xМіністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 4,5**

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

***з розділу***: «\_\_\_\_\_\_\_\_Прості структури даних. Одновимірні масиви. Двовимірні Масиви. Алгоритми обробки\_\_\_\_\_\_\_\_\_\_»

***Виконав:***

студент групи ШІ-13

Кібиш Тарас Кібиш

# **Тема роботи:**

Знайомство з масивами та їхніми видами. Використання простих операцій в роботі з масивами.

# **Мета роботи:**

Дізнатися про нові можливості в роботі з с++, а саме вивчити, що таке масив, динамічний масив і двовимірний масив, а також навчитися використовувати їх на практиці.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Одновимірні масиви
* Тема №2:Основі алгоритми і принцип роботи з масивами
* Тема №3: Динамічні масиви
* Тема №4: Основні команди бібліотеки “vector”
* Тема №5: Двовимірні масиви

1. Індивідуальний план опрацювання теорії:

* Тема №1: Одновимірні масиви.
  + Джерела Інформації
    - <https://www.geeksforgeeks.org/cpp-arrays/>
    - <https://www.youtube.com/watch?v=ULdbOaMBPYc&t=14s&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>
  + Що опрацьовано:
    - Здобуто базове розуміння, що таке масив
    - Досліджено спосіб використання масивів в мові програмування C++
  + Статус: Ознайомлений
  + Початок опрацювання теми: 27.11.2023
  + Звершення опрацювання теми: 28.11.2023
* Тема №2: Основі алгоритми і принцип роботи з масивами
  + Джерела Інформації:
    - <https://chat.openai.com/>
    - <https://unstop.com/blog/array-in-cpp>
  + Що опрацьовано:
    - Вивчено основні методи роботи з масивами
  + Статус: Ознайомлений
  + Початок опрацювання теми: 27.11.2023
  + Звершення опрацювання теми: 28.11.2023
* Тема №3: Динамічні масиви
  + Джерела Інформації:
    - <https://www.youtube.com/watch?v=RXzzE2wnnlo&ab_channel=TechWithTim>
    - <https://chat.openai.com/>
  + Що опрацьовано:
    - Здобуто базове розуміння, що таке динамічний масив
    - Коментар 2
  + Статус: Ознайомлений/ Ознайомлений частково / Не ознайомлений
  + Початок опрацювання теми: Дата
  + Звершення опрацювання теми: Дата
* Тема №4 Основні команди бібліотеки “vector”
  + Джерела Інформації:
    - <https://chat.openai.com/>
  + Що опрацьовано:
    - Ознайомився з роботою динамічних масивів
    - Дізнався про головні команди бібліотеки “vector”
  + Статус: Ознайомлений
  + Початок опрацювання теми: 28.11.2023
  + Звершення опрацювання теми: 30.11.2023
* Тема №5 Двовимірні масиви
  + Джерела Інформації:
    - <https://www.geeksforgeeks.org/cpp-arrays/>
    - <https://www.youtube.com/watch?v=V2g3B9Zbh4Q&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>
  + Що опрацьовано:
    - Здобуті знання про двовимірні масиви
  + Статус: Ознайомлений
  + Початок опрацювання теми: 29.11.2023
  + Звершення опрацювання теми: 30.11.2023

# **Виконання роботи:**

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

Завдання №1 Lab# programming: VNS Lab 4

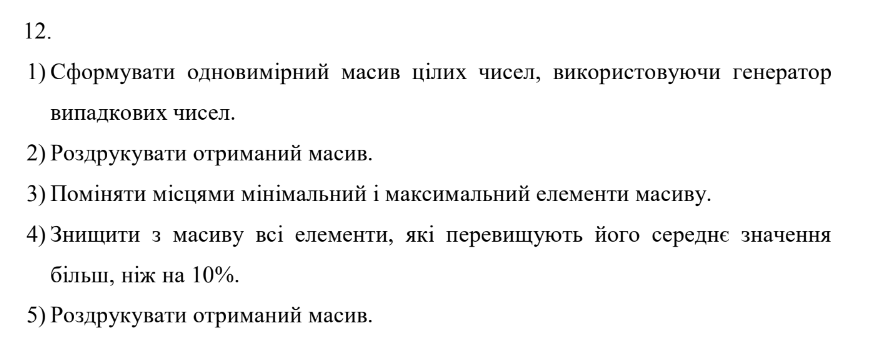
* Варіант 12
* Деталі завдання
* 

Рисунок 1: Lab# programming: VNS Lab 4

* Важливі деталі для врахування в імплементації програми

Завдання №2 Lab# programming: VNS Lab 5

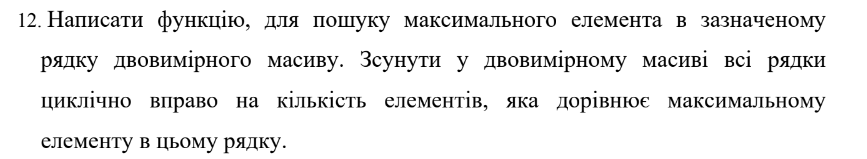
* Варіант завдання 12
* Деталі завдання
* 

Рисунок 2:Lab# programming: VNS Lab 5

* Важливі деталі для врахування в імплементації програми

Завдання №3  Lab# programming: Algotester Lab 2

* Варіант завдання 3
* Деталі завдання
* 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 11 перелетить у клітинку з індексом a1, тобто його наступна позиція рахується як поточна позиція + число у поточній позиції (перегляньте пояснення для візуалізації) 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 ai та ai+1+1 - виведіть **Stopped**
* Врахуйте, що перевіряти треба також до взльоту.
* Важливі деталі для врахування в імплементації програми
* 1≤N≤1000
* 1≤ai≤5
* Гарантується, що дрони не можуть вилетіти за межі [1,N] (тобто не існує масиву, в якому дрони вилітають за межі до того, як станеться одна з 3х вищезазначених ситуацій) та їх швидкість завжди більша за нуль.

Завдання №4 Lab# programming: Algotester Lab 3

* Варіант завдання 3
* Деталі завдання
* 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.
* Важливі деталі для врахування в імплементації програми

# Обмеження

1≤|S|≤105

Завдання №5 Practice# programming: Class Practice Task

* Варіант завдання
* Деталі завдання
* 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»).
* Важливі деталі для врахування в імплементації програми

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.

Завдання №6 Practice# programming:  Self Practice Task

* Варіант завдання
* Деталі завдання
* Зеник та Марічка грають у поле чудес. Спочатку Зеник пише на дошці загадане слово й закриває всі його букви. За один хід Марічка називає букву, а Зеник відкриває всі такі букви у слові.
* Вам необхідно визначити, за яку мінімальну кількість ходів Марічка зможе відкрити всі букви у слові.
* Наприклад, якщо Зеник загадав слово MAMA, то Марічка зможе його відкрити за два ходи, назвавши букви M та A.
* Важливі деталі для врахування в імплементації програми

# Обмеження

1≤|s|≤1001

s містить тільки великі латинські літери (A – Z).

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

Програма №2 :Lab# programming: VNS Lab 4

* Блок-схема
* Планований час на реалізацію 2 години
* Важливі деталі для врахування в імплементації

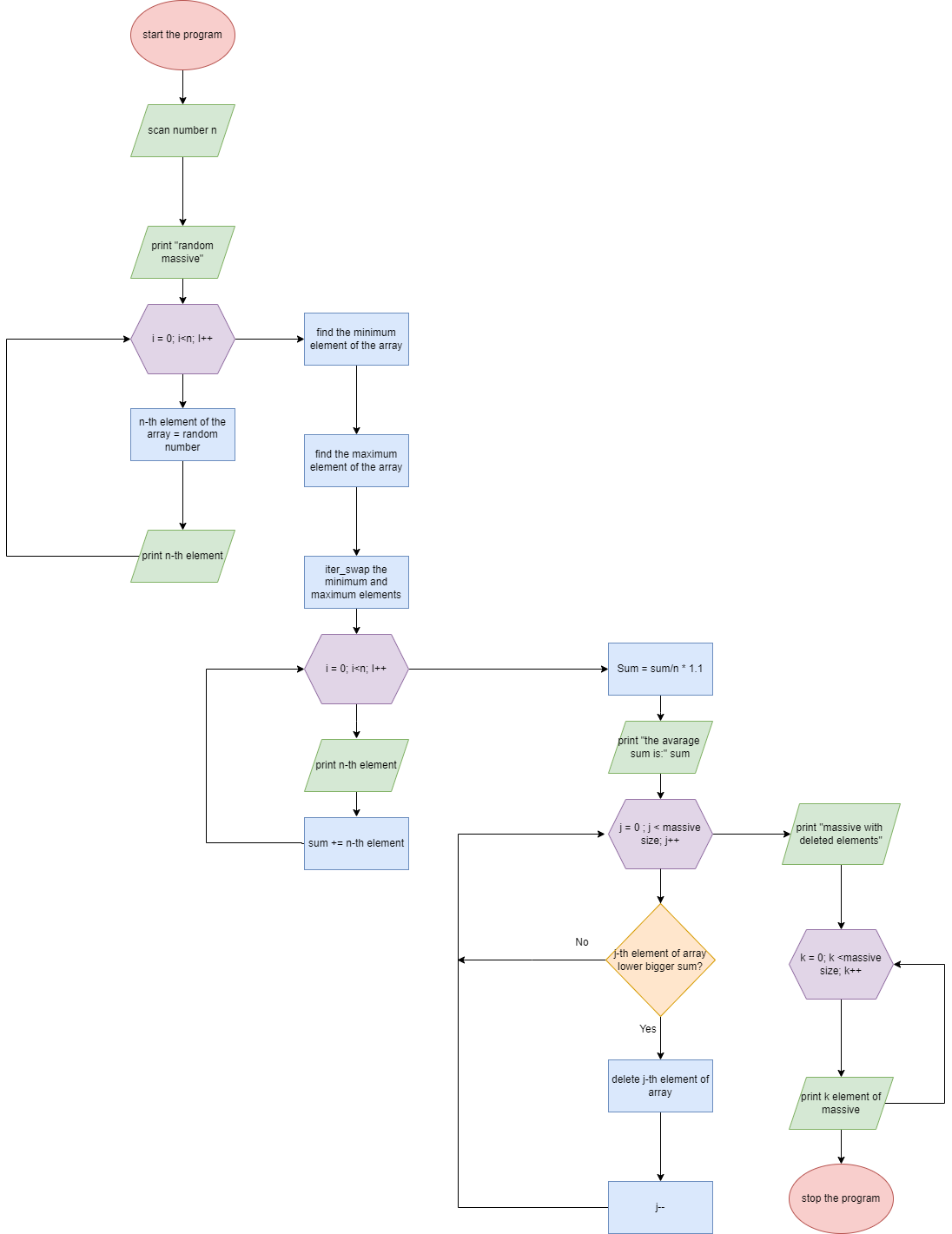


Рисунок 3:Lab# programming: VNS Lab 4

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

Завдання №\_\_ Деталі по конфігурації середовища + скріншоти з підписами до скріншотів.

Підпис та № до блоку з скріншотами до конфігурації

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

Завдання №1 Lab# programming: VNS Lab 4  
<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_taras_kibysh/ai_13/taras_kibysh/Epic%204/vns_lab_4_task_1_variant_12_taras_kibysh.cpp>

#include <iostream>

#include <vector>

#include <cstdlib>

#include <algorithm>

*int* main(){

    using *namespace* std;

*int* n;

*double* sum = 0;

    cin >> n;

    vector<*int*> massive;

    cout << "random massive:" << endl;

    for(*int* i = 0;i<n; i++){

     massive.push\_back(rand() % 100);

    cout << massive[i] << " ";

    }

    cout << endl;

*auto* max = max\_element(massive.begin(),massive.end());

*auto* min = min\_element(massive.begin(), massive.end());

    iter\_swap(max,min);

    cout << "massive with swaped min and max elements:" << endl;

    for(*int* i = 0;i<n; i++){

    cout << massive[i] << " ";

    sum += massive[i];

    }

    cout << endl;

    sum = sum/massive.size();

    cout << "the avarage number is:" << sum;

     cout << endl;

    for(*int* j = 0; j<massive.size();j++){

        if(massive[j]>sum\*1.1){

        massive.erase(massive.begin() + j);

        j--;

        }

    }

    cout << "massive with deleted elements:" << endl;

     for(*int* i = 0;i<massive.size(); i++)

    cout << massive[i] << " ";

return 0;

}

Створено програму, яка створює масив з випадкових чисел, після чого знаходить максимальний і мінімальний елемент масива і міняє їх місцями, після чого програма шукає середнє арефметичне масиву і видаляє всі елементи, які більші від середнього арифметичного на 10%.

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

Завдання №2 : Lab# programming: VNS Lab 5

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_taras_kibysh/ai_13/taras_kibysh/Epic%204/vns_lab_5_task_1_variant_12_taras_kibysh.cpp>

#include <iostream>

#include <vector>

#include <algorithm>

const *int* row = 3;

const *int* column = 3;

*void* MaxAndRotate(*int* *massive*[row][column], *int* *c*) {

*int* max = *massive*[*c*][0];

    for (*int* i = 0; i < column; i++) {

        if (max < *massive*[*c*][i])

            max = *massive*[*c*][i];

    }

    for (*int* j = 0; j < row; j++) {

*int* temp[column];

        for (*int* k = 0; k < column; k++) {

            temp[(k + max) % column] = *massive*[j][k];

        }

        for (*int* k = 0; k < column; k++) {

*massive*[j][k] = temp[k];

        }

    }

    // Print the array after rotation

    for (*int* i = 0; i < row; i++) {

        for (*int* j = 0; j < column; j++) {

            std::cout << *massive*[i][j] << " ";

        }

        std::cout << std::endl;

    }

}

*int* main() {

    using *namespace* std;

*int* answer;

*int* massive[row][column] = {

        {1, 2, 3},

        {4, 5, 5},

        {7, 8, 9}

    };

    cout << "Which row do you wanna take?: ";

    cin >> answer;

    if (answer >= 0 && answer < row) {

        MaxAndRotate(massive, answer);

    }

    return 0;

}

Створено програму, яка шукає найбільший елемент заданого рядка двовимірного масиву, а потім зсуває всі елементи масиву на це число.

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

Завдання №3:  Lab# programming: Algotester Lab 2

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_taras_kibysh/ai_13/taras_kibysh/Epic%204/algotester_lab_2_variant_3_taras_kibysh.cpp>

#include <cstdio>

#include <iostream>

*int* main(){

     using *namespace* std;

*long* *int* N;

    scanf("%i", &N);

*long* *int* drone1 = 0, drone2 = N-1;

*int* field[N];

    for(*int* i = 0;i<N; i++)

    cin>>field[i];

*long* *int* place1 = 0;

*long* *int* place2 = N-1;

drone1 = 0;

drone2 = N-1;

while(true){

    //  if(drone1>=N || drone2 < 0){

    //     return 0;

    // }

    if(drone1 == drone2 ){

    cout << drone1 + 1 << " ";

    cout << drone2 + 1 << "\n";

    cout << "Collision";

    return 0;

    }

    if(drone1>drone2){

        cout << drone1 + 1 <<" ";

        cout << drone2 + 1 << "\n";

        cout << "Miss";

        return 0;

    }

    if(drone2 == drone1+1){

       cout << drone1 + 1 << " ";

       cout << drone2 + 1 << "\n";

       cout<< "Stopped";

        return 0;

    }

     place1 = place1 + field[drone1];

    drone1 = place1;

    //розрахунки 2 дрона

    place2 = place2 - field[drone2];

    drone2 = place2;

}

return 0;

}

Створено програму, яка розраховує положення дронів за заданих умов

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

Завдання №4: Lab# programming: Algotester Lab 3  
<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_taras_kibysh/ai_13/taras_kibysh/Epic%204/algotester_lab_3_variant_3_taras_kibysh.cpp>

#include <iostream>

#include <vector>

*int* main(){

using *namespace* std;

string word;

cin >> word;

vector<*char*> letter;

vector<*long* *int*> capacity;

*int* n = 0;

*int* c = 0;

if(size(word)== 1){

cout << word;

return 0;

}

for(*int* i = 0; i+1<size(word);i++){

    letter.push\_back(word[i]);

    capacity.push\_back(0);

for(*int* j = i; j< size(word); j++){

n = j-1;

if(word[i] == word[j])

capacity[c] += 1;

else{

break;

}

}

i = n;

 if(n == size(word)-2){

    if(word[n+1] != word[n]){

    letter.push\_back(word[i+1]);

    capacity.push\_back(1);

    break;

    }

 }

c += 1;

}

for(*int* n = 0; n < size(letter);n++){

if(capacity[n] == 1)

cout << letter[n];

else{

cout << letter[n] << capacity[n];

}

}

    return 0;

}

Створено програму, яка компресує стрічку

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

Завдання №5 Practice# programming: Class Practice Task

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_taras_kibysh/ai_13/taras_kibysh/Epic%204/practice_work_task_4_taras_kibysh.cpp>

#include <iostream>

#include <string>

*bool* ispalindrom(const std::string& *str*, *int* *start*, *int* *end*){

if(*start* >= *end*)

return true;

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

return ispalindrom(*str*, *start*+1,*end*-1);

}

else{

 return false;

}

}

*bool* ispalindrom(*int* *a*){

std::string s = std::to\_string(*a*);

*int* b = s.size();

return ispalindrom(s,0,b-1);

}

*int* main(){

    using *namespace* std;

    string word;

*int* number;

cout << "Введіть слово:";

cin >> word;

*bool* answear = ispalindrom(word, 0, size(word)-1);

if(answear == true)

cout << "Word " << word << " is palindrom" << endl;

else if(answear == false)

cout << "Word " << word << " isn't palindrom" << endl;

else{

    cout << "error" << endl;

}

cout << "Введіть число:";

cin >> number;

answear = ispalindrom(number);

if(answear == true)

cout << "Number " << number << " is palindrom" <<endl;

else if(answear == false)

cout << "Number " << number << " isn't palindrom" << endl;

else{

    cout << "error" <<endl;

}

    return 0;

}

Створено функцію, яка перевіряє чи слово є паліндромом

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

Завдання №:6 Practice# programming:  Self Practice Task  
<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_taras_kibysh/ai_13/taras_kibysh/Epic%204/self_practice_work_algotester_task_4_taras_kibysh.cpp>

#include <iostream>

#include <vector>

#include <string>

*int* main(){

using *namespace* std;

*int* result = 0;

string word;

cin >> word;

vector<*char*> ans = {0};

*bool* chak = true;

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

*char* letter = word[i];

for(*int* j = 1; j < ans.size(); j++ ){

if( letter == ans[j]){

    chak = false;

    break;

}

}

if(chak == true){

result++;

ans.push\_back(letter);

}

else if(chak == false)

chak = true;

}

cout << result;

    return 0;

}

Створено програму, яка вираховує мінімальну кількість спроб для відгадання слова

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

Завдання №1: Lab# programming: VNS Lab 4

8

random massive:

41 67 34 0 69 24 78 58

massive with swaped min and max elements:

41 67 34 78 69 24 0 58

the avarage number is:46.375

massive with deleted elements:

41 34 24 0

PS C:\Users\User>

Програма вивела масив випадкових чисел, після чого поміняла місцями найбільший і найменший елемент масиву, і в кігці вивела масив без елементів які більше ніж на 10% > за середнє арефмитичне

Час затрачений на виконання завдання 5 годин

Завдання №2: Lab# programming: VNS Lab 5

Which row do you wanna take?: 1

2 3 1

5 5 4

8 9 7

Програма вивела двох вимірний масив, який зміщений на найбільший елемент(5) 2 рядка

Час затрачений на виконання завдання 3 години

Завдання 3: Lab# programming: Algotester Lab 2

6

1 1 1 1 1 1

3 4

Stopped

PS C:\Users\User>

Програма вивела фінальне положення дронів



Рисунок 4: результат на Algotester для Algotester lab 2

Час затрачений на виконання завдання 2 дня

Завдання №4: Lab# programming: Algotester Lab 3

AAAABBBBBCCCCDAAAA

A4B5C4DA4

PS C:\Users\User>

Програма зкомпресувала стрічку

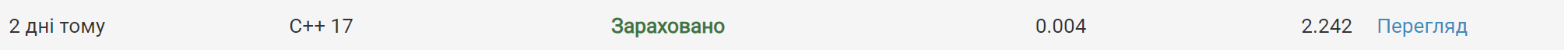


Рисунок 5:результат на Algotester для Algotester lab 3

Час затрачений на виконання завдання 4 години

Завдання №5: Practice# programming: Class Practice Task

* Варіант завдання

Введіть слово:aboba

Word aboba is palindrom

Введіть число:12341221

Number 12341221 isn't palindrom

PS C:\Users\User>

Програма перевірила чи слово і число є паліндромами

Час затрачений на виконання завдання 60 хв

Завдання №6: Practice# programming: self Practice Task

RAMADAN

5

PS C:\Users\User>

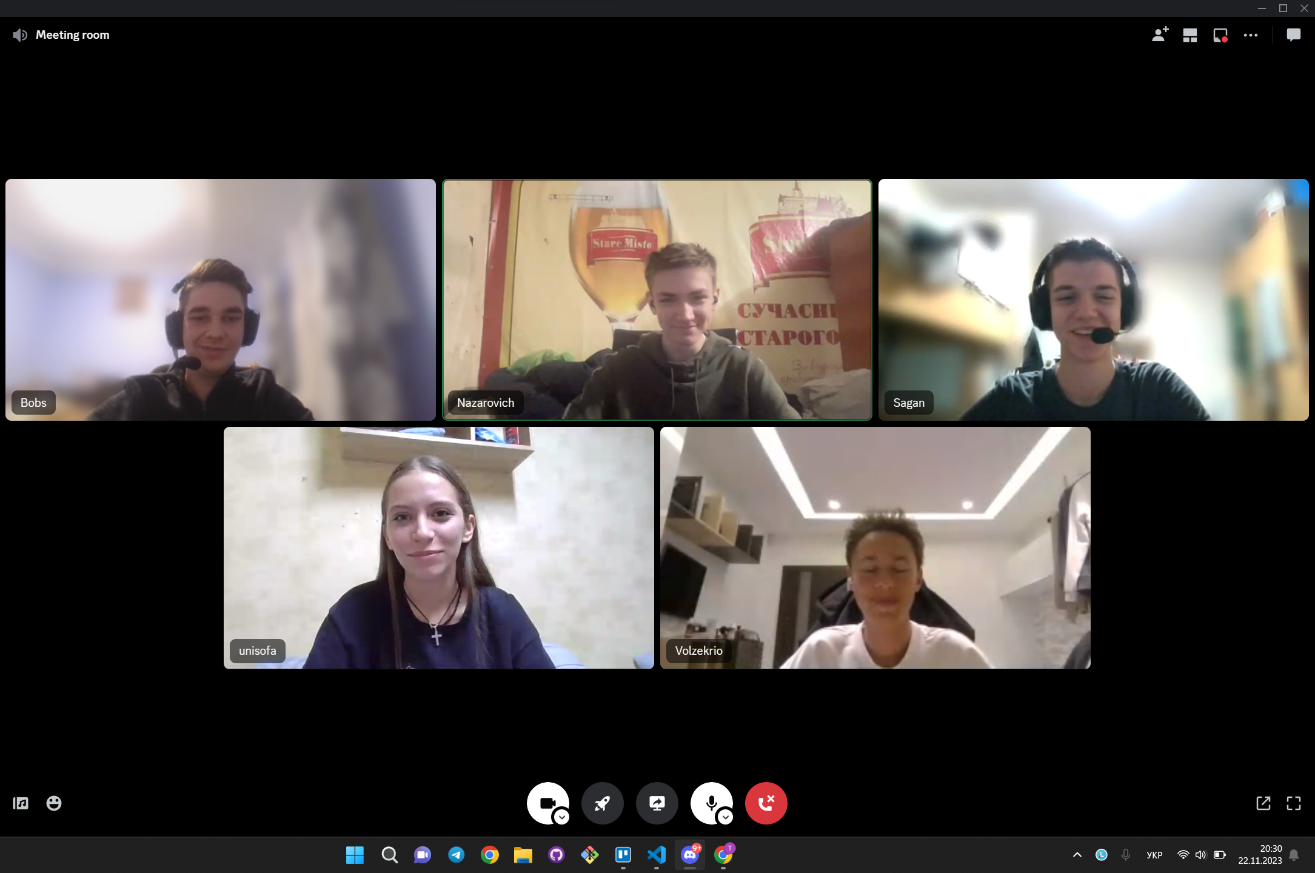
Програма порахувала скільки мінімально потрібно зробити спроб, щоб повністю відкрити слово



Рисунок 6:результат на Algotester для self practise

Час затрачений на виконання завдання 3 години

## **6. Кооперація з командою:**

* Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло
* Скрін з 2-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло
* Скрін з 3-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло (опційно)
* Скрін з 2-му коментарями від учасників команди на пул реквесті з Ревю Роботи

# **Висновки:**

Я дізнався, що таке масив, як його використовувати і для чого він потрібний, також я навчився стоворювати програми на основі масивів.Також я дізнявся про динамічний і двовимірний масиви, дізнався чим вони відрізняються від звичайного масиву, а також в яких цілях вони використовуються.