_rotating(array, size\_arr, 3);  
 cout <<"\nProcessed array: ";  
 print\_array(array, size\_arr);  
 delete []array;  
 return 0;  
}

Час витрачений : 30 хв.

Завдання №2 VNS Lab 5

#include <iostream>  
#include <vector>  
#include <ctime>  
using namespace std;  
  
int max(vector<int> matrix)  
{  
 int max = 0;  
 for(int i = 0; i < matrix.size(); i++)  
 {  
 if(max < matrix[i])  
 {  
 max = matrix[i];  
 }  
 }  
 return max;  
}  
  
  
vector<vector<int>> generate\_matrix(int size)  
{  
 vector<vector<int>> matrix;  
 for(int i = 0; i < size; i++)  
 {  
 matrix.emplace\_back(size);  
 for(int j = 0; j < size; j++)  
 {  
  
 matrix[i][j] = rand() % 100;  
 }  
 }  
 return matrix;  
}  
void print\_matrix(vector<vector<int>> matrix)  
{  
 for(int i = 0; i < matrix.size(); i++)  
 {  
  
 for(int j = 0; j < matrix[i].size(); j++)  
 {  
 cout<< matrix[i][j]<<" ";  
 }  
 cout << endl;  
 }  
}  
int sum(vector<vector<int>> matrix)  
{  
 int sum\_n = 0;  
 for(int i = 0;i < matrix.size();i++)  
 {  
 for(int j = 0;j < i;j++) {  
 sum\_n += matrix[i][j];  
 }  
  
 }  
 return sum\_n;  
}  
  
  
  
int main()  
{  
  
 srand(time(0));  
 vector<int> sums;  
 int number;  
 int size;  
 cout<<"Enter number of matrixes: ";  
 cin>> number;  
 cout <<"Enter size of matrixes: ";  
 cin>> size;  
 for(int i = 0;i < number;i++)  
 {  
 vector<vector<int>> matrix = generate\_matrix(size);  
 cout<<"Number "<< i + 1<< " matrix "<<endl;  
 print\_matrix(matrix);  
 sums.push\_back(sum(matrix));  
 cout<<"Sum under the diagonal:" <<sum(matrix)<<endl;  
 }  
 cout<<"Max sum: "<< max(sums);  
 return 0;  
}

Час витрачений : 20 хв.

Завдання №3 Algotester Lab 2

#include <iostream>  
#include<vector>  
using namespace std;  
  
int main()  
{  
 vector<int> arr;  
 long int count;  
 cin>> count;  
 for(int i = 0; i < count; i++)  
 {  
 int value;  
 cin>> value;  
 arr.push\_back(value);  
 }  
 int first = 0;  
 int last = count - 1 ;  
 while (true)  
 {  
 if(first == last)  
 {  
 cout<< first + 1<<" "<< last + 1<<endl<<"Collision";  
 break;  
 }  
 else if(first > last)  
 {  
 cout<< first + 1<<" "<< last + 1<<endl<<"Miss";  
 break;  
 }  
 else if(first == last - 1)  
 {  
 cout<< first + 1<<" "<< last + 1<<endl<<"Stopped";  
 break;  
 }  
 else  
 {  
 first += arr[first];  
 last -= arr[last];  
 }  
  
 }  
 return 0;  
}

Час витрачений : 10 хв.

Завдання №4 Algotester Lab 3

#include <iostream>  
#include<vector>  
using namespace std;  
  
int main() {  
 vector<int> arr1;  
 vector<int> arr2;  
 long int count;  
 cin >> count;  
 for (int i = 0; i < count; i++) {  
 int value;  
 cin >> value;  
 arr1.push\_back(value);  
 }  
 int count1 = 0;  
 cin >> count1;  
 for (int j = 0; j < count1; j++) {  
 int value;  
 cin >> value;  
 arr2.push\_back(value);  
 }  
 int nonunique = 0;  
 for (int k = 0; k < count; k++)  
 {  
 for (int l = 0; l < count1; l++)  
 {  
 if(arr1[k] == arr2[l])  
 {  
 nonunique+=1;  
 }  
 }  
 }  
 cout <<nonunique<<endl<< count+count1-nonunique;  
 return 0;  
}

Час витрачений : 15 хв.

Завдання №4 Class Practice Work

# #include <iostream>  
using namespace std;  
  
bool is\_palindrome(string str, int first , int last )  
{  
 for (int k = 0; k <str.length(); k++)  
 {  
 str[k] = tolower(str[k]);  
 }  
 if(last > 0 && first < str.length())  
 {  
 if (str[first] == str[last]) {  
 return is\_palindrome(str, ++first, --last);  
 }  
 else  
 {  
 return false;  
 }  
  
 }  
 return true;  
  
  
}  
  
bool is\_palindrome(int num)  
{  
  
 return is\_palindrome(to\_string(num), 0 , to\_string(num).length() - 1 );  
}  
  
int main()  
{  
 string example = "radar";  
 int example1 = 12321;  
 cout<< is\_palindrome(example , 0, example.length() - 1)<<endl;  
 cout<<(is\_palindrome(example1));  
 return 0;  
  
}

Час витрачений : 20 хв.

Завдання №5 Self Practice Work.

#include <iostream>  
#include <vector>  
#include <ctime>  
using namespace std;  
  
void print\_array(vector<int> num)  
{  
 for(int i = 0; i < num.size(); i++)  
 {  
 cout << num[i]<<" ";  
 }  
}  
int max(vector<int> nums)  
{  
 int max\_n = 0;  
 for(int i = 0; i < nums.size(); i++)  
 {  
 if(max\_n < nums[i])  
 {  
 max\_n = nums[i];  
 }  
  
 }  
 return max\_n;  
  
}  
  
  
vector<int> counting\_sort(vector<int> nums)  
{  
 vector<int> count(max(nums) + 1, 0);  
 for(int i :nums)  
 {  
 count[i]++;  
 }  
 for (int j = 1; j < count.size(); j++)  
 {  
 count[j] += count[j - 1];  
 }  
 vector<int> result(nums.size(), 0);  
 for(int k :nums)  
 {  
 result[count[k] - 1] = k;  
 count[k]--;  
 }  
  
 return result;  
}  
  
int main()  
{  
 srand(time(0));  
 vector<int> nums;  
 for(int i = 0; i < 10;i++)  
 {  
 nums.push\_back(rand() % 100);  
 }  
 cout<<"Generated array: ";  
 print\_array(nums);  
 nums = counting\_sort(nums);  
 cout<<"\nSorted array: ";  
 print\_array(nums);  
 return 0;  
}

Час витрачений 30хв.

## **4. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 VNS Lab 4

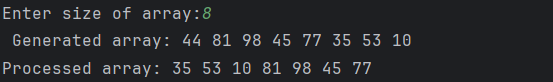


Рисунок 2: Код генерує масив, видаляє перший елемент, і циклічно зсуває масив вправо

Завдання №2 VNS Lab 5

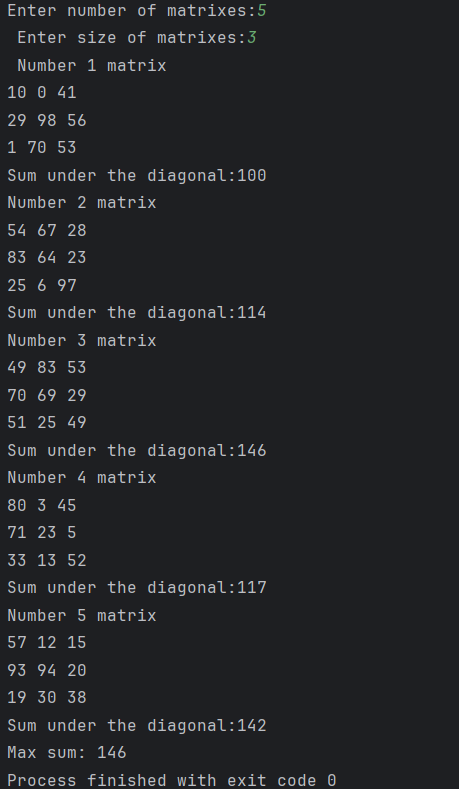
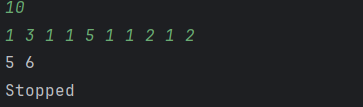


Рисунок 3:Програма генерує задану кількість матриць з заданим розміром ,шукає суму чисел під нижньою діагоналлю і виводить максимальну суму .

Завдання №3 Algotester Lab 2



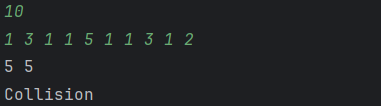


Рисунок 4: Виконує задачу Algotester 2

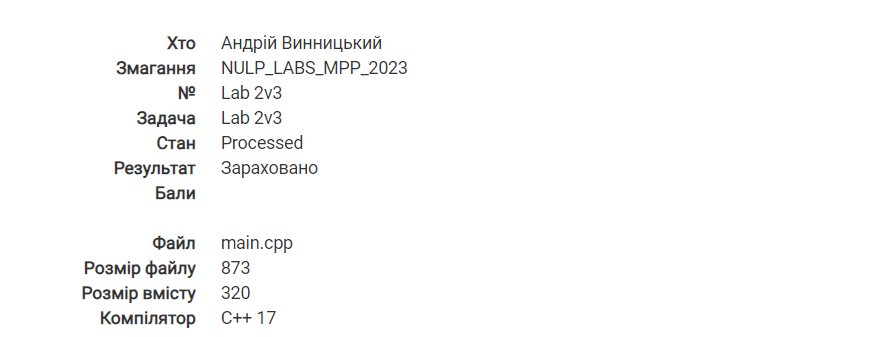


Рисунок 5: Результат коду на алготестері

Завдання №4 Algotester Lab 3



Рисунок 6: Виконує задачу Algotester 3

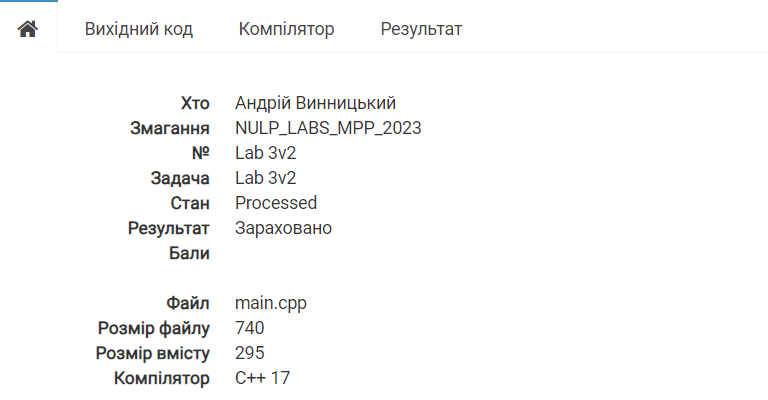


Рисунок 7: Результат коду на алготестері

Завдання №4 Class Practice Work

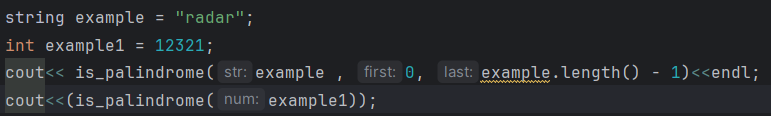




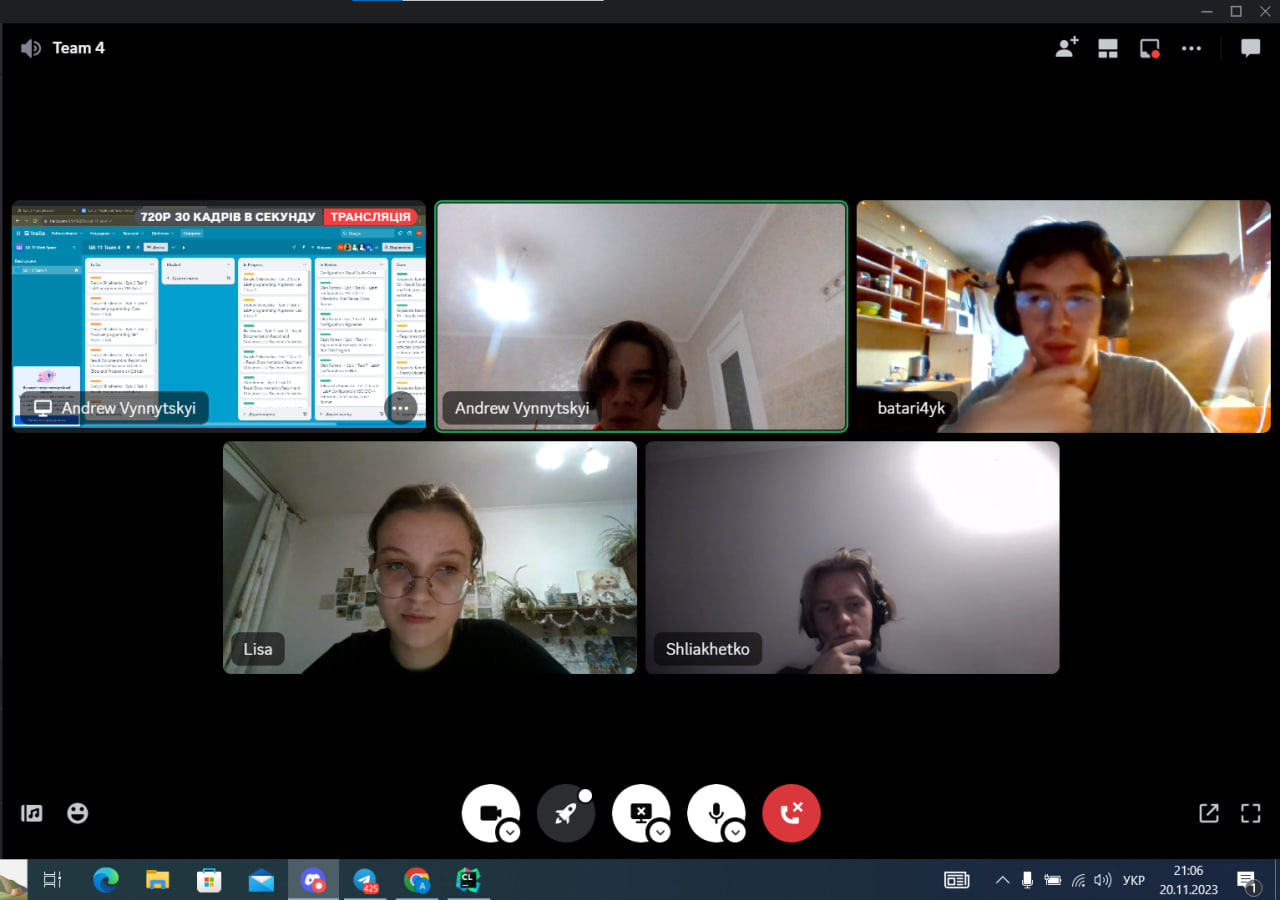
Рисунок 8:Програма перевіряє чи слово або цифра є паліндромом

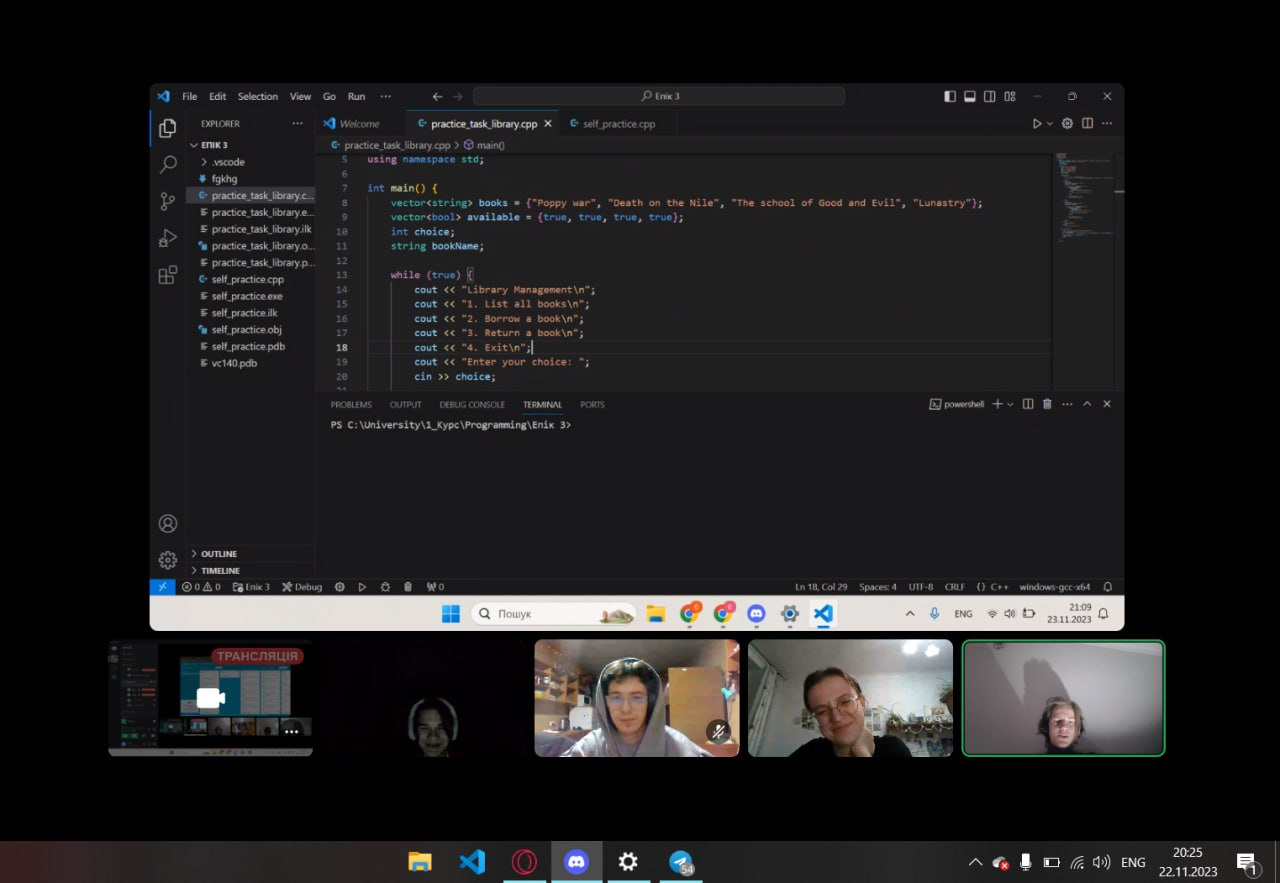
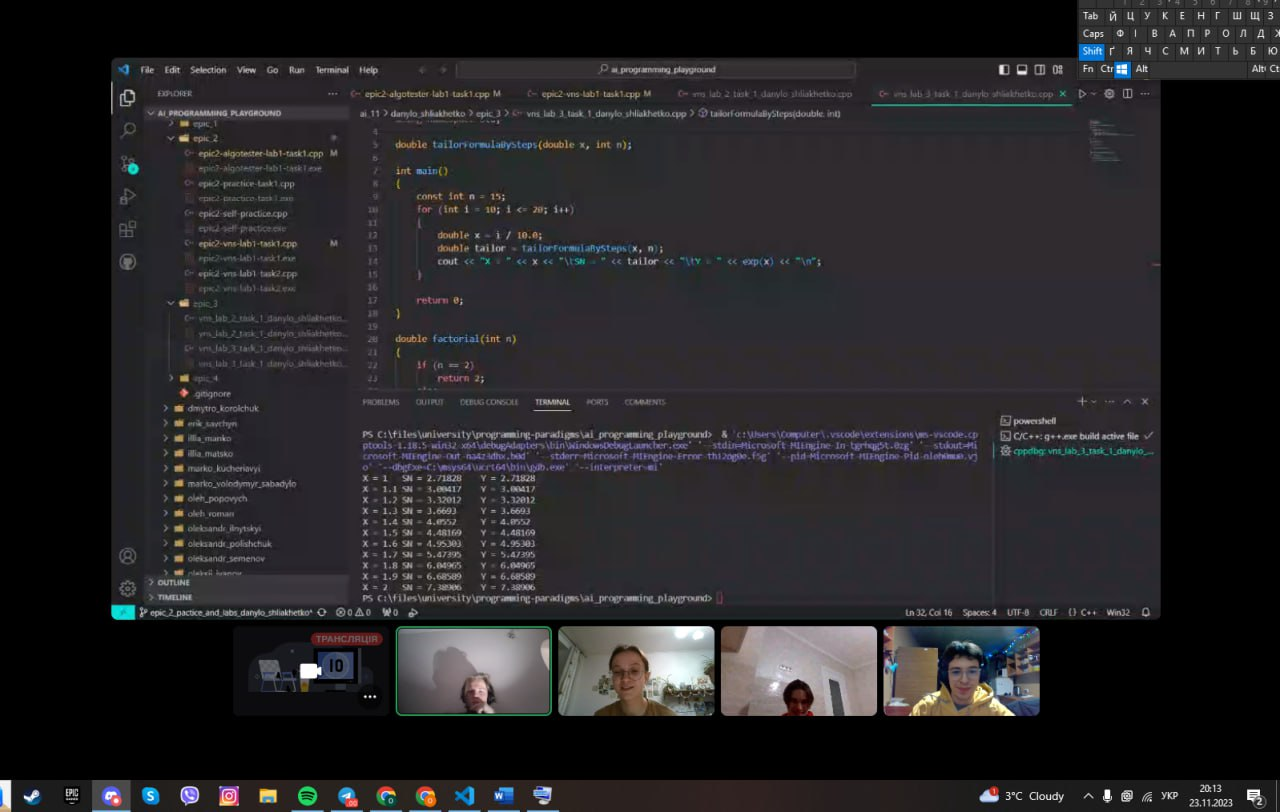
Завдання №5 Self Practice Work.

Рисунок 9: Програма сортує масив алгоритмом counting sort

**6. Командна робота (мітінги, обговорення по написанню програми):**





# **Висновки:**

За час виконання завдання я навчився працювати з динамічними і статичними одновимірними та двовимірними масивами, векторами, ознайомився з їхніми властивостями та особливостями. Я написав завдання з VNS і практичні завдання, в яких використав динамічні масиви, вектори. Під час роботи я видаляв елементи за допомогою динамічного масиву, генерував матриці за допомогою векторів і написав алгоритми сортування.

Всі зроблені задачі і звіт я запушив на GitHub і зробив pull request.