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:*** «Мови та парадигми програмування»

до:

ВНС Лабораторної Роботи №4

ВНС Лабораторної Роботи №5

Алготестер Лабораторної Роботи №2

Алготестер Лабораторної Роботи №3

Практичних Робіт №4

***Виконав:***

студент групи ШІ-14

Шевченко Данііл Вікторович

# **Тема роботи:**

Ознайомлення з одновимірними та багатовимірними масивами, ініціалізація одновимірних та багатовимірних масивів, принципи доступу до елементів масиву, вказівники та посилання, адресна арифметика, поняття про динамічні масиви, конструкції std::array та std::vector

# **Мета роботи:**

Ознайомитись з різними видами масивів, отримати поняття про багатовимірні масиви. Ознайомитесь з вказівниками та посиланнями(що реалізовані на їх основі), зрозуміти принцип роботи та випадки застосування вказівників та посилань, отримати поняття про адресну арифметику і її застосування для індексації масивів, ознайомитись з динамічними масивами та методом їх ініціалізації, отримати поняття про модифіковані масиви std::array та std::vector

Завдання:

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

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Одновимірні та багатовимірні статичні масиви
* Тема №2: Вказівники
* Тема №3: Посилання
* Тема №4: Адресна арифметика, доступ до елементів масива через вказівники
* Тема №5: Динамічні одновимірні масиви, початкове уявлення про динамічні багатовимірні масиви
* Тема №6: Конструкції std::array та std::vector

1. Індивідуальний план опрацювання теорії:

* Тема №1: Одновимірні та багатовимірні статичні масиви
  + Джерела Інформації
    - Практика в НУ “ЛП”
    - Курси https://acode.com.ua/
  + Що опрацьовано:
    - Ініціалізація одновимірних масивів
    - Ініціалізація багатовимірних масивів
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.11.23
  + Звершення опрацювання теми: 26.11.23
* Тема №2: Вказівники
  + Джерела Інформації
    - Практика в НУ “ЛП”
    - Курси https://acode.com.ua/
  + Що опрацьовано:
    - Поняття про адреси змінних в пам’яті
    - Поняття про застосування поінтерів для збереження адрес
    - Використання поінтерів в масивах та функціях
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.11.23
  + Звершення опрацювання теми: 26.11.23

Тема №3: Посилання

* Джерела Інформації
  + - Практика в НУ “ЛП”
    - Курси https://acode.com.ua/
  + Що опрацьовано:
    - Ознайомлення з посиланнями(референсами) та їх застосуванням як псевдонімів змінних
    - Поняття про принцип реалізації референсів через поінтери
    - Використання референсів в функціях
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.11.23
* Звершення опрацювання теми: 26.11.23
* Тема №4: Адресна арифметика, доступ до елементів масива через вказівники
* Джерела Інформації

Практика в НУ “ЛП”

* + - Курси https://acode.com.ua/
  + Що опрацьовано:
    - Зв’язок поняття поінтера з масивами
    - Доступ до елементів масива через вказівники
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.11.23
* Звершення опрацювання теми: 26.11.23
* Тема №5: Динамічні одновимірні масиви, початкове уявлення про динамічні багатовимірні масиви
* Джерела Інформації
  + - Практика в НУ “ЛП”
    - Курси https://acode.com.ua/
  + Що опрацьовано:
    - Застосування поінтерів для ініціалізації динамічних структур
    - Ініціалізація динамічного одновимірного масиву
    - Методи ініціалізації динамічного двовимірного масиву
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.11.23
* Звершення опрацювання теми: 26.11.23
* Тема №6: Конструкції std::array та std::vector

Джерела Інформації

* + - Практика в НУ “ЛП”
    - Курси https://acode.com.ua/
  + Що опрацьовано:
    - Std::array як модифікація статичного масиву
    - Std::vector як модифікація динамічного масиву
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.11.23
* Звершення опрацювання теми: 26.11.23

# **Виконання роботи:**

**VNS LAB4**

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

Завдання №1 VNS Lab4

Варіант 8

1) Сформувати одновимірний масив цілих чисел, використовуючи генератор

випадкових чисел.

2) Роздрукувати отриманий масив.

3) Знищити всі елементи із заданим значенням.

4) Додати перед кожним парним елементом масиву елемент зі значенням 0.

5) Роздрукувати отриманий масив.

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

#include <iostream>

#include <cstdlib>

#include <vector>

using namespace std;

bool presence(int x, int array[], int size)

{

bool inarray = false;

for (int i = 0; i < size; i++)

{

if (x == array[i])

{

inarray = true;

}

}

return inarray;

}

int\* erase(int array[], int n, int erasevalues[], int m, int& count)

{

vector<int> newarray;

newarray.resize(n);

int j = 0;

for (int i = 0; i < n; i++)

{

if (!presence(array[i], erasevalues, m))

{

newarray[j] = array[i];

j++;

}

}

newarray.resize(j);

int\* newarray2 = new int[j];

for (int i = 0; i < j; i++)

{

newarray2[i] = newarray[i];

}

count = j;

return newarray2;

}

int\* modifiedarray(int array[], int n, int &count)

{

vector<int> modified;

modified.resize(2\*n);

int j = 0;

for (int i = 0; i < n; i++)

{

if (array[i] % 2 == 0)

{

modified[j] = 0;

modified[j + 1] = array[i];

j = j + 2;

}

else

{

modified[j] = array[i];

j++;

}

}

modified.resize(j);

int\* newarray3 = new int[j];

count = j;

for (int i = 0; i < j; i++)

{

newarray3[i] = modified[i];

}

return newarray3;

}

int main()

{

srand(time(nullptr));

int N;

cout << "Enter the length of an array: ";

cin >> N;

int\* myarray = new int[N];

for (int i = 0; i < N; i++)

{

myarray[i] = rand() % 100 - 50;

}

cout << "The array is: " << endl;

for (int i = 0; i < N; i++)

{

cout << myarray[i] << " ";

}

cout << endl;

int M;

cout << "Enter the amount of elements you want to erase: ";

cin >> M;

int\* erasearray = new int[M];

for (int i = 0; i < M; i++)

{

cin >> erasearray[i];

}

int erasecount = 0;

myarray = erase(myarray, N, erasearray, M, erasecount);

cout << "The length of the new array is " << erasecount << endl;

cout << "The new array with erased elements is: " << endl;

for (int i = 0; i < erasecount; i++)

{

cout << myarray[i] << " ";

}

cout << endl;

int modifiedcount = 0;

myarray = modifiedarray(myarray, erasecount, modifiedcount);

cout << "The length of the modified array is: " << modifiedcount << endl;

cout << "The new array after adding 0 before even numbers: " << endl;

for (int i = 0; i < modifiedcount; i++)

{

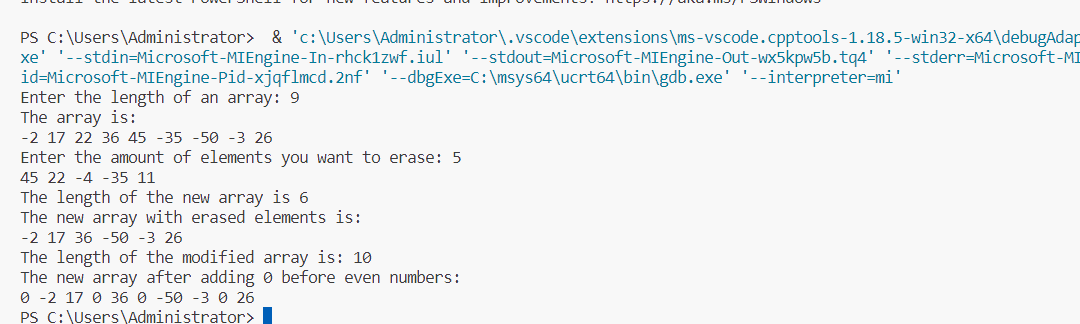
cout << myarray[i] << " ";

}

cout << endl;

}

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



Час затрачений на виконання завдання – 45 хв

**VNS LAB5**

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

Завдання №2 VNS Lab5

* Варіант 23

23. Задано одновимірний масив, який складається з N цілих чисел. Сформувати на його основі двовимірний масив N x N так, щоб сума елементів у першому стовпці дорівнювала першому елементу одновимірного масиву, сума елементів у другому стовпці повинна дорівнювати другому елементу одновимірного масиву й т.ін. Нулі не використовувати.

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

#include <iostream>

#include <cstdlib>

using namespace std;

int\*\* modifier(int array[], int size)

{

srand(time(nullptr));

int\*\* array2 = new int\* [size];

for (int i = 0; i < size; i++)

{

array2[i] = new int[size];

}

int\*\* random = new int\*[size-1];

for (int i = 0; i < size-1; i++)

{

random[i] = new int[size];

}

for (int i = 0; i < size-1; i++)

{

for (int j = 0; j < size; j++)

{

random[i][j] = rand() % 20 - 10;

if (random[i][j] == 0)

{

random[i][j] = 1;

}

}

}

for (int i = 0; i < size-1; i++)

{

for (int j = 0; j < size; j++)

{

array2[i][j] = random[i][j];

}

}

int\* sumarray = new int[size];

for (int k = 0; k < size; k++)

{

sumarray[k] = 0;

}

for (int j = 0; j < size; j++)

{

for (int i = 0; i < size - 1; i++)

{

sumarray[j] = sumarray[j] + array2[i][j];

}

}

for (int k = 0; k < size; k++)

{

array2[size - 1][k] = array[k] - sumarray[k];

}

return array2;

}

int main()

{

int N;

cin >> N;

int\* input = new int[N];

for (int i = 0; i < N; i++)

{

cin >> input[i];

}

int\*\* ourarray = new int\* [N];

for (int i = 0; i < N; i++)

{

ourarray[i] = new int[N];

}

ourarray = modifier(input, N);

cout << "THE MATRIX IS:" << endl;

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

if (ourarray[i][j] > 0)

{

cout << "+";

}

cout << ourarray[i][j] << " ";

}

cout << endl;

}

}

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

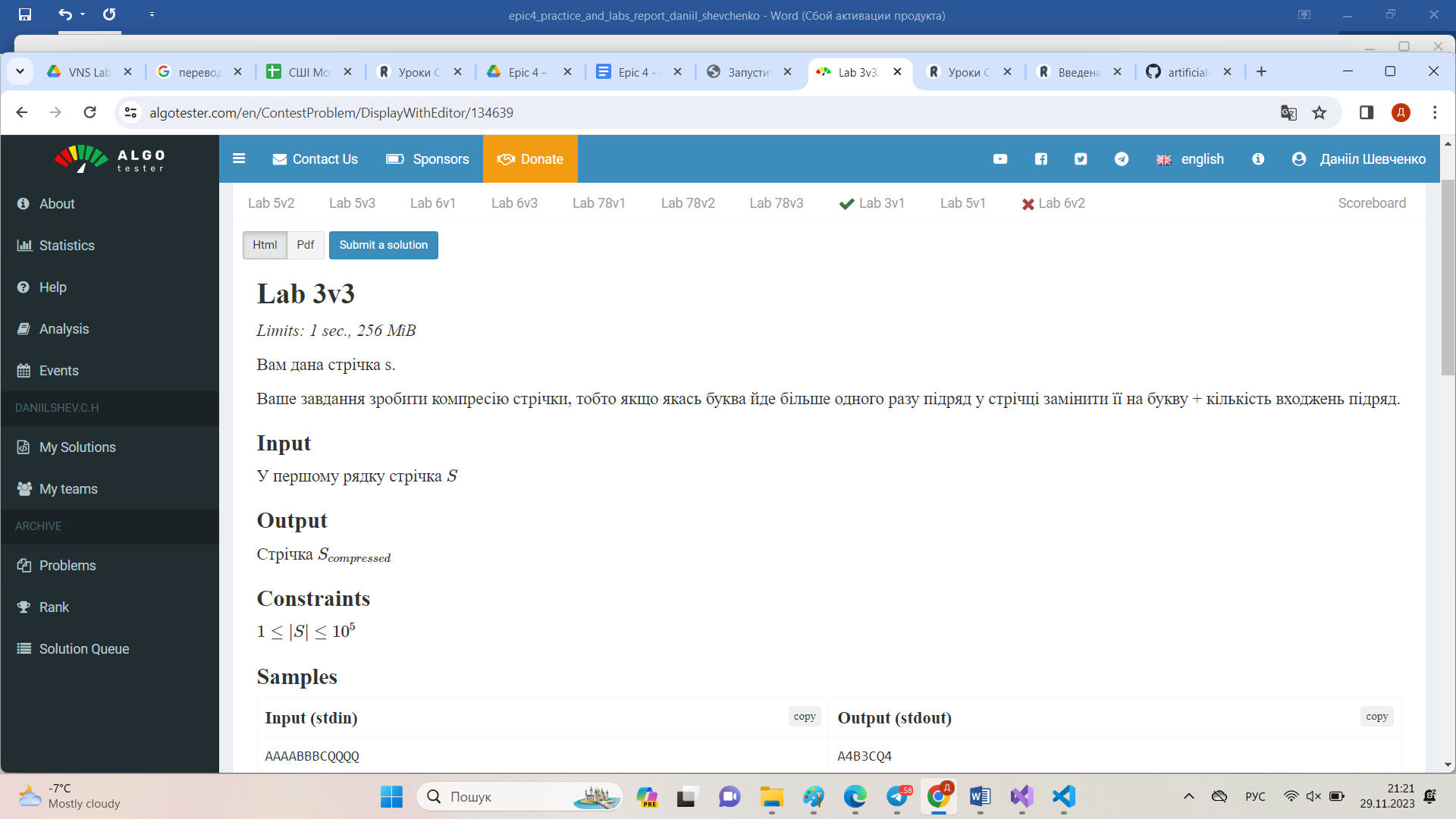


Час затрачений на виконання завдання – 60 хвилин

**Algotester Lab3 V3**

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

Завдання 3 Algotester Lab3 V3



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

#include <iostream>

#include <string>

using namespace std;

int main()

{

string S;

getline(cin, S);

string S2;

S2 = "";

long size = S.size();

long index = 0;

while (index < size)

{

long count = 1;

long index2 = index;

while (S[index2] == S[index2+1])

{

count++;

index2++;

}

string COUNT = to\_string(count);

char Element = S[index];

if (count > 1)

{

S2 = S2 + Element + COUNT;

}

else

{

S2 = S2 + Element;

}

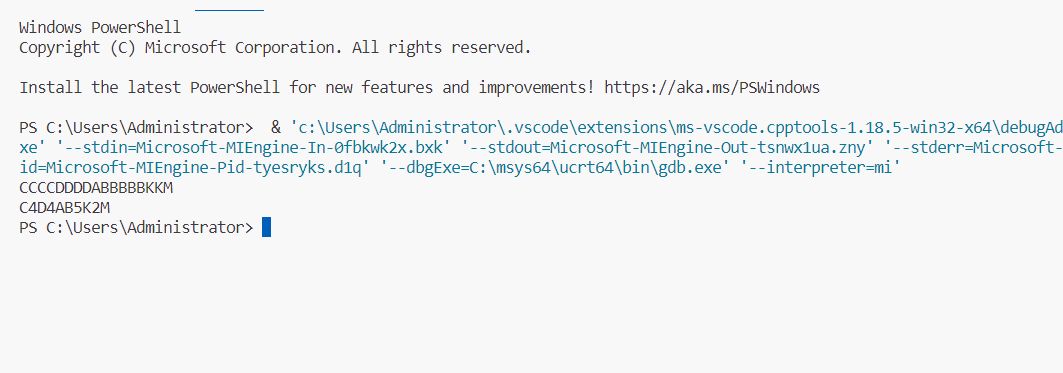
index = index + count;

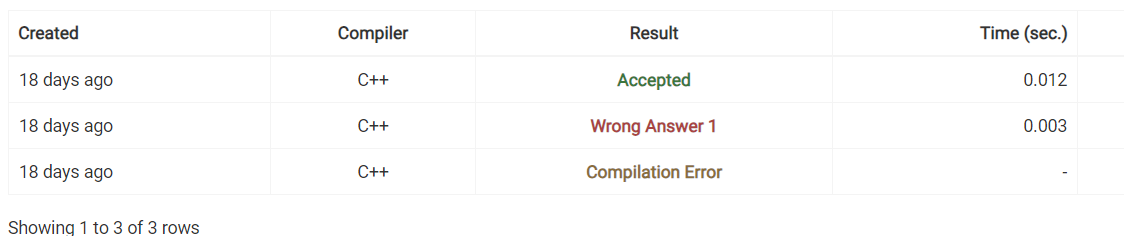
}

cout << S2;

## }

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





Час затрачений на виконання завдання – 1 година

**Practice Task # 4**

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

Завдання №4 Class-Practice Task # 4

# **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:
   1. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:
   1. *bool isPalindrome(const string& str, int start, int end);*
3. Перевантаження функцій:
   1. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
   2. *bool isPalindrome*(ціле число);
4. Рекурсія:
   1. 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. Перетворити ціле число на рядок і використовуйте рядкову версію функції, щоб перевірити, чи це паліндром.

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

#include <iostream>

#include <string>

#include <stack>

using namespace std;

bool IsPalindrome(string word, int start, int end)

{

if (start > end)

{

return true;

}

if (word[start] != word[end])

{

return false;

}

if (word[start] == word[end])

{

return IsPalindrome(word, start + 1, end - 1);

}

}

bool IsPalindrome(int number, int start, int end)

{

string word;

word = to\_string(number);

if (start > end)

{

return true;

}

if (word[start] != word[end])

{

return false;

}

if (word[start] == word[end])

{

return IsPalindrome(word, start + 1, end - 1);

}

}

int main()

{

string strinput;

int intinput;

string intinput2;

cout << "Enter a string: ";

getline(cin, strinput);

cout << "Enter a number: ";

cin >> intinput;

intinput2 = to\_string(intinput);

int strsize = strinput.size();

int intsize = intinput2.size();

if (IsPalindrome(strinput, 0, strsize - 1))

{

cout << "The word is a palindrome" << endl;

}

else

{

cout << "The word is not a palindrome" << endl;

}

if (IsPalindrome(intinput, 0, intsize - 1))

{

cout << "The number is a palindrome" << endl;

}

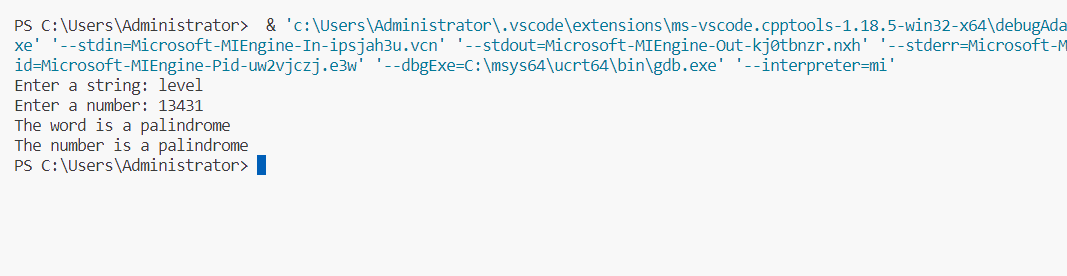
else

{

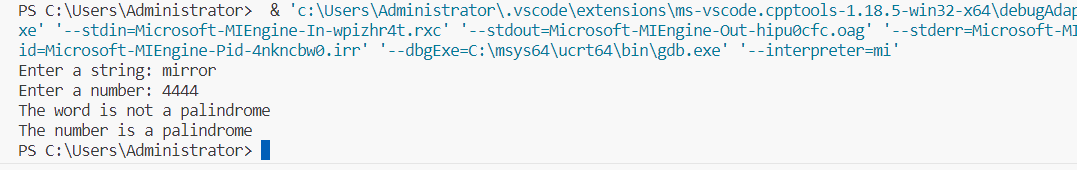
cout << "The number is not a palindrome" << endl;

}

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





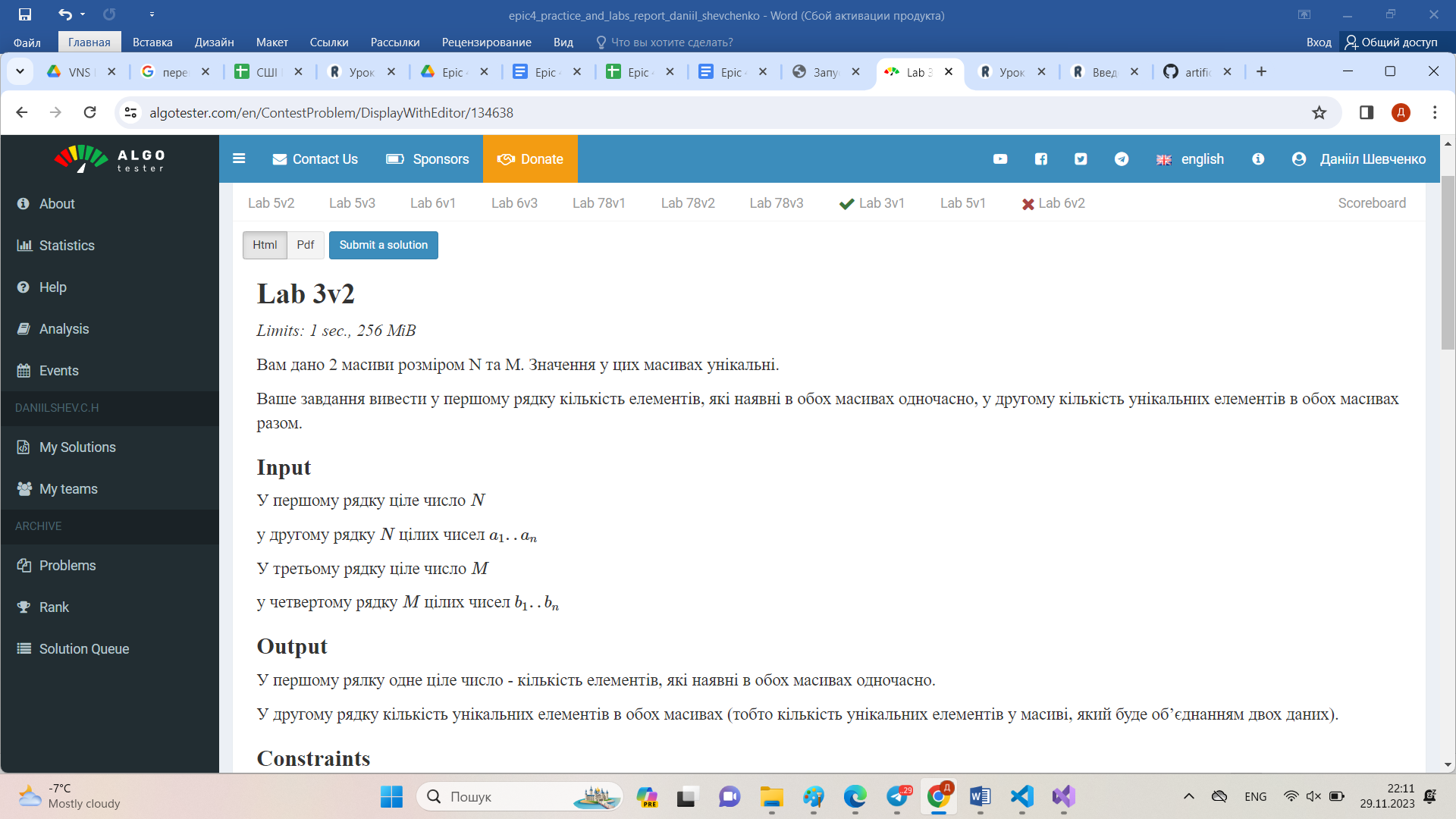


Час затрачений на виконання завдання – 60 хвилин

**Self-Practice(Algotester Lab3 V2)**

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

Завдання №5 Self-Practice(Algotester Lab3 V2)



## **2. Дизайн завдання:**

## **C:\Users\Administrator\Downloads\Диаграмма без названия.drawio.png**

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

#include <iostream>

#include <vector>

using namespace std;

int main()

{

int N, M;

cin >> N;

double\* array1 = new double[N];

for (int i = 0; i < N; i++)

{

cin >> array1[i];

}

cin >> M;

double\* array2 = new double[M];

for (int i = 0; i < M; i++)

{

cin >> array2[i];

}

int k = 0;

for (int i = 0; i < N; i++)

{

for (int j = 0; j < M; j++)

{

if (array1[i] == array2[j])

{

k++;

}

}

}

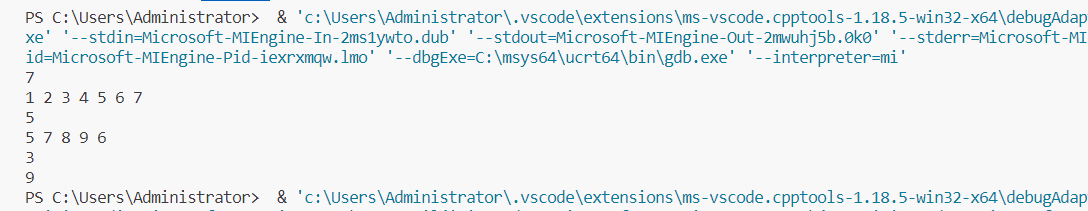
int f = N + M - k;

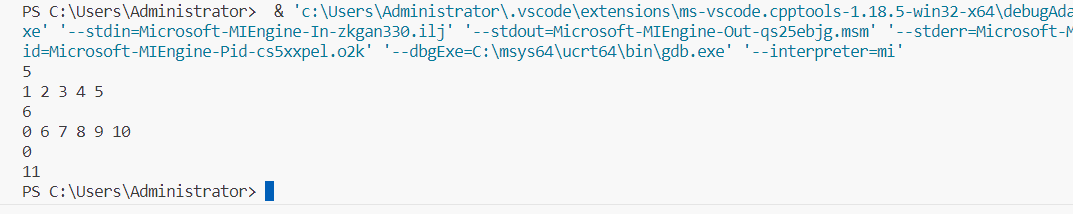
cout << k << endl;

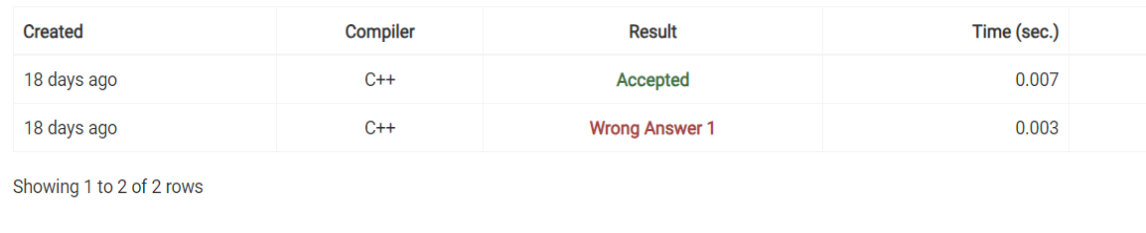
cout << f;

}

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



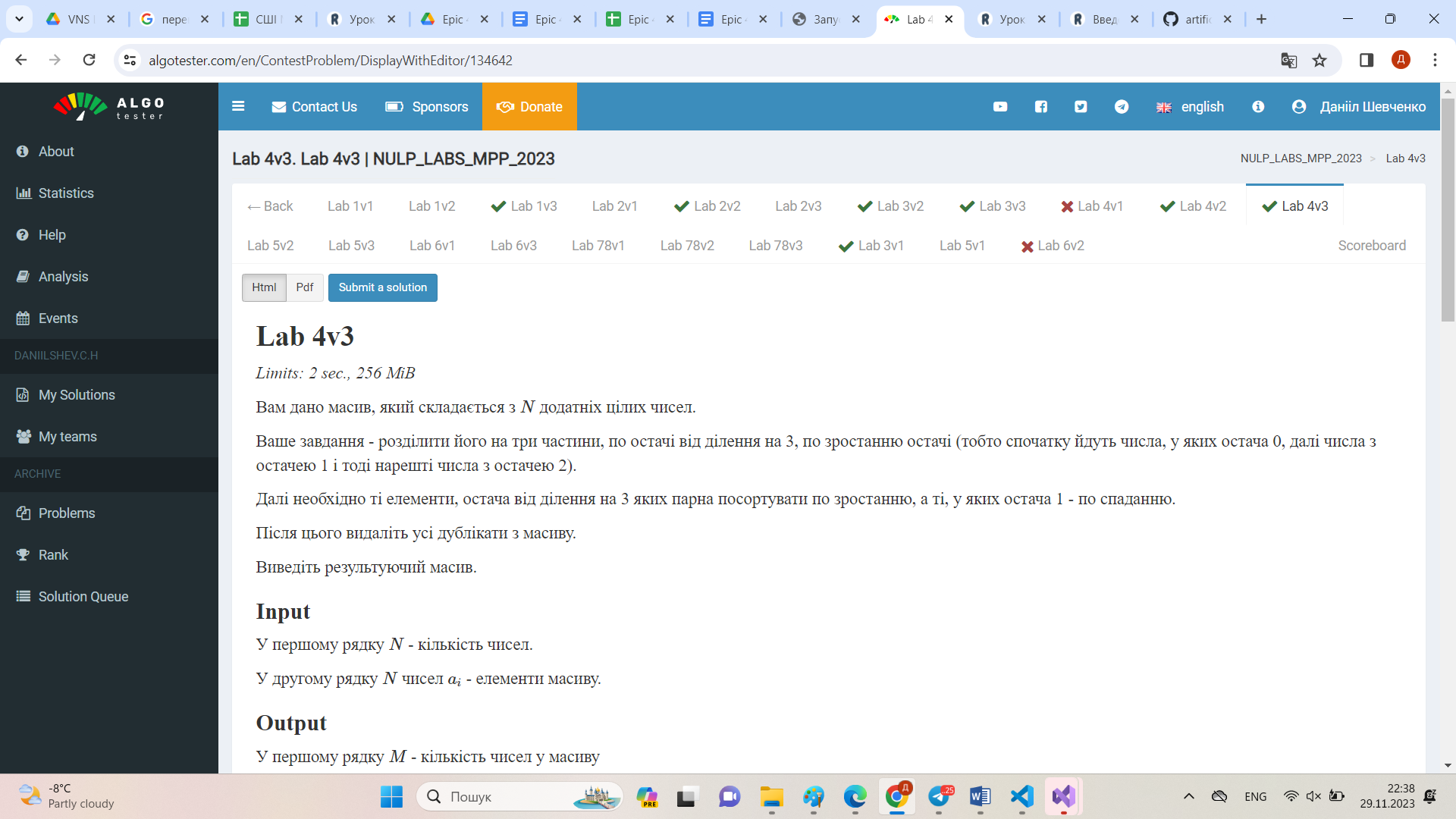




Час затрачений на виконання завдання – 40 хвилин  
**Self-Practice(Algotester Lab4V3)**

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

Завдання №6 Self-Practice(Algotester Lab4 V3)



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

#include <iostream>

using namespace std;

void swap(int& a, int& b)

{

int x = a;

a = b;

b = x;

}

void upgradesort(int array[], int length)

{

for (int startindex = 0; startindex < length; startindex++)

{

int minindex = startindex;

for (int currentindex = startindex; currentindex < length; currentindex++)

{

if (array[currentindex] < array[minindex])

{

minindex = currentindex;

}

}

swap(array[startindex], array[minindex]);

}

}

void downgradesort(int array[], int length)

{

for (int startindex = 0; startindex < length; startindex++)

{

int maxindex = startindex;

for (int currentindex = startindex; currentindex < length; currentindex++)

{

if (array[currentindex] > array[maxindex])

{

maxindex = currentindex;

}

}

swap(array[startindex], array[maxindex]);

}

}

int main()

{

int N;

cin >> N;

int\* array = new int[N];

for (int i = 0; i < N; i++)

{

cin >> array[i];

}

int\* array0 = new int[N];

int\* array1 = new int[N];

int\* array2 = new int[N];

int l0, l1, l2;

l0 = 0;

l1 = 0;

l2 = 0;

for (int i = 0; i < N; i++)

{

if (array[i] % 3 == 0)

{

array0[l0] = array[i];

l0++;

}

}

for (int i = 0; i < N; i++)

{

if (array[i] % 3 == 1)

{

array1[l1] = array[i];

l1++;

}

}

for (int i = 0; i < N; i++)

{

if (array[i] % 3 == 2)

{

array2[l2] = array[i];

l2++;

}

}

upgradesort(array0, l0);

upgradesort(array2, l2);

downgradesort(array1, l1);

int\* newarray = new int[N];

for (int i = 0, k = 0; k < l0; i++, k++)

{

newarray[i] = array0[k];

}

for (int i = l0, k = 0; k < l1; i++, k++)

{

newarray[i] = array1[k];

}

for (int i = l0+l1, k = 0; k < l2; i++, k++)

{

newarray[i] = array2[k];

}

int\* finalarray = new int[N];

int newsize = 0;

for (int i = 0; i < N; i++)

{

bool presence = false;

for (int j = 0; j < newsize; j++)

{

if (finalarray[j] == newarray[i])

{

presence = true;

}

}

if (!presence)

{

finalarray[newsize] = newarray[i];

newsize++;

}

}

cout << newsize << endl;

for (int i = 0; i < newsize; i++)

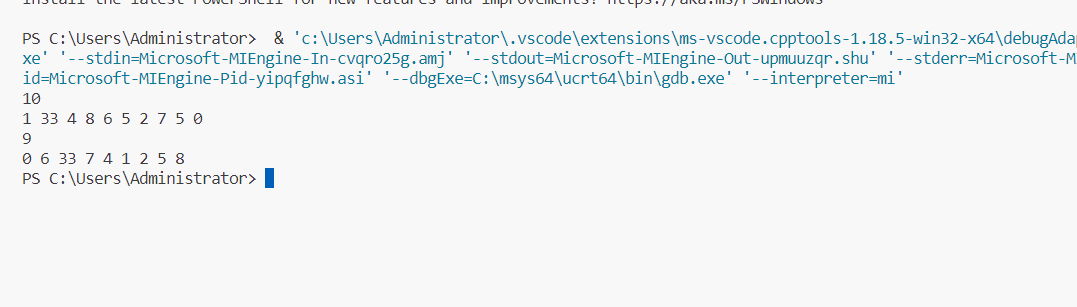
{

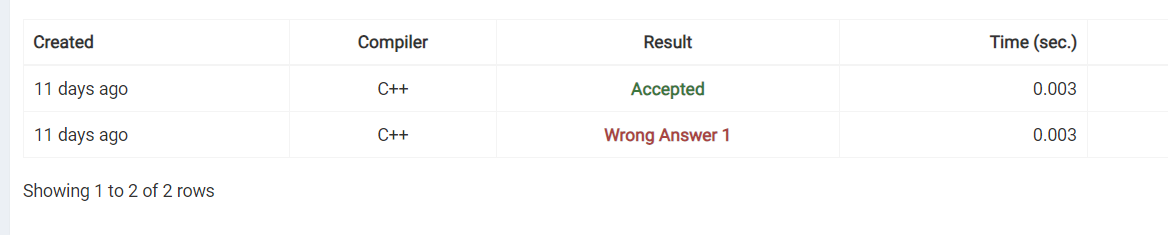
cout << finalarray[i] << " ";

}

}

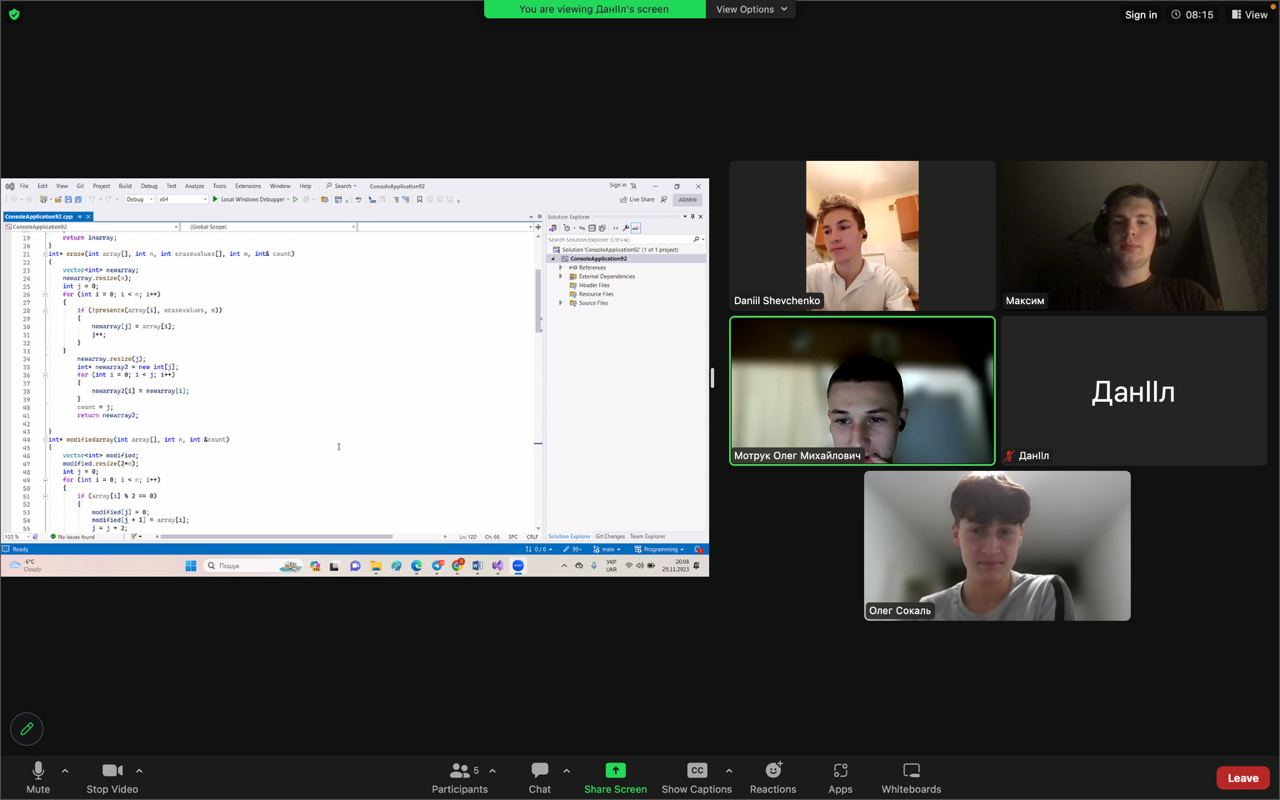
## **3. Результати виконання завдань, тестування та фактично затрачений час:**

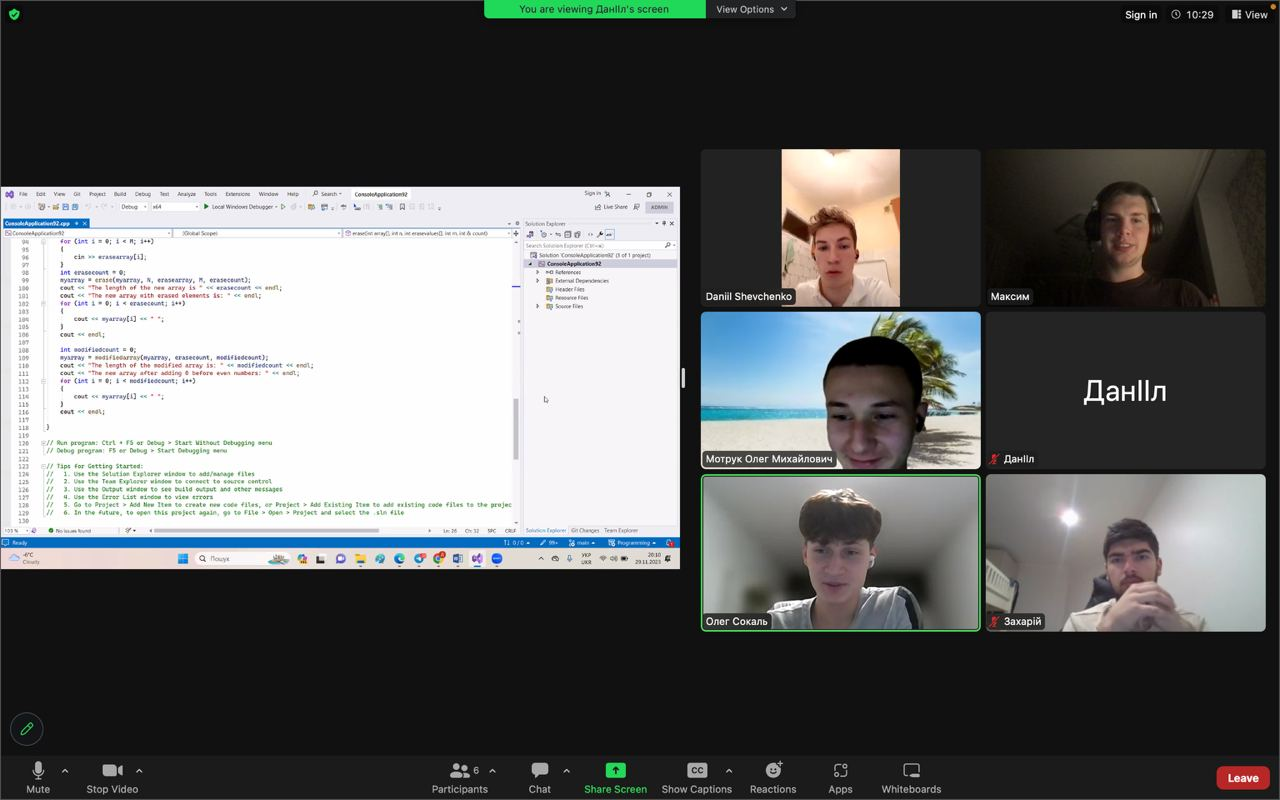


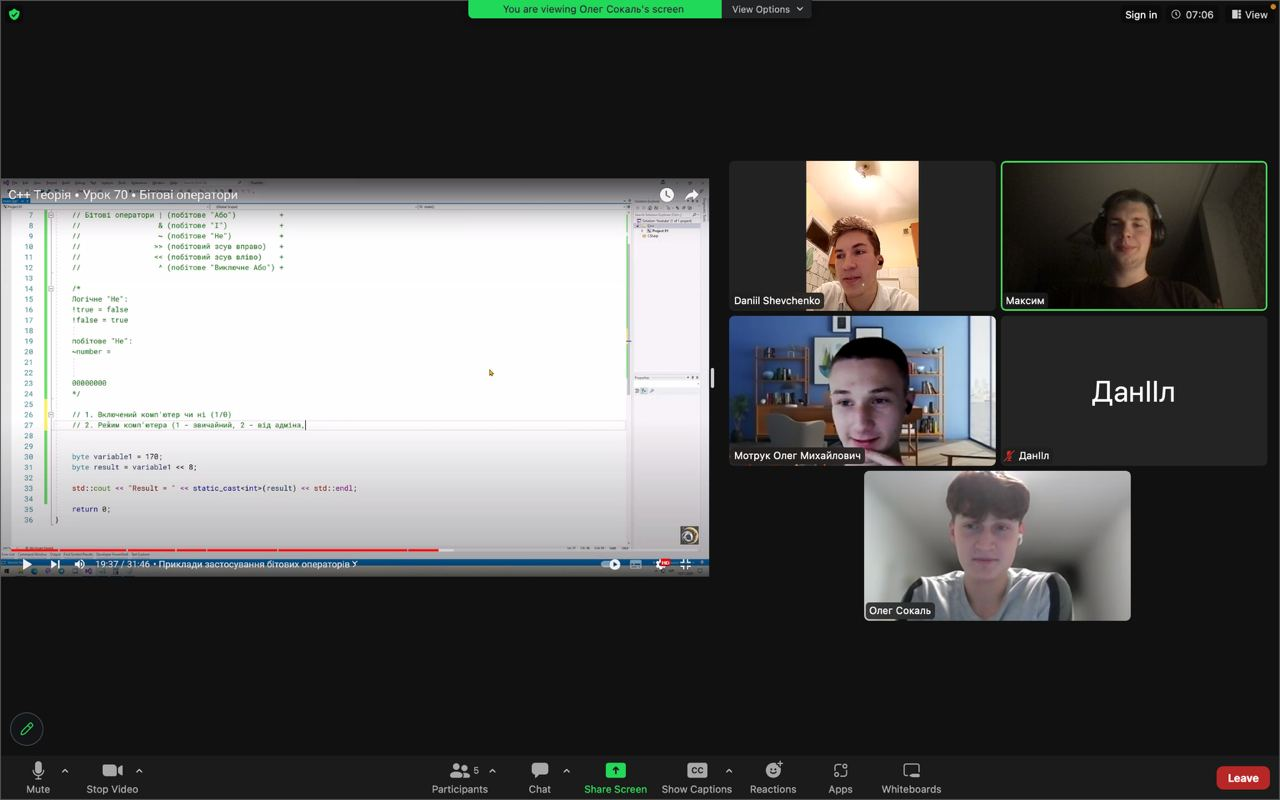


Час затрачений на виконання завдання – 3 години

**Докази командної роботи**







# **Висновки:**

Отже, в ході роботи над епіком 4 я отримав знання про те, що таке масиви і які вони бувають. Були вивчені одномірні та багатовимірні масиви, принципи їх ініціалізації та доступу до членів. Окремо було вивчено поняття поінтерів та референсів, зокрема як їх ефективно використовувати в функціях та масивах. Було розглянуто поняття адресної арифметики та доступу до елементів масиву через вказівники. Я вивчив поняття динамічних масивів,як їх ініціалізовувати та яку роль в цьому грають вказівники. Крім того, в якості додаткової інформації я вивчив std::array та std::vector як модифіковані різновиди статичних та динамічних масивів.

Pull-Request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/563