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

Стожук Роман Ігорович

# **Тема роботи:**

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

# **Мета роботи:**

# Навчитися працювати з одновимірними та двовимірними масивами

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Одновимірні масиви
* Тема №2: Двовимірні Масиви

1. Індивідуальний план опрацювання теорії:

* Тема №1: Одновимірні масиви
  + Джерела Інформації
    - <https://acode.com.ua/urok-77-masyvy/>
    - <https://acode.com.ua/urok-78-fiksovani-masyvy/>
    - <https://acode.com.ua/urok-79-masyvy-i-tsykly/>
    - <https://acode.com.ua/urok-82-ryadky-c-style/>
  + Що опрацьовано:
    - Ініціалізацію, доступ до елементів, передачу в функції
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.09.2023
  + Звершення опрацювання теми: 16.09.2023
* Тема №2: Двовимірні масиви
  + Джерела Інформації:
    - <https://acode.com.ua/urok-81-bagatovymirni-masyvy/>
  + Що опрацьовано:
    - Ініціалізацію, доступ до елементів, передачу в функції
  + Статус: Ознайомлений
  + Початок опрацювання теми: 16.09.2023
  + Звершення опрацювання теми: 17.09.2023

# **Виконання роботи:**

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

Завдання №3 VNS Lab 4

* Варіант завдання: 6
* 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел. 2) Роздрукувати отриманий масив.
* 3) Знищити елемент із заданим номером.
* 4) Додати після першого парного елемента масиву елемент зі значенням M[ I-1 ]+2.
* 5) Роздрукувати отриманий масив.
* При виконанні роботи використовуються статичні масиви. Для організації статичних масивів із псевдозмінними межами необхідно оголосити масив досить великої довжини, наприклад, 100 елементів: int N=100; int a[N]; Потім користувач вводить реальну довжину масиву (не більше N) і працює з масивом тієї довжини, що він сам вказав. Інші елементи не розглядаються.

Завдання №4 VNS Lab 5

* Варіант завдання: 6
* Елемент матриці є сідловою точкою, якщо він є найменшим у своєму рядку й найбільшим у своєму стовпці (або навпаки: найбільшим у своєму рядку й найменшим у своєму стовпці). Для заданої матриці визначити всі сідлові точки
* Масив повинен передаватися у функцію як параметр

Завдання №5 Algotester Lab 2

* Варіант завдання: 2
* У вас є масив r 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 Nnew−1 (розмір нового масиву після видалення елементів), який буде відображати суми сусідніх елементів нового масиву. Далі необхідно вивести масив сум на екран.

# Обмеження: 1≤N≤10000; 0≤ri≤10000; 0≤a,b,c≤100000

Завдання №6 Algotester Lab 3

* Варіант завдання: 2
* Вам дано 2 масиви a і b розміром відповідно 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.
* Обмеження:

0≤N≤100

0≤ai≤100

0≤M≤100

0≤bi≤100

Завдання №7 Class Practice Task

* 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»).
* 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.

Завдання №8.1 Self Practice Task 0023

* Колись давним-давно під час довгого нічного пленарного засідання Верховної Ради жіноча депутатська половина помітила, що не всі чоловіки-депутати прийшли у краватках. "Яке обурливе жахіття!" — подумали вони. Більше того, ті депутати-самці, що мали краватки, не узгодили їхню довжину. І це ми називаємо обличчям країни?!

Тож не гаючи часу, жіночки зібрали усі наявні краватки та вирішили змайструвати з них нові краватки для усіх чоловіків на засіданні. Їхня мета — краватки однакової довжини на усіх депутатах, що називають себе чоловіками. Для цього жінки можуть розрізати краватки ножицями довільну кількість разів. Поряд з цим депутатки хочуть зробити однакові нові краватки якомога довшими.

* Обмеження:

1≤n,m≤1000

1≤ai≤

Завдання №8.2 Self Practice Task 1401

* Зеник і Марічка наступного року складають ЗНО. Готуй сани влітку, а воза — взимку. Тому вони хочуть почати підготовку якнайшвидше. Аби підготовка була цікавішою, вони вирішили запропонувати своїм однокласникам приєднатися. Їхні однокласники не поспішають готуватися до ЗНО, а натомість готуються до олімпіад з інформатики на Алготестері. Зеник і Марічка мають відвідати кожного однокласника і переконати в тому, що треба готуватися до ЗНО, добре його скласти, вступити до університету, а там уже можна готуватися і до студентських олімпіад. На щастя, їм не доведеться ходити по всьому місту, адже всі однокласники живуть на одній вулиці. З одного боку вулиці всі будинки мають парні номери, а з іншого — непарні. Усього треба відвідати n однокласників. Зеник і Марічка будуть відвідувати їх у такому порядку, в якому вони записані у списку учнів класу. i-ий у списку однокласник мешкає в будинку з номером ai. Зеник і Марічка пропонують вам теж приєднатися до підготовки. Для початку розв’яжіть простеньку задачку — порахуйте, скільки разів їм доведеться перейти дорогу.
* Обмеження: 1≤n≤, 1≤ai≤

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

Програма №8.1 Self Practice Task 0023

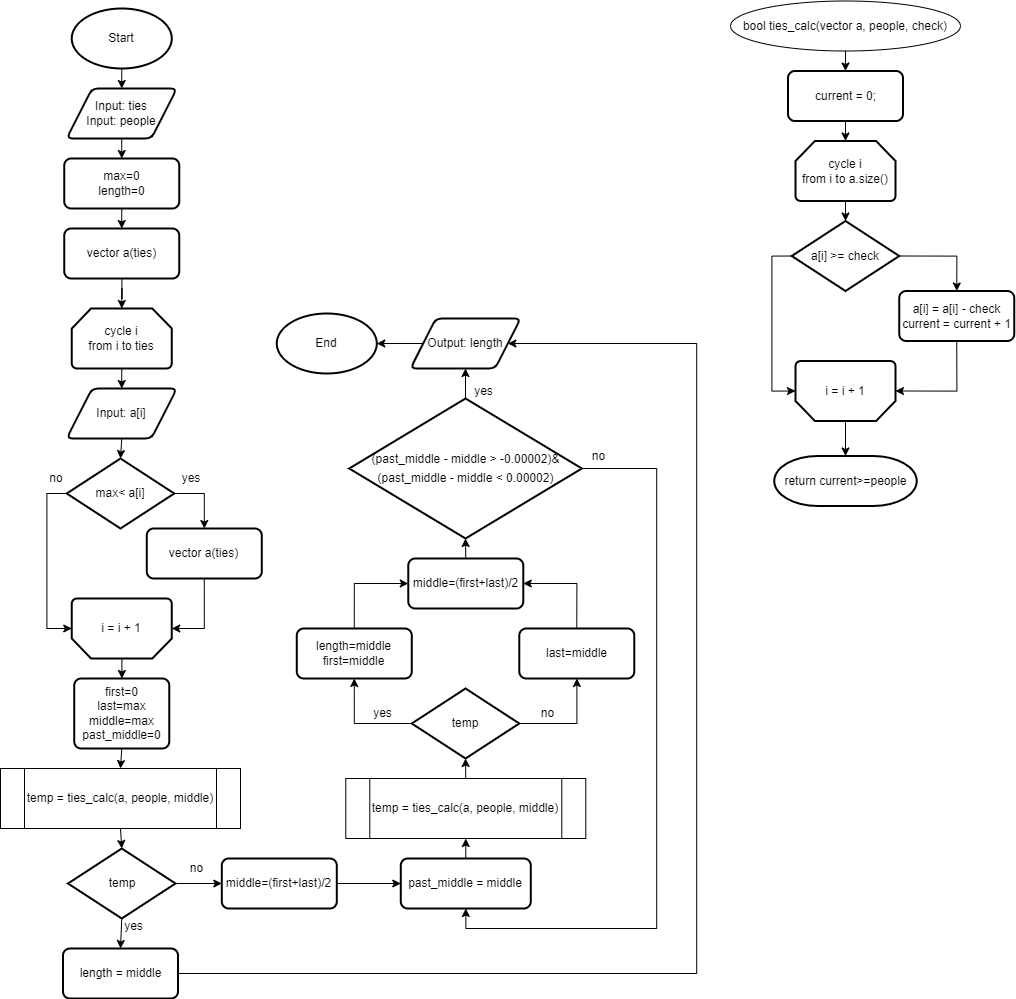
* 

Рисунок Блок схема для програми №8.1

* Планований час на реалізацію: 55хв

## **3. Конфігурація середовища до виконання завдань:**

## Змін у конфігурацію середовища не вносив.

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

Завдання №3 VNS Lab 4

#include <iostream>

#include <ctime>

#include <cstdlib>

int main() {

    const short n=100;

    int array[n];

    std::cout << "Enter length of array: \n";

    short size;

    std::cin >> size;

    srand(time(0));

    std::cout << "Random-generated array: \n";

    for (short i = 0; i < size; i++) {

        array[i] = rand();

        std::cout << array[i] << "  ";

    }

    std::cout << "\nEnter number of element you want to erase: \n";

    short destroy;

    std::cin >> destroy;

    destroy--, size--;

    for (; destroy < size; destroy++) { array[destroy] = array[destroy+1]; }

    for (short i = 0; i <size; i++) {

        if (array[i] % 2 == 0) {

            for (short b = size; b > i+1; b--) { array[b] = array[b-1]; }

            size++, i++;

            array[i] = array[i-1] + 2;

            break;

        }

    }

    std::cout << "Updated array: \n";

    for (short i = 0; i<size; i++) {

        std::cout << array[i] << "  ";

    }

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/631/files#diff-61384b39bb167ecdf8ea82efbfc8375d8afe3b174282e52cf042ff9eb2416591>

Завдання №4 VNS Lab 5

#include <iostream>

bool saddle\_point(int input, int \*column, int \*row, int size) {

    int c\_min = INT\_MAX, c\_max = INT\_MIN, r\_min = INT\_MAX, r\_max = INT\_MIN;

    for (int i = 0; i < size; i++) {

        if (column[i] > c\_max) {c\_max = column[i];}

        if (column[i] < c\_min) {c\_min = column[i];}

        if (row[i] > r\_max) {r\_max = row[i];}

        if (row[i] < r\_min) {r\_min = row[i];}

    }

    if ((c\_min == input && r\_max == input) || (c\_max == input && r\_min == input)) {return true;}

    else {return false;}

}

int main() {

    std::cout << "Enter size of your matrix: ";

    int size;

    std::cin >> size;

    std::cout << "Enter your matrix:\n";

    int \*\*rows = new int\*[size];

    for (int r = 0; r < size; r++) {

        rows[r] = new int[size];

        for (int c = 0; c < size; c++ ) {

                std::cin >> rows[r][c];

        }

    }

    std::cout << "Saddle points: ";

    int \*cur\_col = new int[size];

    int \*cur\_row = new int[size];

    for (int r = 0; r < size; r++ ) {

        for (int c = 0; c < size; c++ ) {

            for (int temp = 0; temp < size; temp++) {

                cur\_row[temp] = rows[r][temp];

                cur\_col[temp] = rows[temp][c];

            }

            if(saddle\_point(rows[r][c], cur\_col, cur\_row, size)) {std::cout << rows[r][c] << " "; }

        }

    }

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/631/files#diff-19fb7f01bb11dd002e9a77448eaca184f5e1cf4e0e125bb2fc7113d1a8d261df>

Завдання №5 Algotester Lab 2

#include <iostream>

#include <vector>

int main() {

    long N;

    std::cin >> N;

    std::vector<long> r(N);

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

    long del[3];

    std::cin >> del[0] >> del[1] >> del[2];

    for (auto currentCheck : del) {

        for (long currentDelete = 0; currentDelete < r.size(); currentDelete++) {

            if (currentCheck == r[currentDelete]) { r.erase(r.begin() + currentDelete); break;}

        }

    }

    for (long sum = 0; sum < ((long)(r.size()) - 1); sum++) {

        r[sum] += r[sum+1];

    }

    if (r.size()) r.pop\_back();

    std::cout << r.size() << std::endl;

    for (auto print : r) { std::cout << print << ' '; }

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/631/files#diff-a13961e1860999a9e09334222597679c512bb76389baa096a1cec457947d765b>

Завдання №6 Algotester Lab 3

#include <iostream>

int main() {

    int n;

    std::cin >> n;

    int \*a = new int[n];

    for (int i = 0; i<n; i++) {

        std::cin>>a[i];

    }

    int m;

    std::cin >> m;

    int \*b = new int[m];

    for (int i = 0; i<m; i++) {

        std::cin>>b[i];

    }

    int not\_unique = 0;

    for (int i=0;i<n; i++) {

        for (int j=0;j<m;j++) {

            if (a[i] == b[j]) {

                not\_unique++;

                break;

            }

        }

    }

    int unique = m+n-not\_unique;

    std::cout << not\_unique << std::endl << unique;

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/631/files#diff-276f19409523c81b21062f464586623fd2a2e9881bbb946ea0b34ac3048bbaa7>

Завдання №7 Class Practice Task

#include <iostream>

#include <string>

using namespace std;

bool isPalindrome(const string& str, int start, int end) {

    if ((end - start > 1)) {

        if (!isPalindrome(str, 1+start, end-1))

        return 0;

    }

    if (str[start] == str[end]) { return 1; }

    else { return 0;}

}

bool isPalindrome(int number) {

    string abc = to\_string(number);

    return isPalindrome(abc, 0, abc.length()-1);;

}

int main() {

    string input;

    cin >> input;

    cout << (isPalindrome(input, 0, input.length()-1)?"Yes":"No") << endl << (isPalindrome(123421)?"Yes":"No");

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/631/files#diff-6c70f1ea662c2205e34b1a71928f476806df62f2e8a55690d49dec308b335f0e>

Завдання №8.1 Self Practice Task 0023

#include <iostream>

#include <vector>

#include <stdio.h>

bool ties\_calc(std::vector<double> a, short people, double check) {

    double current = 0;

    for (short i =0; i < a.size(); i++) {

        while (a[i] >= check) {

            a[i] -= check;

            current++;

        }

    }

    return (current >= people);

}

int main() {

    short ties, people;

    std::cin >> ties >> people;

    double max = 0, length = 0;

    std::vector<double> a(ties);

    for (short i = 0; i < ties; i++) {

        std::cin >> a[i];

        if (max < a[i]) max = a[i];

    }

    double first = 0, last = max, middle = max, past\_middle = 0;

    if (ties\_calc(a, people, middle)) length = middle; //first iteration

    else {

        middle = (last + first)/2;

        do {

            past\_middle = middle;

            if (ties\_calc(a, people, middle)) {

                length = middle;

                first = middle;

            }

            else {

                last = middle;

            }

            middle = (last + first)/2;

        } while (!((past\_middle - middle > -0.00002)&&(past\_middle - middle < 0.00002)));

    }

    printf("%.5lf", length);

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/631/files#diff-8d2006fd7e014ccb05641ab605c3e476716fb26796c45c86a01adba7ad3a9cea>

Завдання №8.2 Self Practice Task 1401

#include <iostream>

#include <vector>

int main() {

    long n, count = 0;

    std::vector<long> a;

    std::cin >> n;

    a.resize(n);

    for (int i = 0; i < n; i++) {

        std::cin >> a[i];

        if (i != 0) {

            if ((!(a[i]%2) && (a[i-1]%2)) || (!(a[i-1]%2) && (a[i]%2))) count++;

        }

    }

    std::cout << count;

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/631/files#diff-9e512155bf6456331366f6fb56ecb862a1b957751e28709cadd6cba4dc6ac797>

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

Завдання №3 VNS Lab 4

Ввід: 3 2

Вивід:

19894 11673 10122

19894 19896 10122

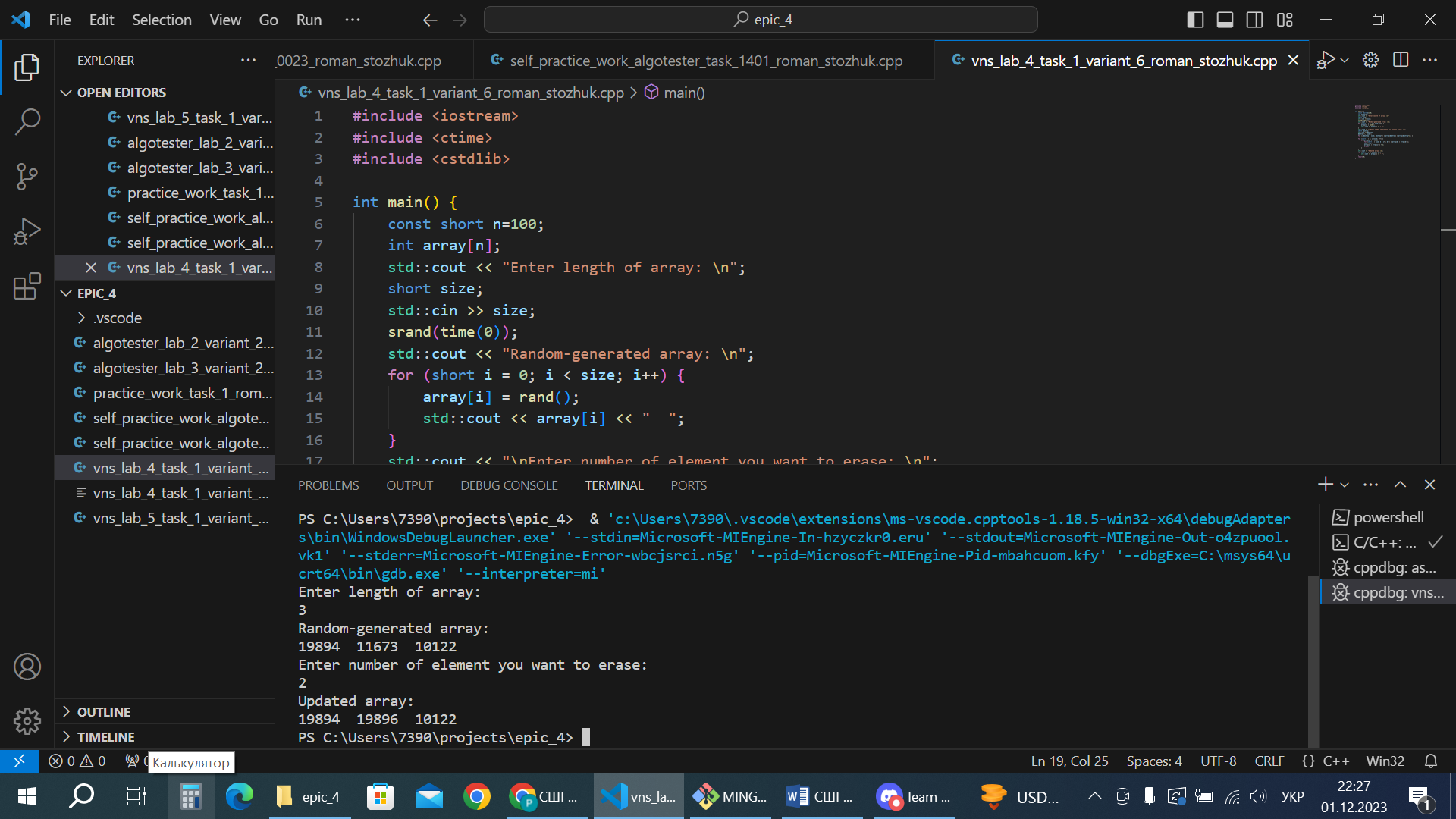


Рисунок Результат виконання завдання №3

Час затрачений на виконання завдання: 35хв

Завдання №4 VNS Lab 5

Ввід: 3 1 2 3 4 5 6 7 8 9

Вивід: 3 7

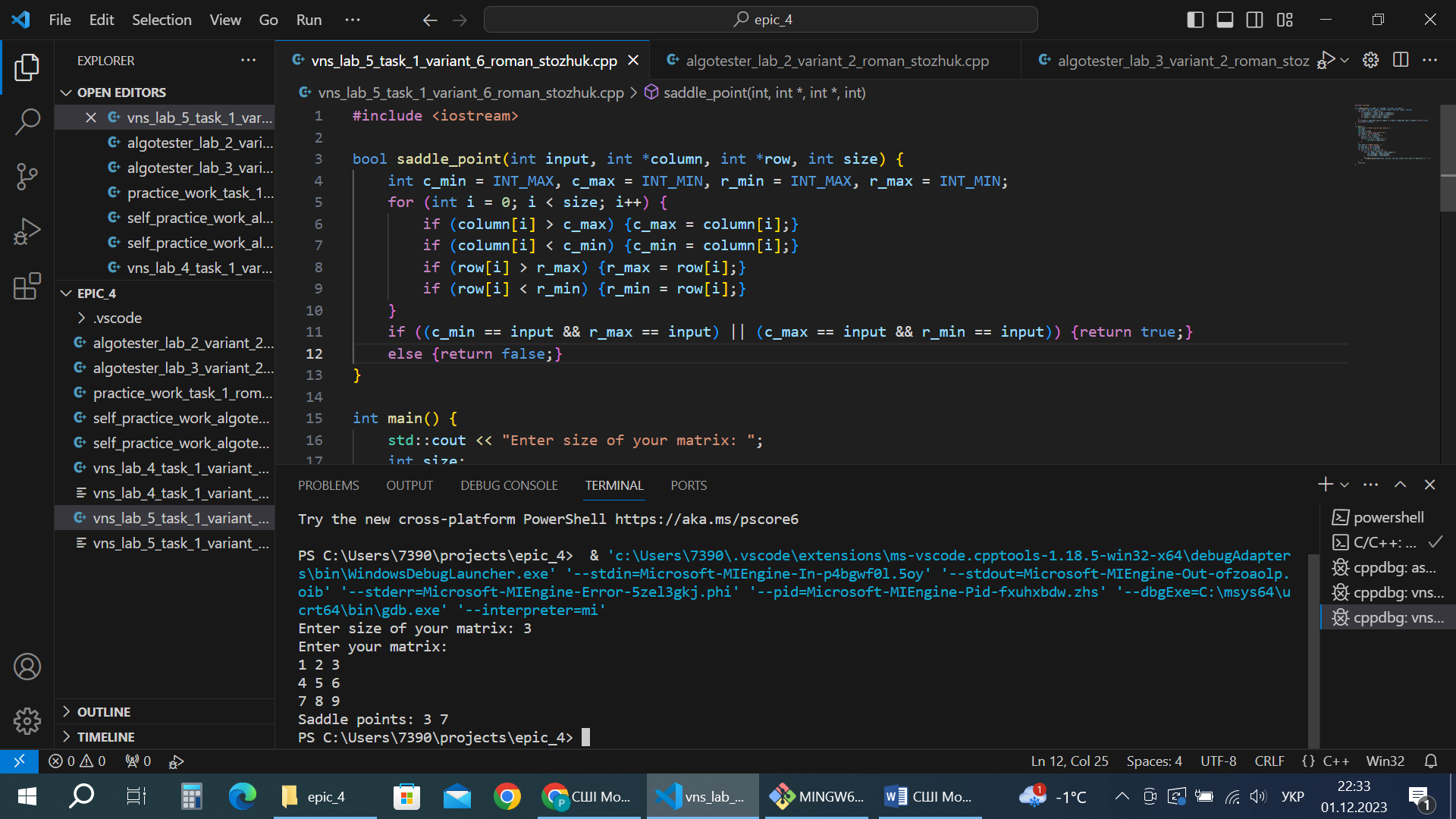


Рисунок Результат виконання завдання №4

Час затрачений на виконання завдання: 40хв

Завдання №5 Algotester Lab 2

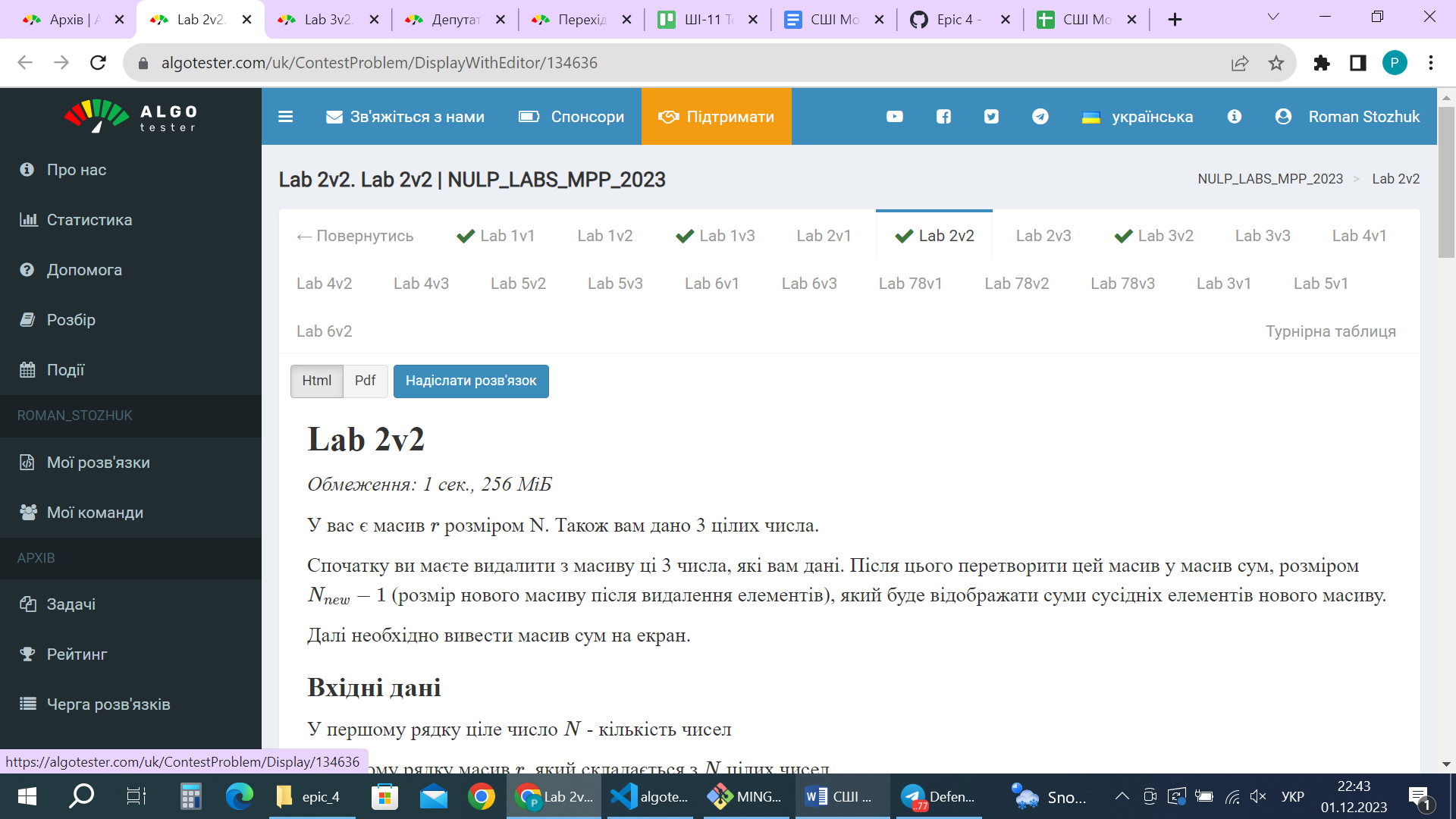


Рисунок Результат виконання завдання №5

Час затрачений на виконання завдання: 30хв

Завдання №6 Algotester Lab 3

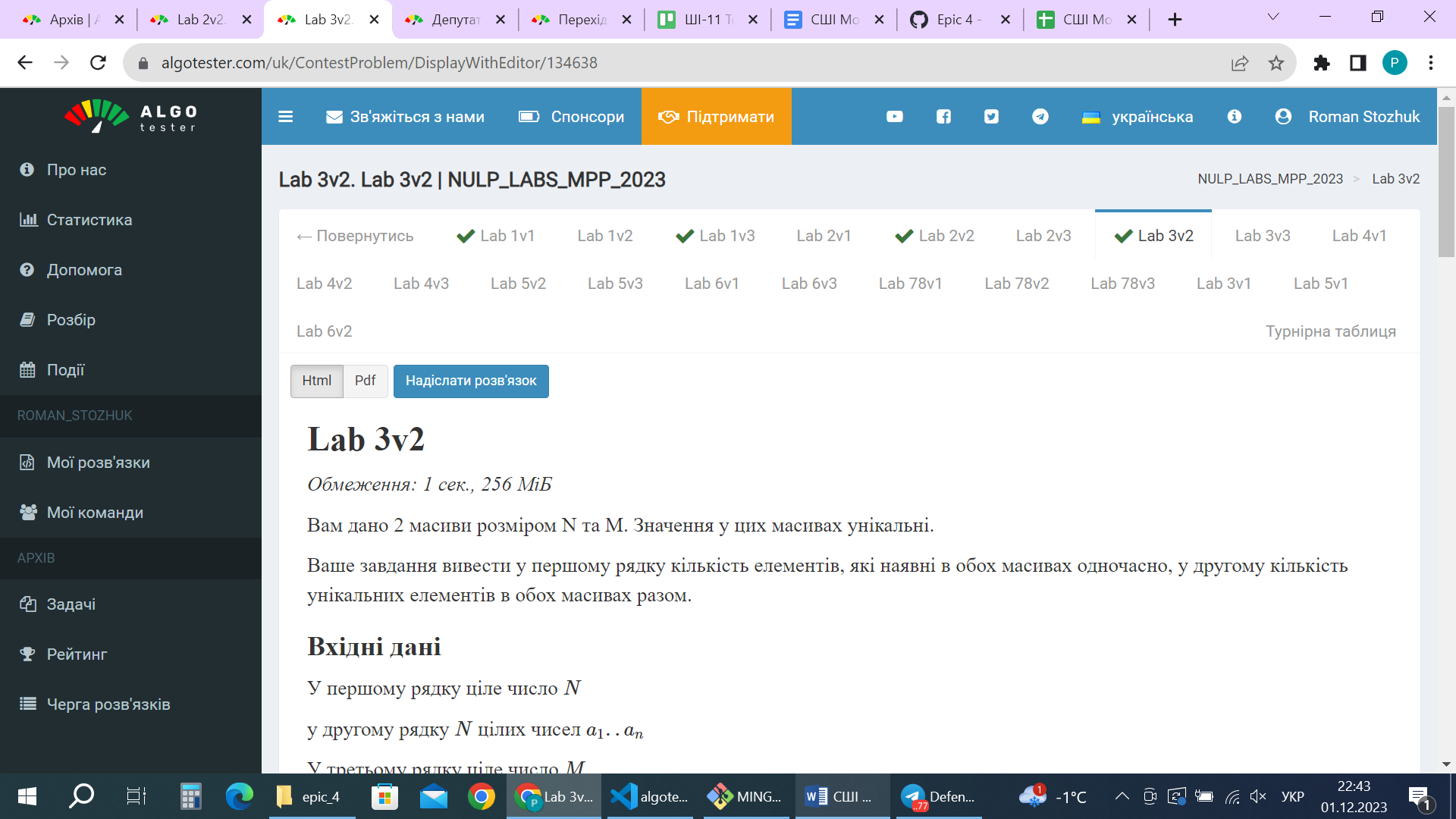


Рисунок Результат виконання завдання №6

Час затрачений на виконання завдання: 120хв

Завдання №7 Class Practice Task

Ввід: 1234321

Вивід: Yes No

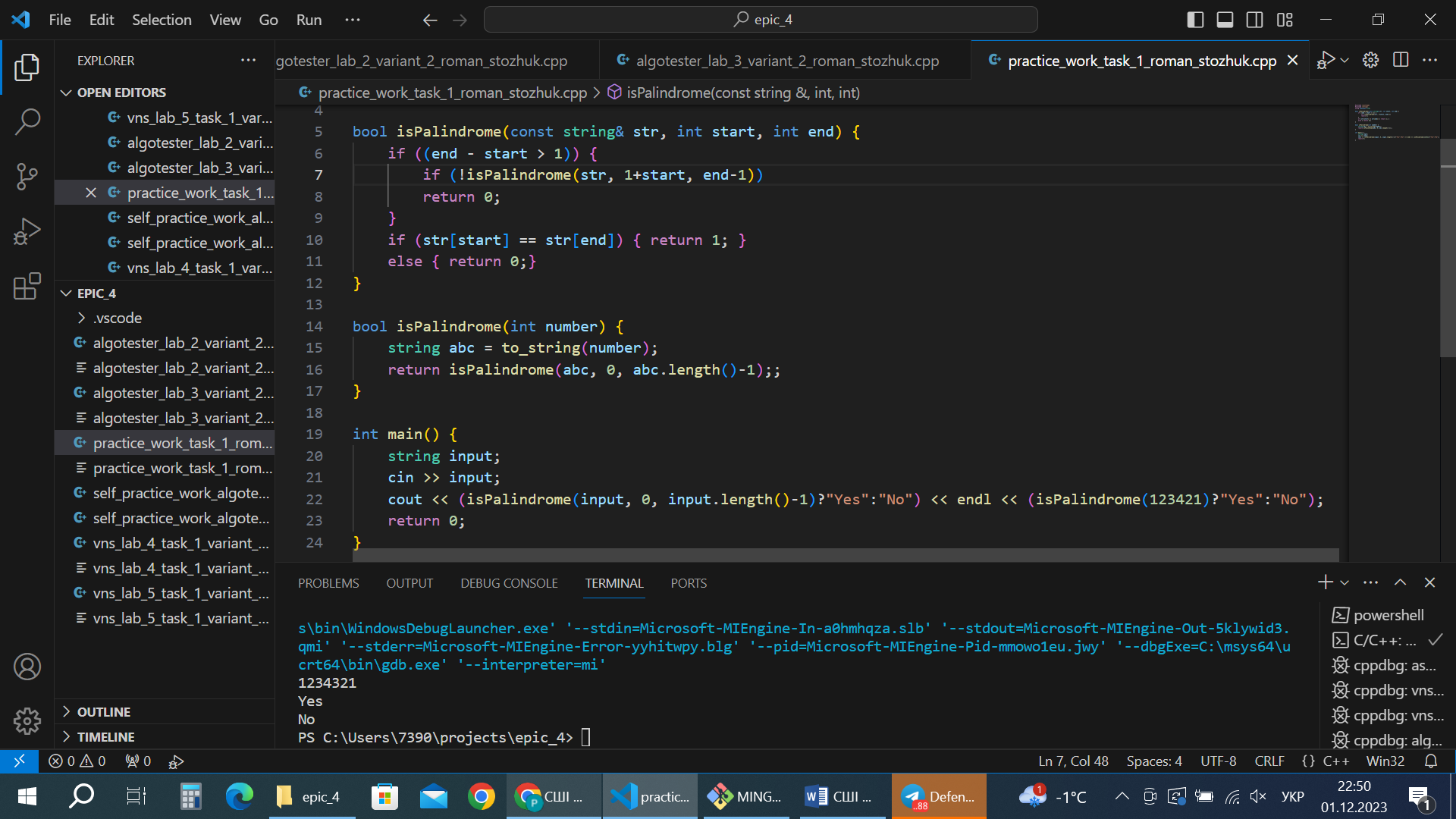


Рисунок Результат виконання завдання №7

Час затрачений на виконання завдання: 55хв

Завдання №8.1 Self Practice Task 0023

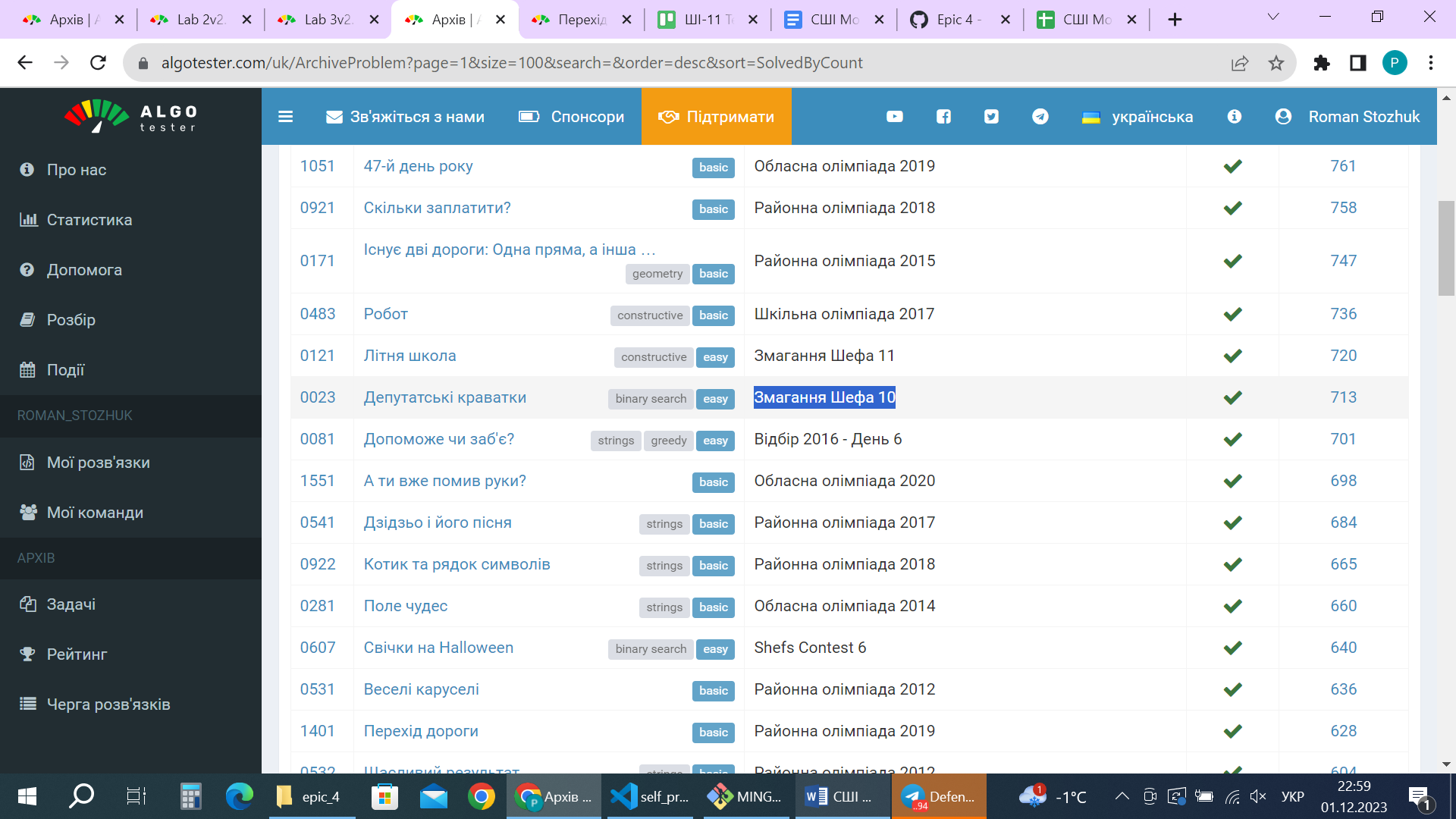


Рисунок Результат виконання завдання №8.1

Час затрачений на виконання завдання: 100хв

Завдання №8.2 Self Practice Task 1401

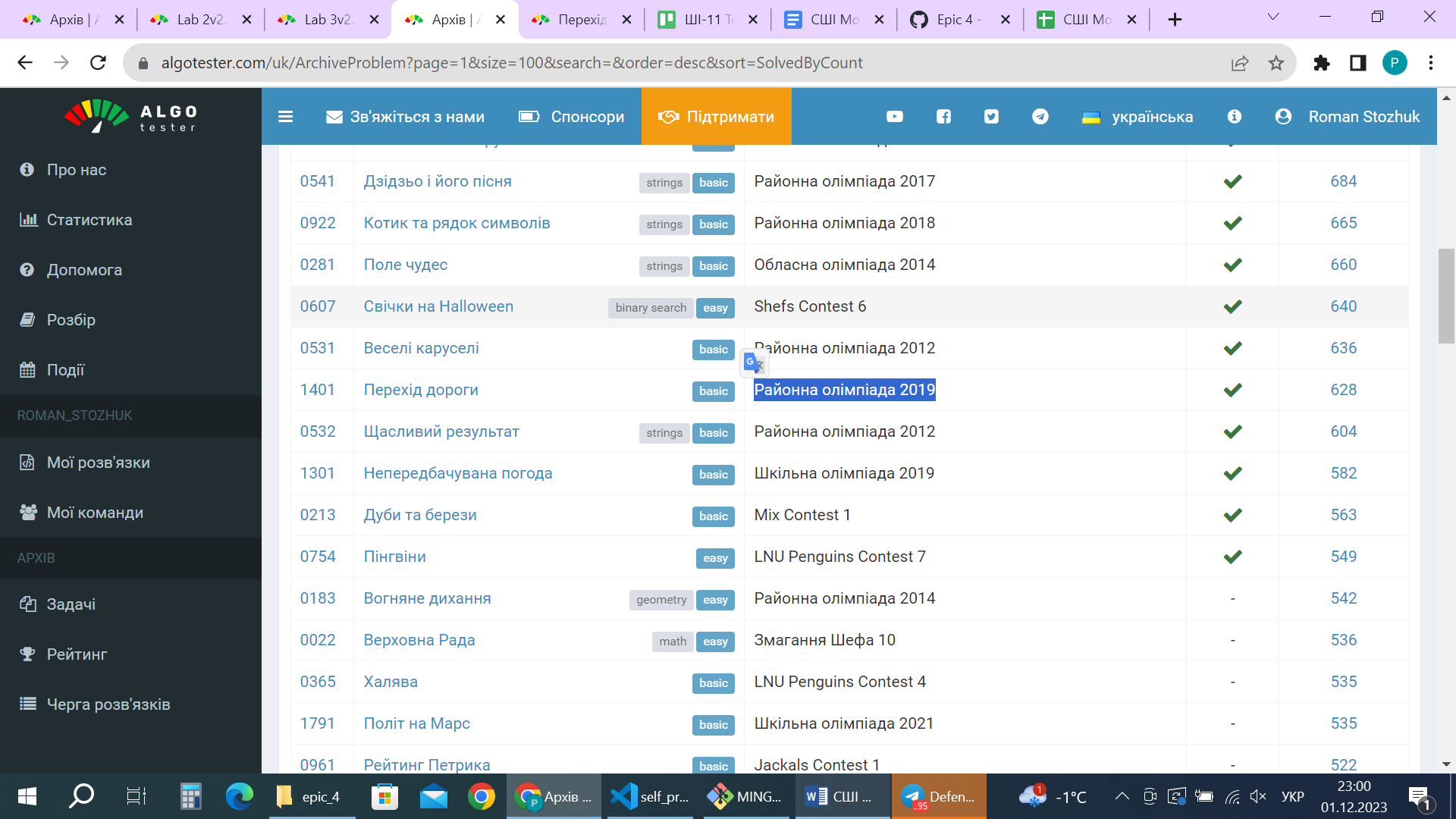
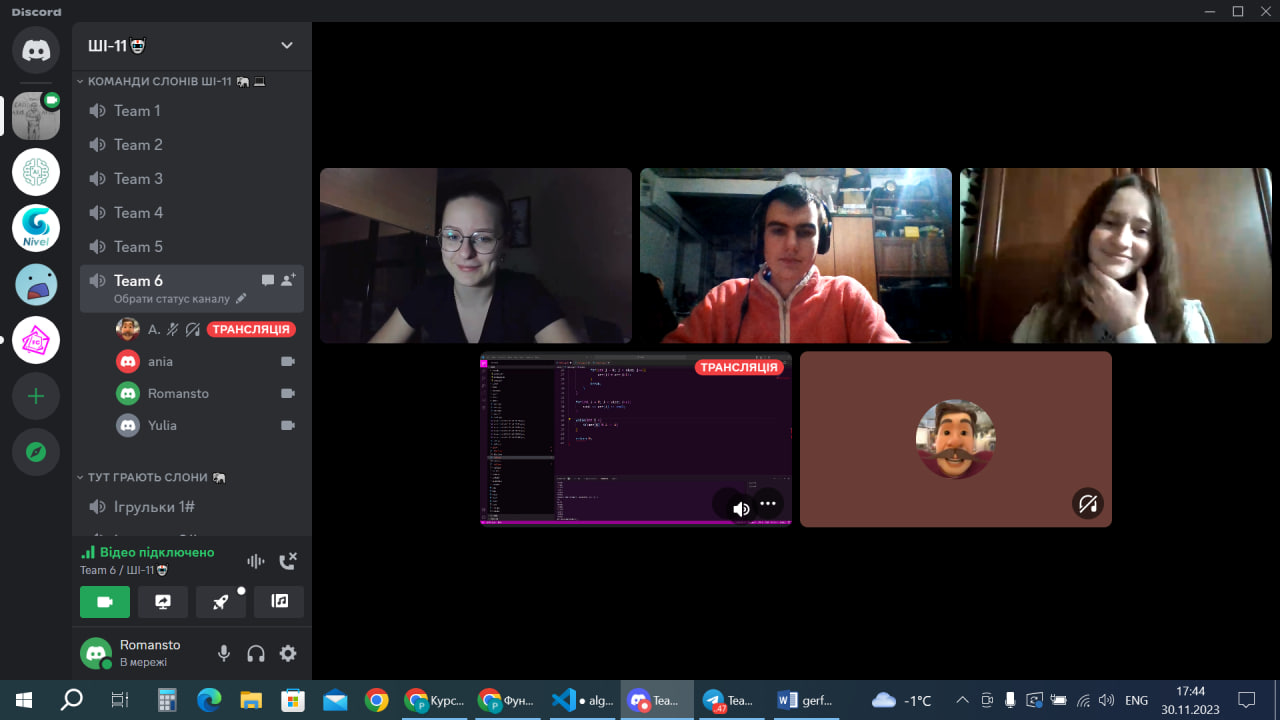
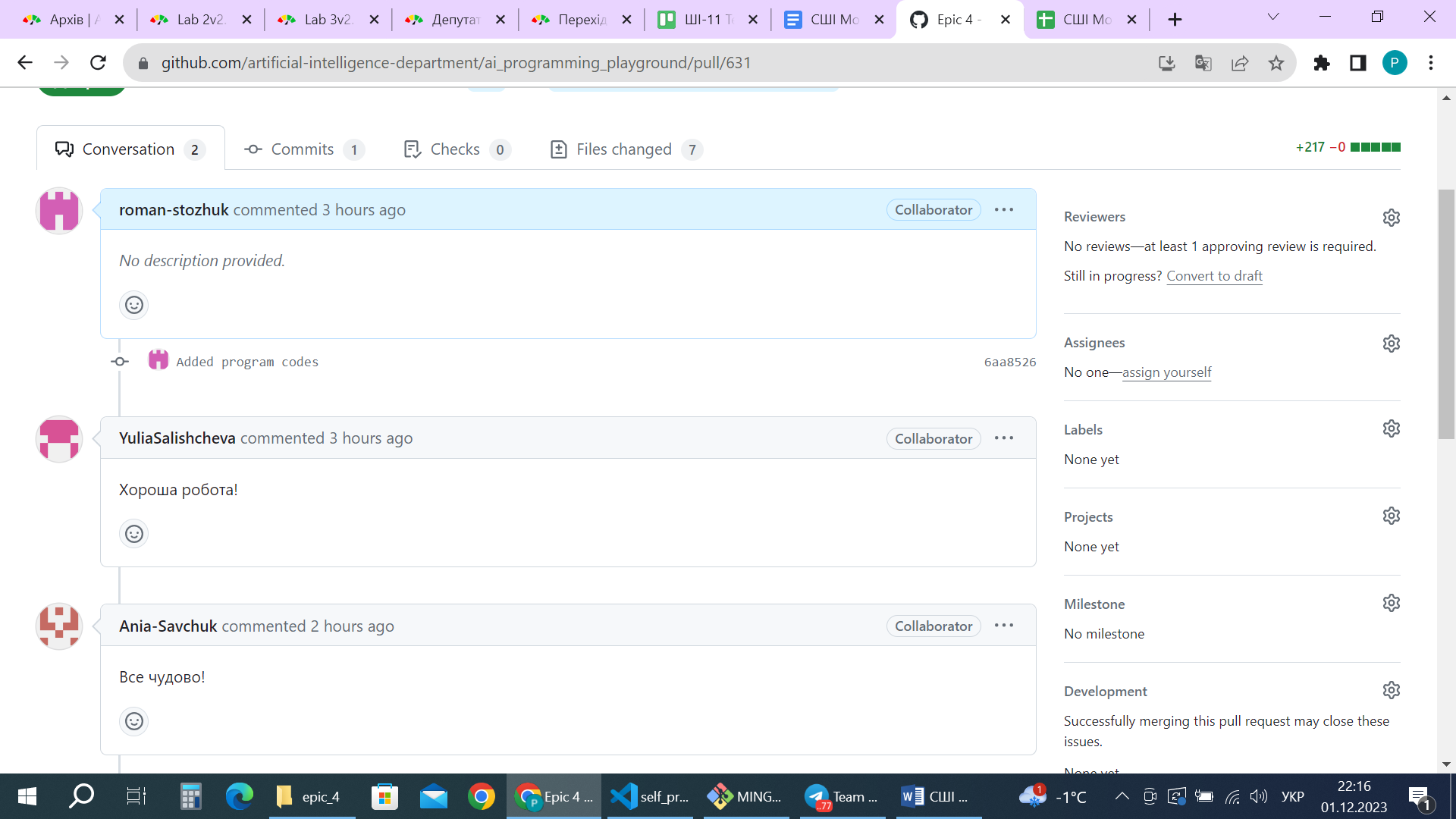


Рисунок Результат виконання завдання №8.2

Час затрачений на виконання завдання: 20хв

## **6. Кооперація з командою:**

* 
* 

# **Висновки:**

Протягом четвертого епіку я навчився працювати з одновимірними та двовимірними масивами