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***: «Просто структури даних. Одновимірні масиви. Двовимірні масиви. Алгоритми обробки.»

***Виконав:***

студент групи ШІ-14

Іжелюк Назарій Миколайович

# **Тема роботи:**

Вивчення та Застосування Одновимірних та Двовимірних Масивів у Контексті Простих Структур Даних та Алгоритмів Обробки.

# **Мета роботи:**

Метою даної лабораторної роботи є ознайомлення з основними концепціями одновимірних та двовимірних масивів як простих структур даних у програмуванні. Робота спрямована на вивчення та розуміння принципів їх використання для зберігання та обробки інформації.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Ознайомлення із простими структурами даних.
* Тема №2: Ознайомлення із поняттям статичного, динамічного, одновимірного та двовимірного масиву.
* Тема №3: Алгоритми обробки масивів.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Ознайомлення із простими структурами даних.
  + Джерела Інформації
    - Стаття: <https://acode.com.ua/urok-78-fiksovani-masyvy/>
    - Стаття: <http://cpp.dp.ua/vykorystannya-masyviv/>
  + Що опрацьовано:
    - Опрацьовано структури даних їх типи та відмінності.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 13.11.2023
  + Звершення опрацювання теми: 17.11.2023
* Тема №2: Ознайомлення із поняттям статичного, динамічного, одновимірного та двовимірного масиву.
  + Джерела Інформації:
    - Відео: <http://surl.li/oadox>
    - Відео: <http://surl.li/oadpk>
    - Стаття: <https://acode.com.ua/urok-77-masyvy/>
  + Що опрацьовано:
    - Опрацьовано поняття та синтаксис статичного масиву.
    - Опрацьовано поняття та синтаксис динамічного масиву.
    - Опрацьовано поняття та синтаксис одновимірного та двовимірного масиву.
    - Опрацьовано варіанти застосування цих масивів.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 13.11.2023
  + Звершення опрацювання теми: 17.11.2023
* Тема №3: Алгоритми обробки масивів.
  + Джерела Інформації:
    - Стаття:<https://learn.microsoft.com/en-us/cpp/standard-library/algorithm?view=msvc-170>
  + Що опрацьовано:
    - Опрацьовані основні алгоритми обробки масивів бібліотеки <algorithm>.
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 13.11.2023
  + Звершення опрацювання теми: 17.11.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 4

* Варіант: 23
* Деталі завдання:

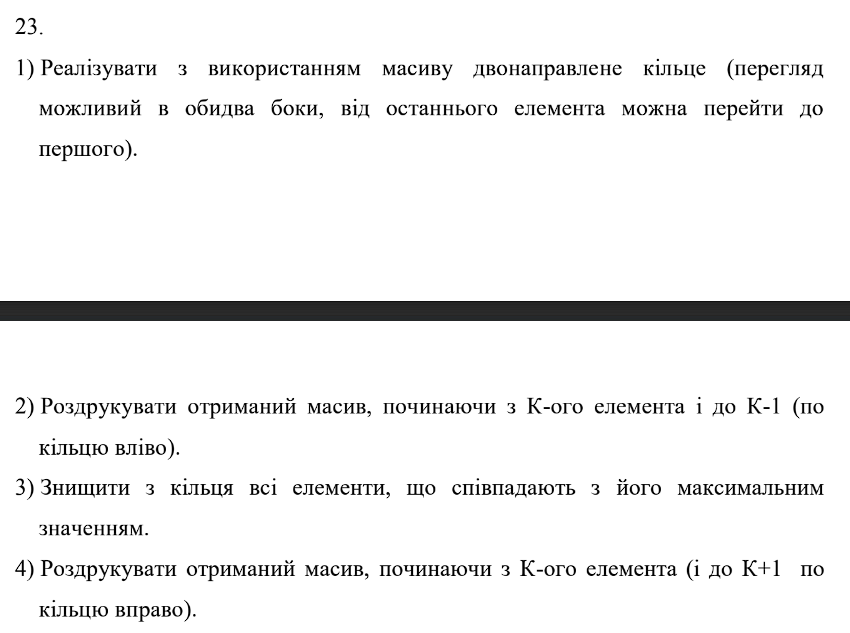


Рисунок Завдання VNS Lab 4

* Важливі деталі: Реалізувати потрібно за допомогою масиву і здійснити вивід який указано в завданні.

Завдання №2 VNS Lab 5

* Варіант: 23
* Деталі завдання:

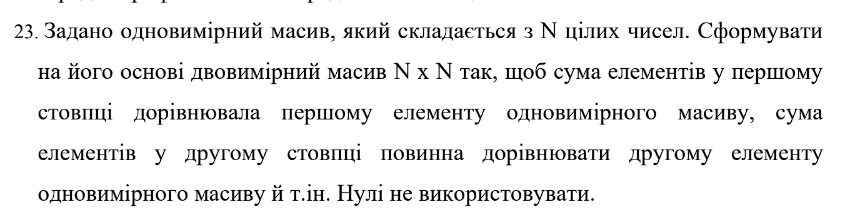


Рисунок Завдання VNS Lab 5

* Важливі деталі: Сума елементів в стовпці двовимірного масиву повинна дорівнювати відповідному елементу одновимірного масиву.

Завдання №3 Algotester Lab 2

* Варіант: 2
* Деталі завдання:

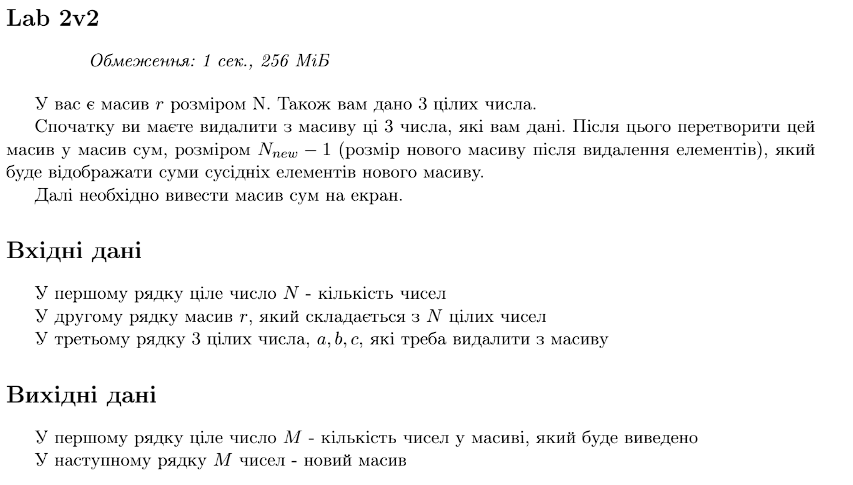


Рисунок Завдання Algotester Lab 2

* Важливі деталі: Вхідні та вихідні дані повинні бути в порядку заданим завданням.

Завдання №4 Algotester Lab 3

* Варіант: 2
* Деталі завдання:

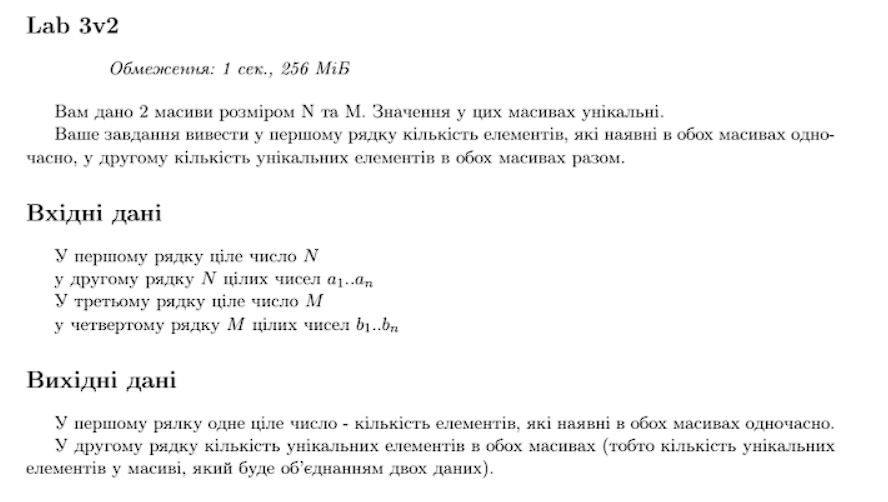


Рисунок Завдання Algotester Lab 3

* Важливі деталі: Вхідні та вихідні дані повинні бути в порядку заданим завданням.

Завдання №5 Class Practice Work

* Варіант: 2
* Деталі завдання:

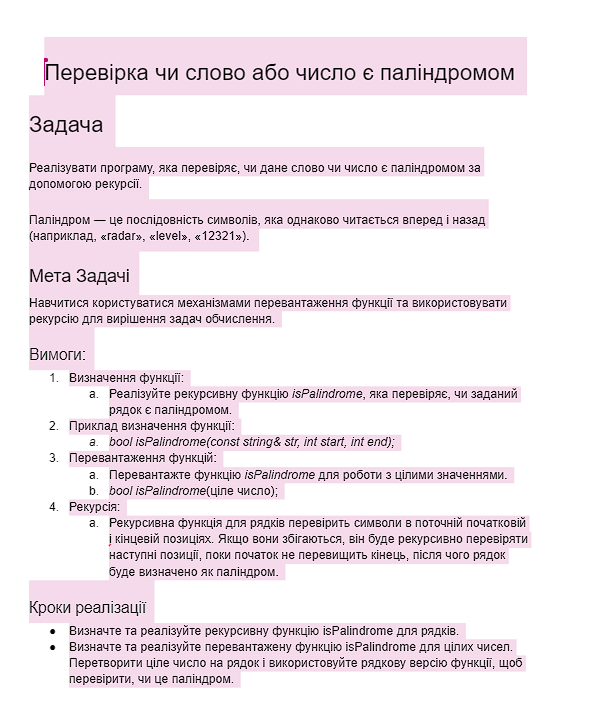


Рисунок Завдання Class Practice Work

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

Завдання №6 Self Practice Work

* Варіант: Верховна Рада
* Деталі завдання:

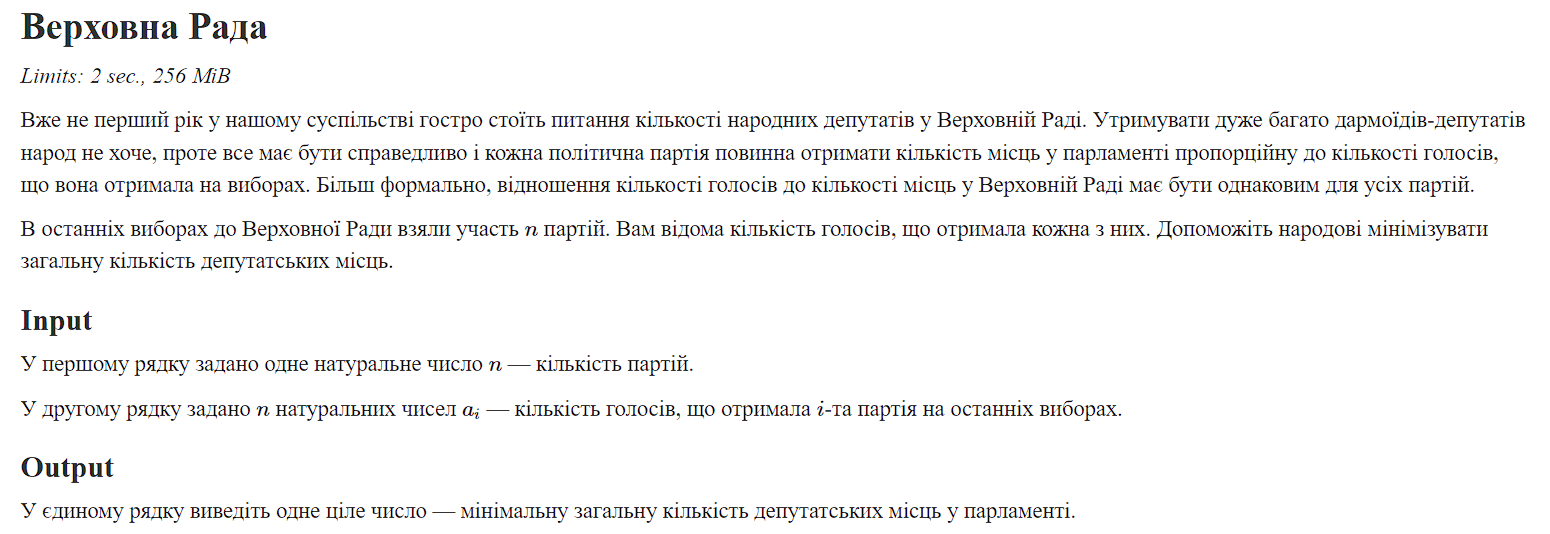


Рисунок Завдання Sеlf Practice Work

* Важливі деталі: Вхідні та вихідні дані повинні бути в порядку заданим завданням.

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

Програма №1 VNS Lab 4

* Блок-схема: Не робив
* Планований час на реалізацію: 1 год.
* Важливі деталі для врахування в імплементації: Дотримуватись умови завдання

Програма №2 VNS Lab 5

* Блок-схема: Не робив
* Планований час на реалізацію: 1 год.
* Важливі деталі для врахування в імплементації: Потрібно щоб суму елементів стовпця двовимірного масиву дорівнювала відповідному елементу одновимірного масиву.

Програма №3 Algotester Lab 2

* Блок-схема: Не робив
* Планований час на реалізацію: 1 год.
* Важливі деталі для врахування в імплементації: Потрібна перевірка на платформі algotester.

Програма №4 Algotester Lab 3

* Блок-схема: Не робив
* Планований час на реалізацію: 1 год.
* Важливі деталі для врахування в імплементації: Потрібна перевірка на платформі algotester.

Програма №5 Class Practice Work

* Блок-схема: Не робив
* Планований час на реалізацію: 1 год.
* Важливі деталі для врахування в імплементації: Потрібно використати рекурсію для перевірки чи слово є паліндромом.

Програма №6 Self Practice Work

* Блок-схема:

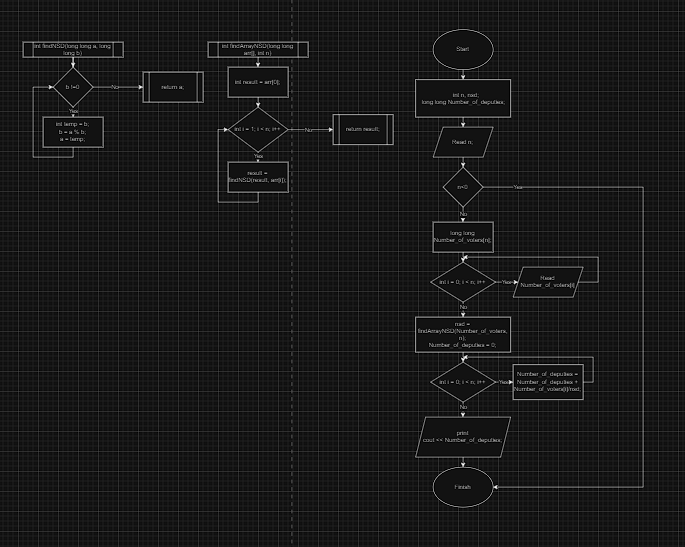


Рисунок Блоксзема до завдання Sеlf Practice Work

* Планований час на реалізацію: 1 год.
* Важливі деталі для врахування в імплементації: Потрібна перевірка на платформі algotester.

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

Конфігурацію середовища не змінювали.

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

Завдання №1

Деталі по програмі: За допомогою масиву реалізовано кільце та його вивід за умовою завдання.

#include<iostream>

using namespace std;

int main(){

int size, k, size2;

cout << "Enter the ring size: ";

cin >> size;

int ring[size];

cout << "Enter the elements of the ring: ";

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

cin >> ring[i];

}

cout << "Ring:";

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

cout << " [ " << ring[i] << " ] ";

}

//друк вліво

cout << endl << "Enter k: ";

cin >> k;

cout << "Left from k";

for(int i = k; i >= 0; i--){

cout << " [ " << ring[i] << " ] ";

}

for(int i = size-1; i >= k+1 ; i--){

cout << " [ " << ring[i] << " ] ";

}

cout << endl;

//видалення

size2 = 0;

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

if (ring[i] != size) {

ring[size2++] = ring[i];

}

}

cout << "Ring after removing elements: ";

for (int i = 0; i < size2; ++i) {

cout << " [ " << ring[i] << " ] ";

}

cout << endl;

//друк вправо

cout << "ring right from k: ";

for(int i = k; i < size2; i++){

cout << " [ " << ring[i] << " ] ";

}

for(int i = 0; i < k; i++){

cout << " [ " << ring[i] << " ] ";

}

return 0;

}

Завдання №2

Деталі по програмі: Реалізовано програму яка заповнює двовимірний масив спираючись на значення одновимірного масиву.

#include<iostream>

using namespace std;

int main(){

int N, i, last\_element;

cout << "Enter size of aray: " ;

cin >> N;

int aray1[N];

int aray2[N][N];

cout << "Enter elements of aray: " << endl;

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

cin >> aray1[i] ;

if(aray1[i] < N){

cout << "Element must be > N";

exit(0);

}

}

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

int fill = aray1[i];

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

fill = fill / 2;

aray2[j][i] = fill;

}

}

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

aray2[N-1][i] = 0;

}

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

cout << "[ " << aray2[N-1][i] << " ] ";

}

cout << endl;

cout << endl;

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

last\_element = aray1[i];

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

last\_element = last\_element - aray2[j][i];

}

aray2[N-1][i] = last\_element;

cout << "[ " << last\_element << " ]" << endl;

}

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

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

cout << " [ " << aray2[i][j] << " ] ";

}

cout << endl;

}

return 0;

}

Завдання №3

Деталі по програмі: Створено програму яка запитує розмір масиву та його елементи та 3 числа які необхідно видалити масиву і після видалення елементів виведено масиву сум елементів.

#include <iostream>

using namespace std;

int main(){

long long N, a, b, c;

cin >> N;

long long r[N];

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

cin >> r[i];

}

cin >> a >> b >> c;

int n = 0;

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

if(r[i] != a && r[i] != b && r[i] != c){

r[n++] = r[i];

}

}

if(n-1 < 0){

cout << 0;

exit(0);

}

cout << n-1 << endl;

long long M[n-1];

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

M[i] = r[i] + r[i+1];

}

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

cout << M[i] << " ";

}

return 0;

}

Завдання №4

Деталі по програмі: Потрібно ввести розмір та елементи 2 масивів і отримати результат кiлькiсть елементiв, якi наявнi в обох масивах одночасно, у другому кiлькiсть унiкальних елементiв в обох массивах разом.

#include<iostream>

using namespace std;

int main(){

int N, M;

cin >> N;

int arreaN[N];

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

cin >> arreaN[i];

}

cin >> M;

int arreaM[M];

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

cin >> arreaM[i];

}

int count = 0;

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

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

if(arreaN[i] == (arreaM[j])){

count = count + 1;

}

}

}

cout << count << endl;

cout << N + M - count;

return 0;

}

Завдання №5

Деталі по програмі: Потрібно ввести слово або число і програма виведе чи є слово паліндромом.

#include <iostream>

#include <string>

using namespace std;

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

if(start >= end)

return true;

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

return false;

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

}

bool isPalindrome(string str) {

int n = str.length();

return isPalindrome(str, 0, n-1);

}

int main() {

string input;

cout << "Введіть слово: ";

getline(cin, input);

if(isPalindrome(input)) {

cout << input << ": паліндром" << endl;

} else {

cout << input << ": не паліндром" << endl;

}

int number;

cout << "Введіть число: ";

cin >> number;

string numStr = to\_string(number);

if(isPalindrome(numStr)) {

cout << number << ": паліндром" << endl;

} else {

cout << number << ": не паліндром" << endl;

}

return 0;

}

Завдання №6

Деталі по програмі: Програма за допомогою функцій findNSD і findArrayNS вираховує середнє арифметичне масиву чисел в який записані кількість голосів.

Посилання на задачу: <https://algotester.com/en/ArchiveProblem/DisplayWithEditor/7>

#include<iostream>

using namespace std;

int findNSD(long long a, long long b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

int findArrayNSD(long long arr[], int n) {

int result = arr[0];

for (int i = 1; i < n; i++) {

result = findNSD(result, arr[i]);

}

return result;

}

int main() {

int n, nsd;

long long Number\_of\_deputies;

cin >> n;

if(n < 0){

exit(0);

}

long long Number\_of\_voters[n];

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

cin >> Number\_of\_voters[i];

}

nsd = findArrayNSD(Number\_of\_voters, n);

Number\_of\_deputies = 0;

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

Number\_of\_deputies = Number\_of\_deputies + Number\_of\_voters[i]/nsd;

}

cout << Number\_of\_deputies;

return 0;

}

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

Завдання №1

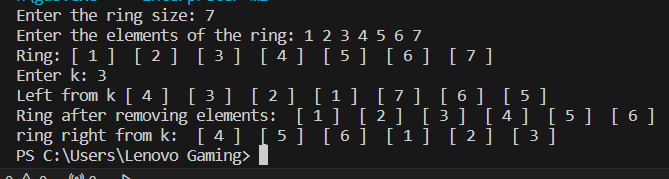
Деталі по виконанню і тестуванню програми: Спочатку введено розмір кільця та його елементи і число k з якого його виводити.

Рисунок Тестування VNS Lab 4

Час затрачений на виконання завдання 1 год.

Завдання №2

Деталі по виконанню і тестуванню програми: Ведено розмір масиву та його елементи і виведено двовимірний масив так щоб сума елементів стовпця дорівнювала відповідному елементу 1 масиву.

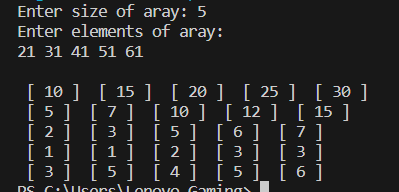


Рисунок Тестування VNS Lab 5

Завдання №3

Деталі по виконанню і тестуванню програми: Програму протестовано на платформі algotester.

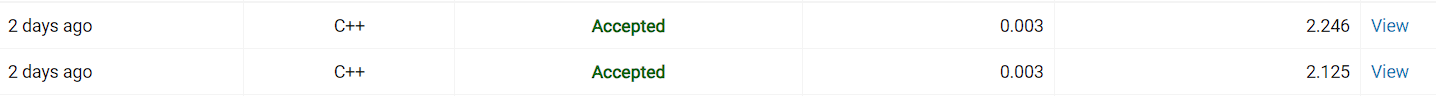


Рисунок Тестування Algotester Lab 2

Завдання №4

Деталі по виконанню і тестуванню програми: Програму протестовано на платформі algotester.

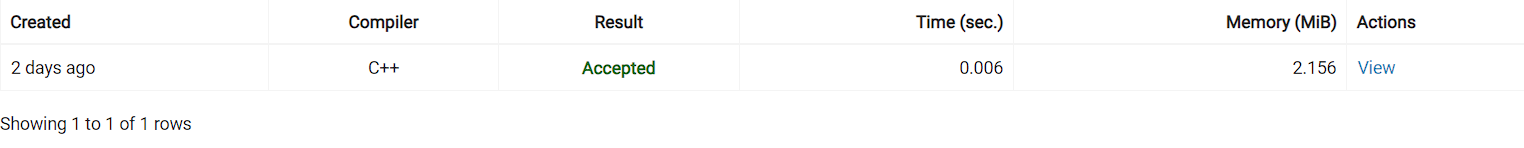


Рисунок Тестування Algotester Lab 3

Час затрачений на виконання завдання 1 год.

Завдання №5

Деталі по виконанню і тестуванню програми: Спочатку введено слово і число і виведено чи являються вони паліндромом .

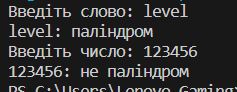


Рисунок Тестування Class Practice Work

Час затрачений на виконання завдання 1 год.

Завдання №4

Деталі по виконанню і тестуванню програми: Програму протестовано на платформі algotester.

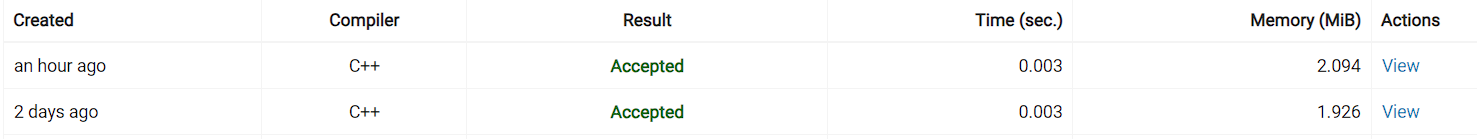


Рисунок Тестування Sеlf Practice Work

Час затрачений на виконання завдання 1 год.

# **Висновки:**

Ознайомився із простими структурами даних, різними типами масивів та на практиці засвоїв їх використання під час виконання задач. Також ознайомився із основними алгоритмами обробки масивів.