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.»

***Виконала:***

студент групи ШІ-12

Виклюк Яна Ярославівна

# **Тема роботи:**

Ознайомлення з простими структурами даних, одновимірними масивами, двовимірними масивами, векторами, алгоритмами обробки.

# **Мета роботи:**

Ознайомитись та навчитись працювати з простими структурами даних, одновимірними масивами, двовимірними масивами, векторами, алгоритмами обробки.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Масиви та їх види.
* Тема №2: Вектори.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли та їх види.
  + Джерела Інформації:
    - <https://acode.com.ua/urok-77-masyvy/>
    - <https://acode.com.ua/urok-78-fiksovani-masyvy/>
    - <https://acode.com.ua/urok-81-bagatovymirni-masyvy/>
  + Що опрацьовано:
    - Опрацьовано масиви, їх види, особливості роботи з ними.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 24.11
* Звершення опрацювання теми: 24.11
* Тема №2 : Вектори.
  + Джерела Інформації
    - [https://acode.com.ua/urok-99-vvedennya-v-std-vector/](https://acode.com.ua/urok-99-vvedennya-v-std-vector/ )
  + Що опрацьовано:
    - Опрацьовано вектори та особливості їх роботи.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 24.11
  + Звершення опрацювання теми: 24.11

# **Виконання роботи:**

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

Завдання №1 VNS Lab 4

* Варіант №8
* Деталі завдання
* Сформувати одновимірний масив цілих чисел, використовуючи генератор
* випадкових чисел, роздрукувати отриманий масив, знищити всі елементи із заданим значенням, додати перед кожним парним елементом масиву елемент зі значенням 0, роздрукувати отриманий масив.
* Важливі деталі для врахування в імплементації програми
* Уважно працювати з масивом при видаленні та додаванні елементів, змінюючи реальний розмір масиву.

Завдання №2 VNS Lab 5

* Варіант №8
* Деталі завдання
* Написати функцію транспонування квадратної матриці, з її допомогою визначити чи є задана матриця симетричною.
* Важливі деталі для врахування в імплементації програми
* Врахувати, що транспонування це зміна номеру рядка на номер стовпця елемента і навпаки. Матриця симетрична, якщо початкова матриця буде дорівнювати транспонованій.

Завдання №3 Algotester Lab 2

* Варіант №1
* Умова
* У вас є дорога, яка виглядає як N чисел.
* Після того як ви по ній пройдете - вашу втому можна визначити як різницю максимального та мінімального елементу.
* Ви хочете мінімізувати втому, але все що ви можете зробити - викинути одне число з дороги, тобто забрати його з масиву.
* В результаті цієї дії, яку мінімальну втому ви можете отримати в кінці дороги?

# Вхідні дані

* У першому рядку ціле число N - кількість чисел
* У другому рядку масив r, який складається з N цілих чисел
* Деталі завдання
* Вивести мінімальну втому, яку можна отримати.
* Важливі деталі для врахування в імплементації програми
* Зрозуміти, що потрібно викинути або мінімальне, або максимальне число, в залежності від того, де різниця буде меншою.

Завдання №4 Algotester Lab 3

* Варіант №3
* Умова
* Вам дана стрічка s.
* Ваше завдання зробити компресію стрічки, тобто якщо якась буква йде більше одного разу підряд у стрічці замінити її на букву + кількість входжень підряд.

# Вхідні дані

* У першому рядку стрічка S
* Деталі завдання
* Вивести компресовану стрічку.
* Важливі деталі для врахування в імплементації програми
* Врахувати, що спочатку потрібно додати елемент, а потім порахувати чи є ідентичні, якщо є додати число, якщо нема перейти далі.

Завдання №5 Class Practice Work

* Варіант завдання відсутній
* Деталі завдання
* Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.
* Важливі деталі для врахування в імплементації програми
* Написати функції, зробити перевантаження, використати рекурсію.

Завдання №6 Self Practice Work (Зуби)

* Варіант завдання відсутній
* Умова
* Мале Бісеня любить гострити зуби. А Зла Тітонька любить до нього підходити і питатися: «Що, зуби гостриш?». Бісеняті таке не дуже подобається, тому воно придумало робити таке.
* У Малого Бісеняти є n зубів. Кожен зуб має коефіцієнт загостреності ai. Також існує межа загостреності k. Якщо коефіцієнт загостреності певного зуба є більшим чи рівним межі загостреності, то такий зуб вважається загостреним.
* Мале Бісеня хоче наступного разу, коли Зла Тітонька його щось запитає, показати їй якнайбільше загострених зубів, що розташовані поспіль.
* Допоможіть Малому Бісеняті дізнатися, скільки найбільше зубів воно зможе показати.
* Вхідні дані
* У першому рядку задані два цілих числа n та k — кількість зубів та межа загостреності відповідно.
* В другому рядку задано n цілих чисел ai — коефіцієнти загостреності зубів.
* Деталі завдання
* Вивести єдине ціле число — відповідь на задачу.

Завдання №7 Self Practice Work (Algotester lab1v1)

* Варіант 1
* Умова
* У вашого персонажа є H хітпойнтів та M мани.
* Персонаж 3 рази використає закляття, кожне з яких може використати хітпойнти та ману одночасно.
* Якщо якесь закляття забирає і хітпойнти і ману - ваш персонаж програє, отже для виграшу треба використовувати при одному заклинанні **АБО**хітпойнти, **АБО** ману.
* Якщо в кінці персонаж буде мати додатню кількість хітпойнтів та мани (H,M>0) - він виграє, в іншому випадку програє.
* Ваше завдання у випадку виграшу персонажа вивести **YES**, вивести **NO** у іншому випадку.

# Вхідні дані

* 2 цілих числа H та M - хітпойнти та мана персонажа
* 3 рядки по 2 цілих числа, hi та mi - кількість хітпойнтів та мани, які ваш персонаж потратить за хід на i заклинання
* Деталі завдання
* Вивести YES - якщо ваш персонаж виграє
* NO - у всіх інших випадках

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

Програма №1 VNS Lab 4

* Планований час на реалізацію 60 хв
* Важливі деталі для врахування в імплементації
* Уважно працювати з масивом при видаленні та додаванні елементів, змінюючи реальний розмір масиву.

Програма №2 VNS Lab 5

* Блок-схема

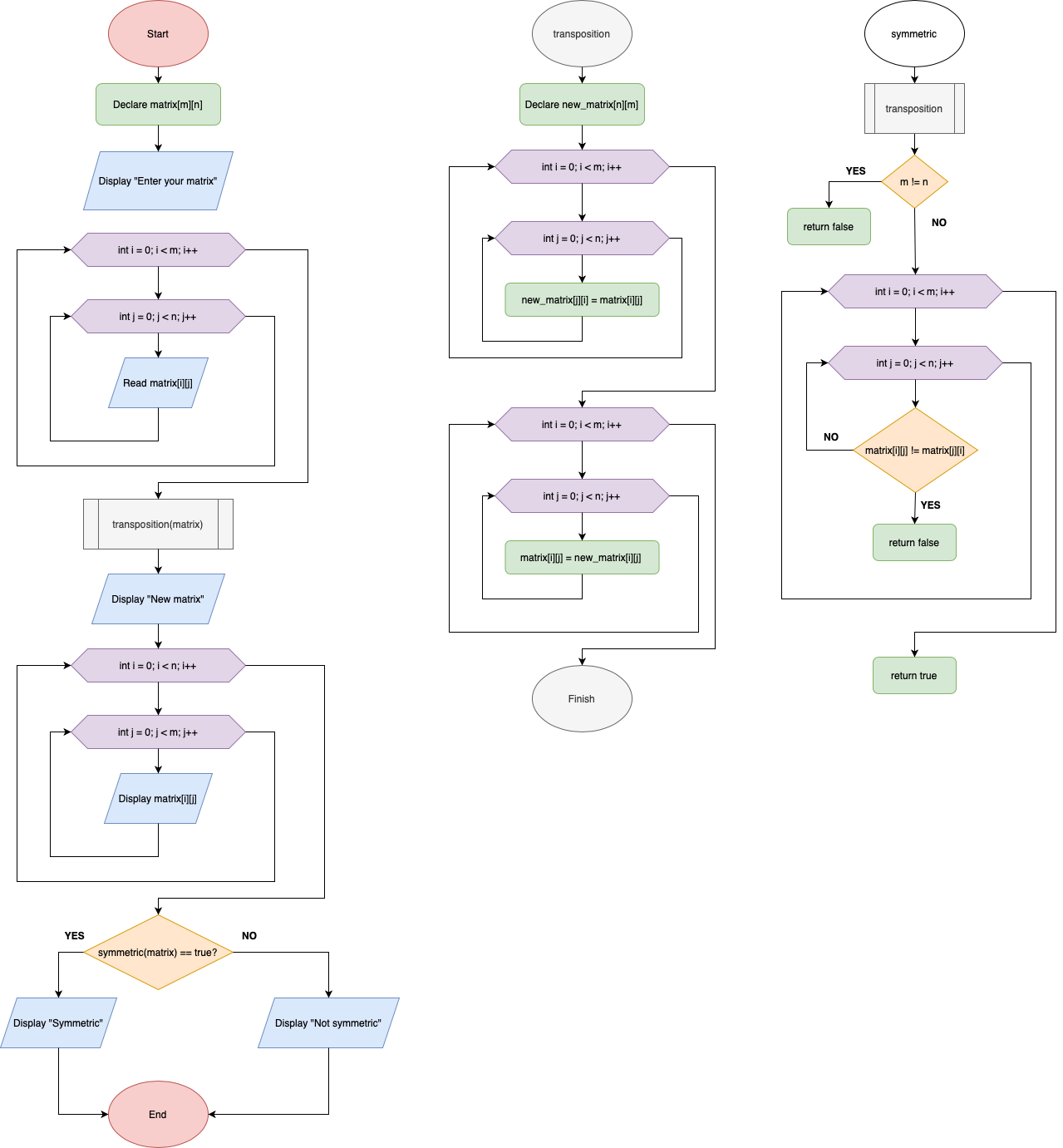


Рисунок 1 Блок-схема до програми №2

* Планований час на реалізацію 90 хв
* Важливі деталі для врахування в імплементації
* Врахувати, що транспонування це зміна номеру рядка на номер стовпця елемента і навпаки. Матриця симетрична, якщо початкова матриця буде дорівнювати транспонованій.

Програма №3 Algotester Lab 2

* Планований час на реалізацію 2 год
* Важливі деталі для врахування в імплементації
* Зрозуміти, що потрібно викинути або мінімальне, або максимальне число, в залежності від того, де різниця буде меншою

Програма №4 Algotester Lab 3

* Планований час на реалізацію 2 год
* Важливі деталі для врахування в імплементації
* Врахувати, що спочатку потрібно додати елемент, а потім порахувати чи є ідентичні, якщо є додати число, якщо нема перейти далі.

Програма №5 Class Practice Work

* Планований час на реалізацію 45 хв
* Важливі деталі для врахування в імплементації
* Написати функції, зробити перевантаження, використати рекурсію.

Програма №6 Self Practice Work (Зуби)

* Планований час на реалізацію 90 хв
* Важливі деталі для врахування в імплементації
* Потрібно порахувати кількість добре загострених зубів, котрі стоять поспіль, та визначити найбільшу з них.

Програма №7 Self Practice Work (Algotester lab1v1)

* Планований час на реалізацію 40 хв
* Важливі деталі для врахування в імплементації
* Врахувати, що або мани, або хітпоінти мають бути більші за нуль, для перемоги.

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

Деталі по конфігурації середовища

Достатньо базових розширень та налаштувань Visual Studio Code, C++, нічого додаткового не потрібно для кожного завдання.

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

Завдання №1 VNS Lab 4

Деталі по програмі

Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел, роздрукувати отриманий масив, знищити всі елементи із заданим значенням, додати перед кожним парним елементом масиву елемент зі значенням 0, роздрукувати отриманий масив.

# #include <iostream>

#include <cstdlib>

using namespace std;

int main()

{

int N = 100;

int a[N];

int length;

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

cin >> length;

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

{

a[i] = rand()%10;

}

cout << "Received array: ";

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

{

cout << a[i] << " ";

}

cout << endl;

int delete\_number;

cout << "Which elements do you want to delete? (Enter one number)" << endl;

cin >> delete\_number;

int index\_delete\_number = -1;

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

{

if(a[i] == delete\_number)

{

--N;

--length;

index\_delete\_number = i;

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

{

a[j] = a[j + 1];

}

}

}

if(index\_delete\_number == -1)

{

cout << "There wasnn't such an element in this array" << endl;

}

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

{

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

{

++N;

++length;

for(int j = length - 1; j > i; j--)

{

a[j] = a[j - 1];

}

a[i] = 0;

++i;

}

}

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

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

{

cout << a[i] << " ";

}

cout << endl;

return 0;

}

Блок з кодом до завдання №1

Завдання №2 VNS Lab 5

Деталі по програмі

Функція транспонує квадратну матрицю, з її допомогою програма визначає чи є задана матриця симетричною.

# #include <iostream>

using namespace std;

const int m = 4;

const int n = 4;

void transposition (int matrix[m][n])

{

int new\_matrix[n][m];

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

{

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

{

new\_matrix[j][i] = matrix[i][j];

}

}

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

{

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

{

matrix[i][j] = new\_matrix[i][j];

}

}

};

bool symmetric (int matrix[m][n])

{

transposition (matrix);

if(m != n)

{

return false;

}

else

{

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

{

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

{

if(matrix[i][j] != matrix[j][i])

{

return false;

}

}

}

}

return true;

};

int main()

{

int matrix [m][n];

cout << "Enter your " << m << "x" << n << " matrix: " << endl;

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

{

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

{

cin >> matrix[i][j];

}

}

transposition(matrix);

cout << "New matrix: " << endl;

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

{

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

{

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

}

cout << endl;

}

if(symmetric(matrix) == true)

{

cout << "The matrix is symmetric" << endl;

}

else

{

cout << "The matrix is not symmetric" << endl;

}

return 0;

}

Блок з кодом до завдання №2

Завдання №3 Algotester Lab 2

Деталі по програмі

Програма виводить мінімальну втому, яку можна отримати.

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

int N, m;

int main()

{

cin >> N;

if (N < 1)

{

return 0;

}

else if (N==1)

{

cout << 0 << endl;

return 0;

}

vector <int> r;

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

{

int number;

cin >> number;

r.push\_back(number);

}

sort(r.begin(), r.end());

if (r[N-1] - r[1] > r[N-2] - r[0])

{

m = r [N-2] - r[0];

}

else

{

m = r[N-1] - r[1];

}

cout << m << endl;

return 0;

}

Блок з кодом до завдання №3

Завдання №4 Algotester Lab 3

Деталі по програмі

Програма виводить компресовану стрічку.

#include <iostream>

#include <string>

using namespace std;

int main()

{

string s, s\_compressed;

cin >> s;

int count = 1;

for (int i = 0; i < s.length(); i++)

{

if (s[i] == s[i + 1])

{

count++;

}

else

{

s\_compressed += s[i];

if (count > 1)

{

s\_compressed += to\_string(count);

}

count = 1;

}

}

cout << s\_compressed << endl;

return 0;

}

Блок з кодом до завдання №4

Завдання №5 Class Practice Work

Деталі по програмі

Програма перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.

#include <iostream>

using namespace std;

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

{

if(start >= end)

{

return true;

}

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

{

return(isPalindrome(str, start + 1, end - 1));

}

else

{

return false;

}

return true;

};

bool isPalindrome(int number)

{

string str = to\_string(number);

int start = 0;

int end = str.length() - 1;

return isPalindrome(str, start, end);

};

int main()

{

string str;

int number;

cout << "Enter your word: ";

cin >> str;

cout << "Enter your number: ";

cin >> number;

int start = 0;

int end = str.length() - 1;

if(isPalindrome(str, start, end) == true)

{

cout << "This is a palindrome" << endl;

}

else

{

cout << "This is not a palindrome" << endl;

}

if(isPalindrome(number) == true)

{

cout << "This is a palindrome" << endl;

}

else

{

cout << "This is not a palindrome" << endl;

}

return 0;

}

Блок з кодом до завдання №5

Завдання №6 Self Practice Work (Зуби)

Деталі по програмі

Програма виводить максимальну кількість добре загострених зубів, котрі ідуть підряд.

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

long long k;

int n, counter;

vector <long long> a;

vector <int> c;

int main()

{

cin >> n >> k;

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

{

long long ai;

cin >> ai;

a.push\_back(ai);

}

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

{

if (a[j]>=k)

{

counter++;

}

else

{

counter = 0;

}

c.push\_back(counter);

}

sort(c.begin(), c.end());

cout << c[n-1] << endl;

return 0;

}

Блок з кодом до завдання №6

Завдання №7 Self Practice Work (Algotester lab1v1)

Деталі по програмі

* Програма виводить YES - якщо ваш персонаж виграє
* NO - у всіх інших випадках

#include <iostream>

using namespace std;

long long H, M, h1, h2, h3, m1, m2 ,m3;

int main()

{

cin >> H >> M >> h1 >> m1 >> h2 >> m2 >> h3 >> m3;

if (H < 1 || M < 1)

{

return 0;

}

long long h[3] = {h1, h2, h3};

long long m[3] = {m1, m2, m3};

for(int i = 0; i < 3; i++)

{

if(h[i] > 0 && m[i] > 0)

{

cout << "NO" << endl;

return 0;

}

H -= h[i];

M -= m[i];

}

if (H > 0 && M > 0)

{

cout << "YES" << endl;

}

else

{

cout << "NO" << endl;

}

return 0;

}

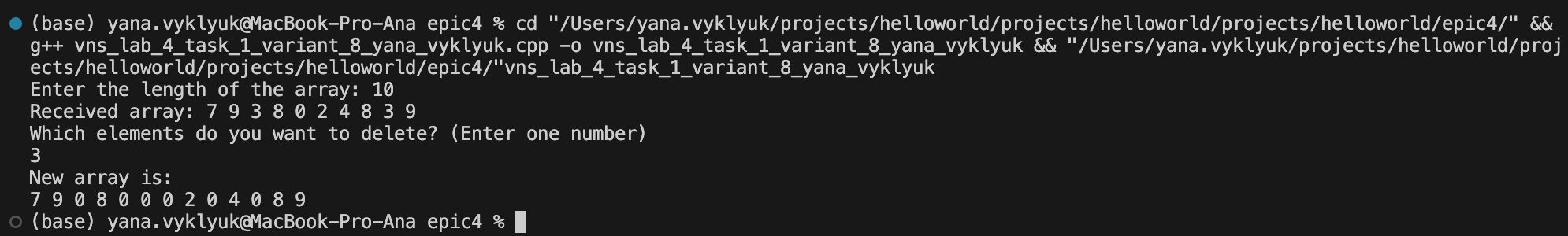
Блок з кодом до завдання №7

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

Завдання №1 VNS Lab 4

Деталі по виконанню і тестуванню програми

При виконанні цієї програми спочатку користувач задає кількість елементів з якою він хоче працювати, тоді отримує самі елементи, обирає котре число хоче повністю видалити, тоді програма видаляє ті числа, зменшуючи реальну довжину масиву, та перед парними числами ставить 0,збільшуючи реальну довжину масиву.



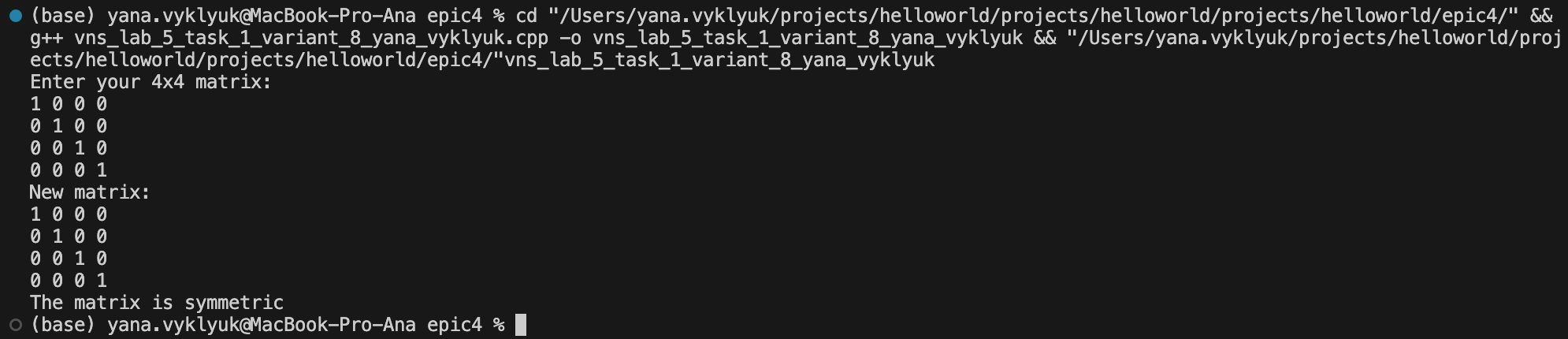
Блок з виконанням та тестуванням програми до завдання №1

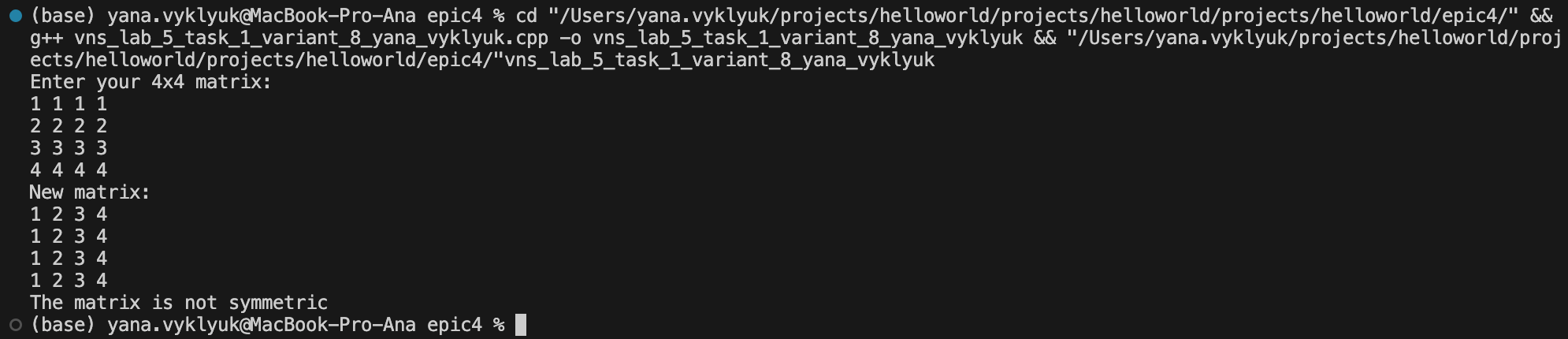
Час затрачений на виконання завдання 60 хв.

Завдання №2 VNS Lab 5

Деталі по виконанню і тестуванню програми

Програма за допомогою функції транспонує введену матрицю, виводить транспоновану, іншою функцією отримує значення true або false, в залежності від нього програма показує чи матриця симетрична чи ні..





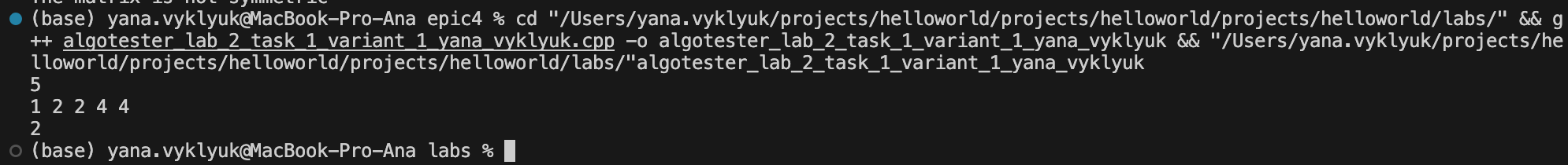
Блок з виконанням та тестуванням програми до завдання №2

Час затрачений на виконання завдання 90 хв

Завдання №3 Algotester Lab 2

Деталі по виконанню і тестуванню програми

Програма виводить мінімальну втому, в залежності, яка різниця менша, при відкиданні мінімального елемента чи при відкиданні максимального..



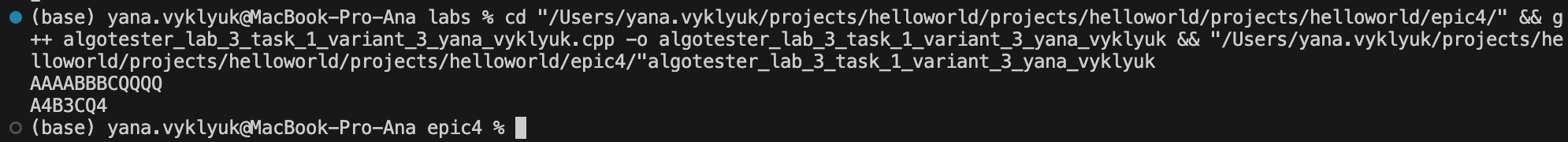
Блок з виконанням та тестуванням програми до завдання №3

Час затрачений на виконання завдання 90 хв

Завдання №4 Algotester Lab 3

Деталі по виконанню і тестуванню програми

Програма виводить компресовану стрічку.



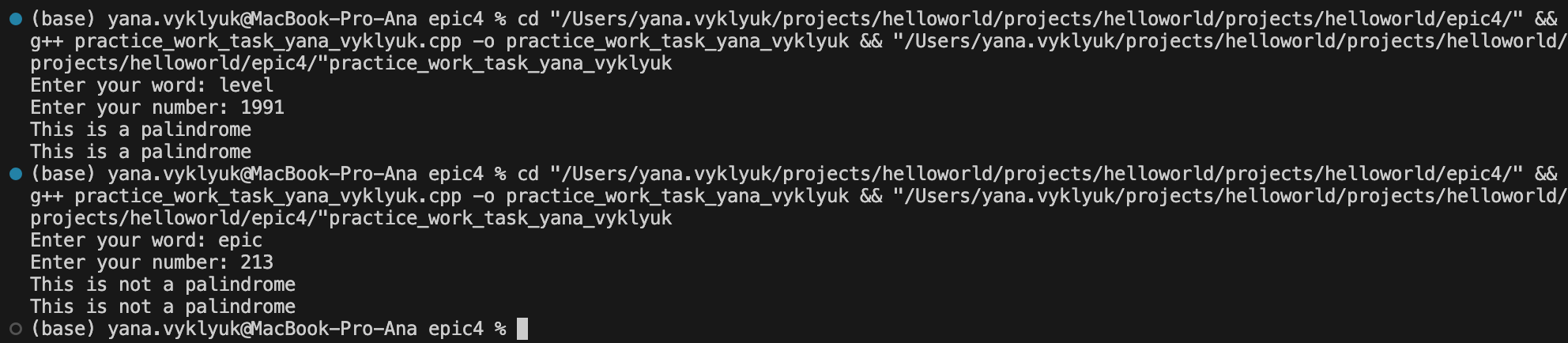
Блок з виконанням та тестуванням програми до завдання №4

Час затрачений на виконання завдання 90 хв.

Завдання №5 Class Practice Work

Деталі по виконанню і тестуванню програми

Програма перевіряє чи слово або число є паліндромом, при цьому використовується рекурсія та перевантаження.



Блок з виконанням та тестуванням програми до завдання №5

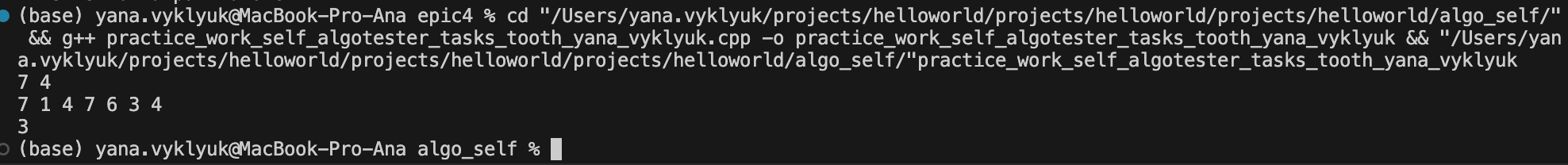
Час затрачений на виконання завдання 45 хв.

Завдання №6 Self Practice Work (Зуби)

Деталі по виконанню і тестуванню програми

Код програми загружено на Algotester. Програма виводить максимальну кількість добре загострених зубів, котрі ідуть підряд.





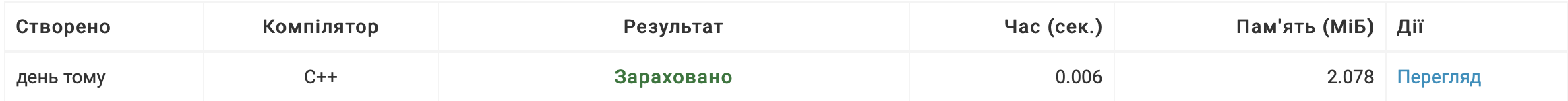
Блок з виконанням та тестуванням програми до завдання №6

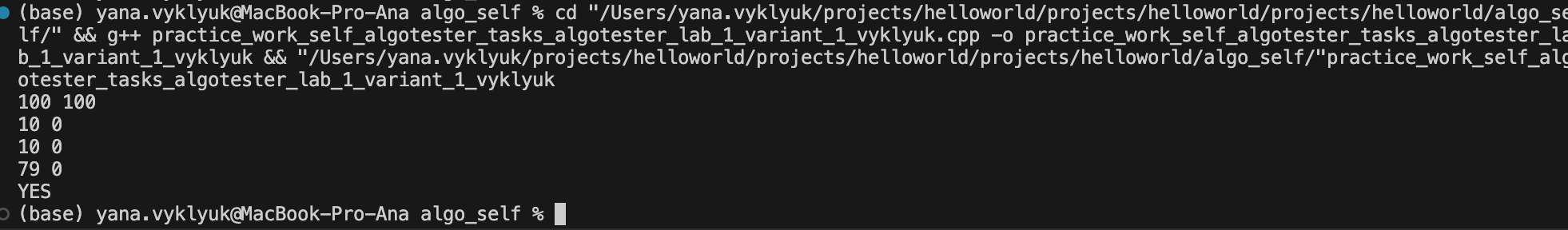
Час затрачений на виконання завдання 60 хв.

Завдання №7 Self Practice Work (Algotester lab1v1)

Деталі по виконанню і тестуванню програми

* Програма виводить YES - якщо ваш персонаж виграє
* NO - у всіх інших випадках

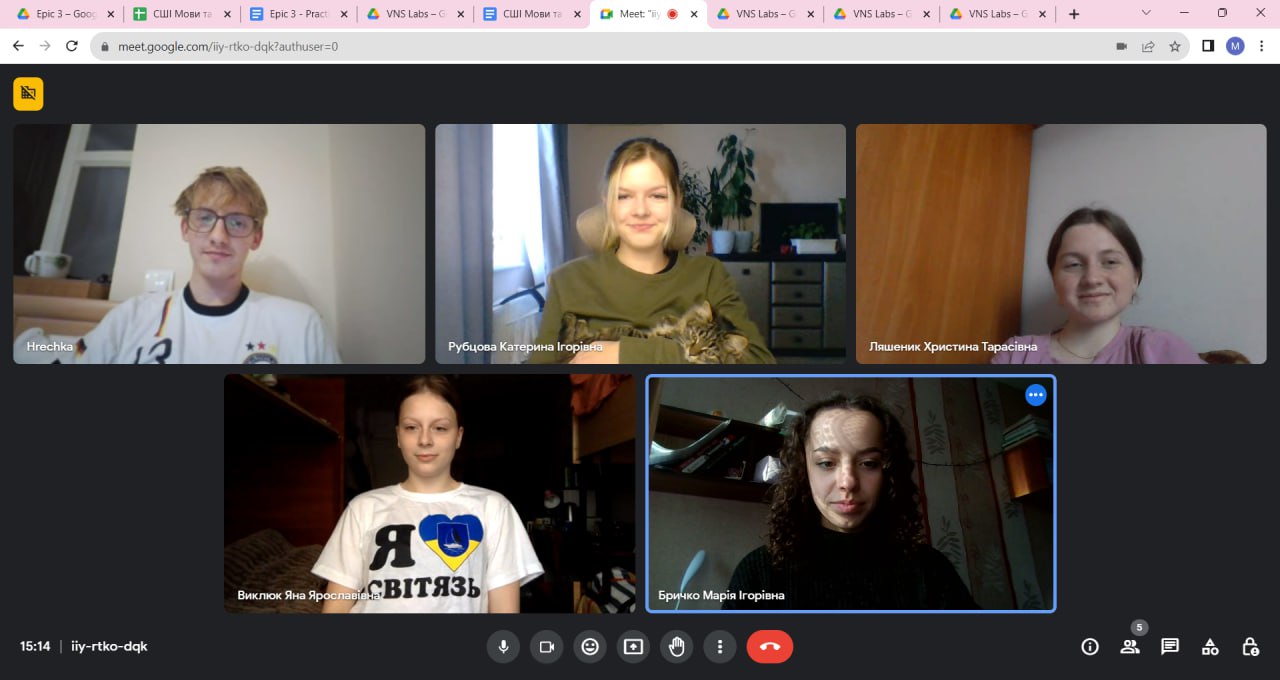


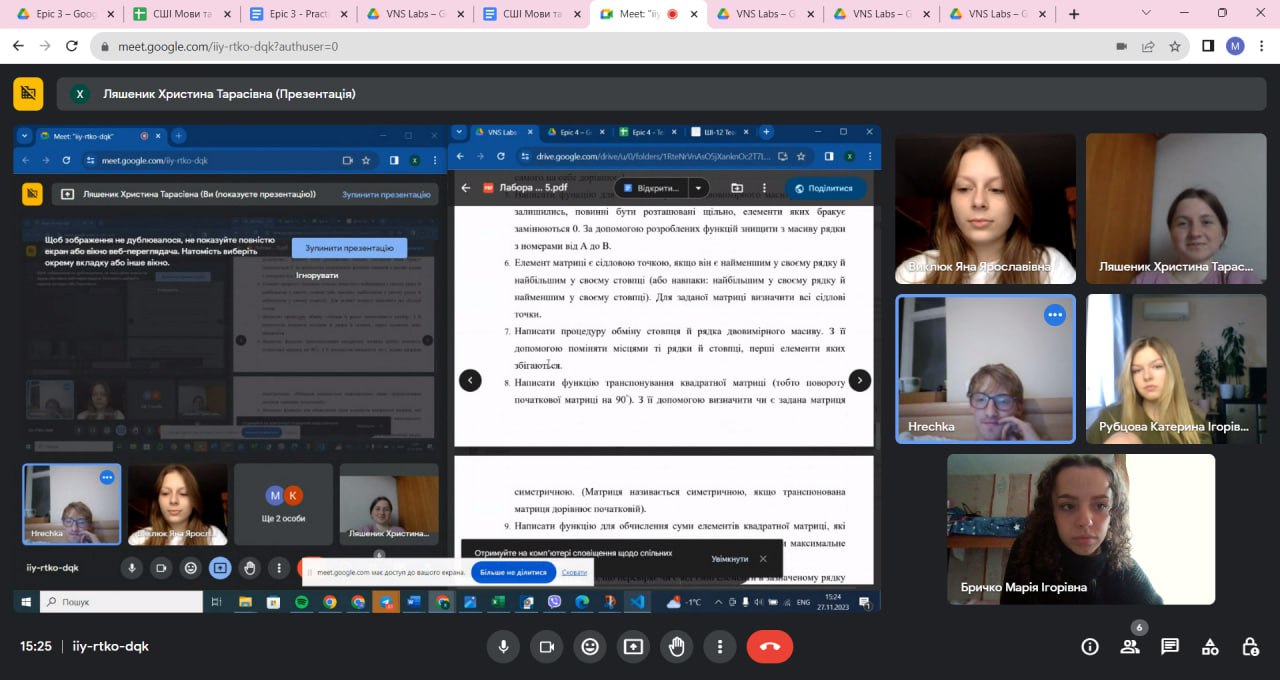


Блок з виконанням та тестуванням програми до завдання №7

Час затрачений на виконання завдання 40 хв.

## **6. Командна робота(мітинги, постановка задач та їх обговорення):**





# **Висновки:**

Ознайомилась працювати з простими структурами даних. Ознайомилась масивами та їх видами. Працювала з одновимірними та двовимірними масивами. Ознайомилась та працювала з векторами. Ознайомилась з алгоритмами обробки.