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-13

Ласкавий Артем Олександрович

# **Тема роботи:**

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

# **Мета роботи:**

Ознайомитись з структурами даних та їх видами, одно- та двовимірними масивами, алгоритмами обробки (пошуку, сортування, обробки рядків)

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Прості структури даних. Одновимірні масиви
* Тема №2: Двовимірні масиви. Алгоритми обробки

1. Індивідуальний план опрацювання теорії:

* Тема №1: Прості структури даних. Одновимірні
  + Джерела Інформації.
    - Відео.
      * <https://youtu.be/8jLOx1hD3_o?si=4Ny410cAV5mNPQoT>
      * <https://youtu.be/86CQq3pKSUw?si=ttZQqjBVVFaoZYNX>
    - Стаття.
      * <https://www.geeksforgeeks.org/data-structures/>
      * <https://www.geeksforgeeks.org/array-data-structure/>
      * <https://www.geeksforgeeks.org/largest-sum-contiguous-subarray/>
  + Що опрацьовано:
    - Опрацьовано види простих структур даних, поняття масивів та методи роботи з ними.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.08.2023
  + Звершення опрацювання теми: 28.11.2023
* Тема №2: Двовимірні масиви. Алгоритми обробки
  + Джерела Інформації:
    - Відео.
      * <https://youtu.be/8jLOx1hD3_o?si=4Ny410cAV5mNPQoT>
      * <https://youtu.be/yCQN096CwWM?si=Quf0d_6wu2-IPfqR>
    - Стаття.
      * <https://www.geeksforgeeks.org/multidimensional-arrays-in-c/>
      * <https://www.geeksforgeeks.org/maximum-sum-rectangle-in-a-2d-matrix-dp-27/>
      * <https://www.geeksforgeeks.org/bubble-sort/>
      * <https://www.geeksforgeeks.org/merge-sort/>
  + Що опрацьовано:
    - Опрацьовано поняття двовимірних масивів, методи роботи з ними та алгоритмів обробки.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.10.2023
  + Звершення опрацювання теми: 20.11.2023

# **Виконання роботи:**

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

Завдання №1 **VNS Lab 4 - Task 1-17**

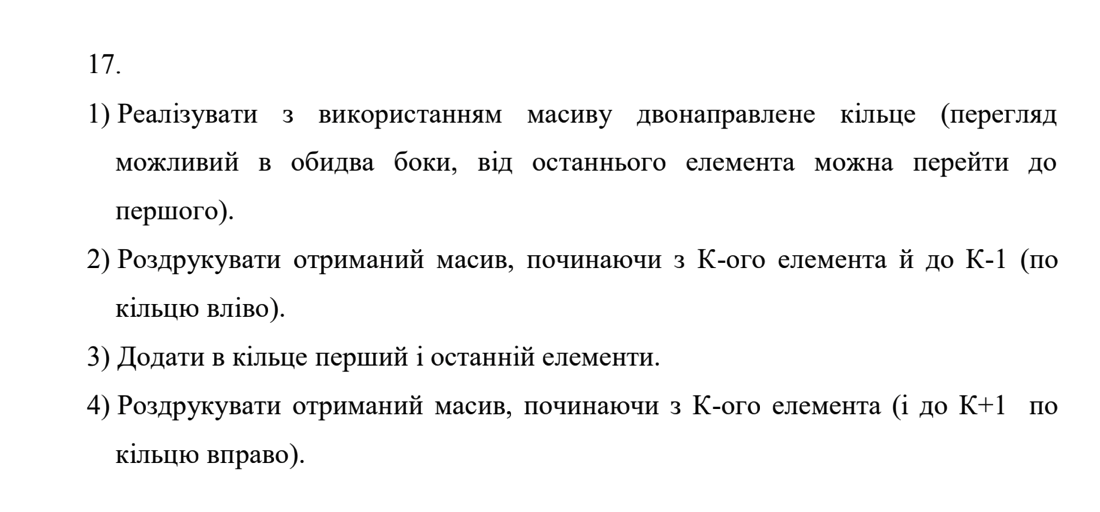
* Варіант завдання 17
* Деталі завдання:

Рисунок 1 - Умова до завдання №1

Завдання №2 **VNS Lab 5 - Task 1-17**

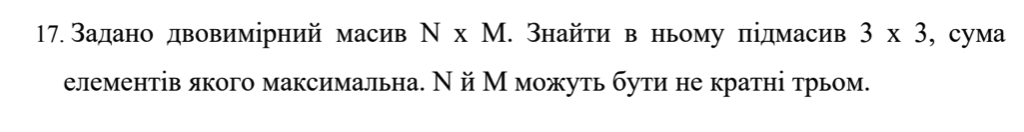
* Варіант завдання 17
* Деталі завдання:
* Використовуючи функції, розв’язати зазначене у варіанті завдання. Масив
* повинен передаватися у функцію як параметр. 

Рисунок 2 - Умова до завдання №2

Завдання №3 **Algotester Lab 2**

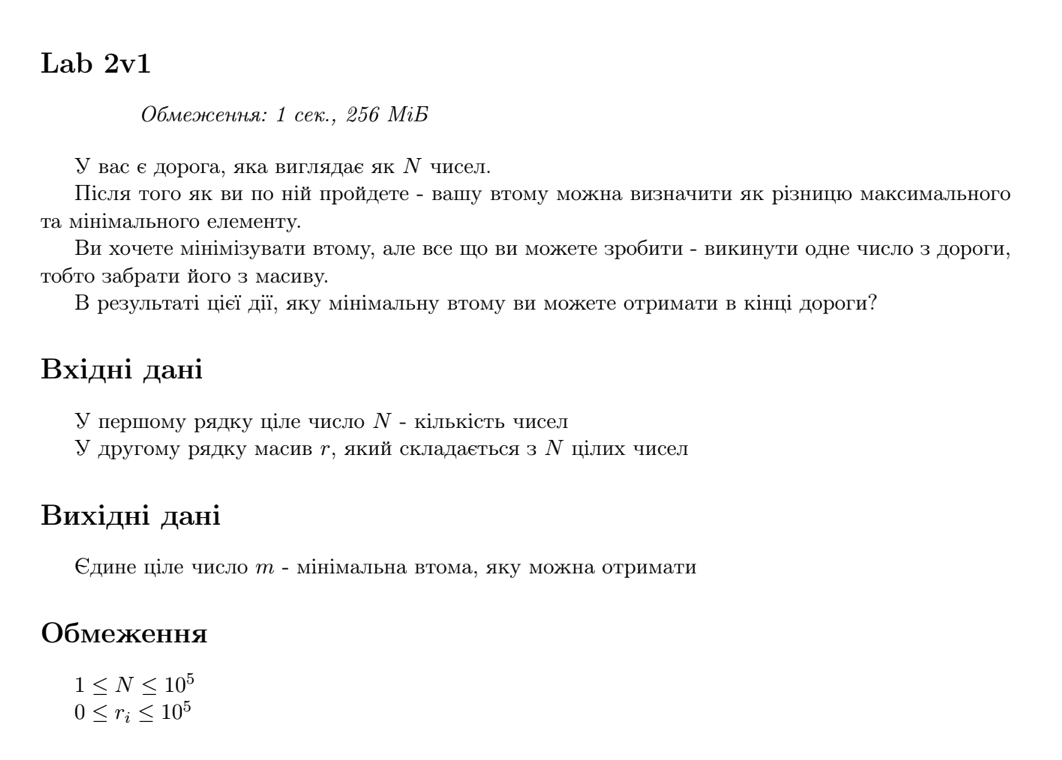
* Варіант завдання 1
* Деталі завдання:
* 

Рисунок 3 - Умова до завдання №3

* Важливі деталі для врахування в імплементації програми: мінімальна втома може бути отримана як викиданням з масиву найбільшого, так і найменшого елементів.

Завдання №4 **Algotester Lab 3**

Варіант завданння 2

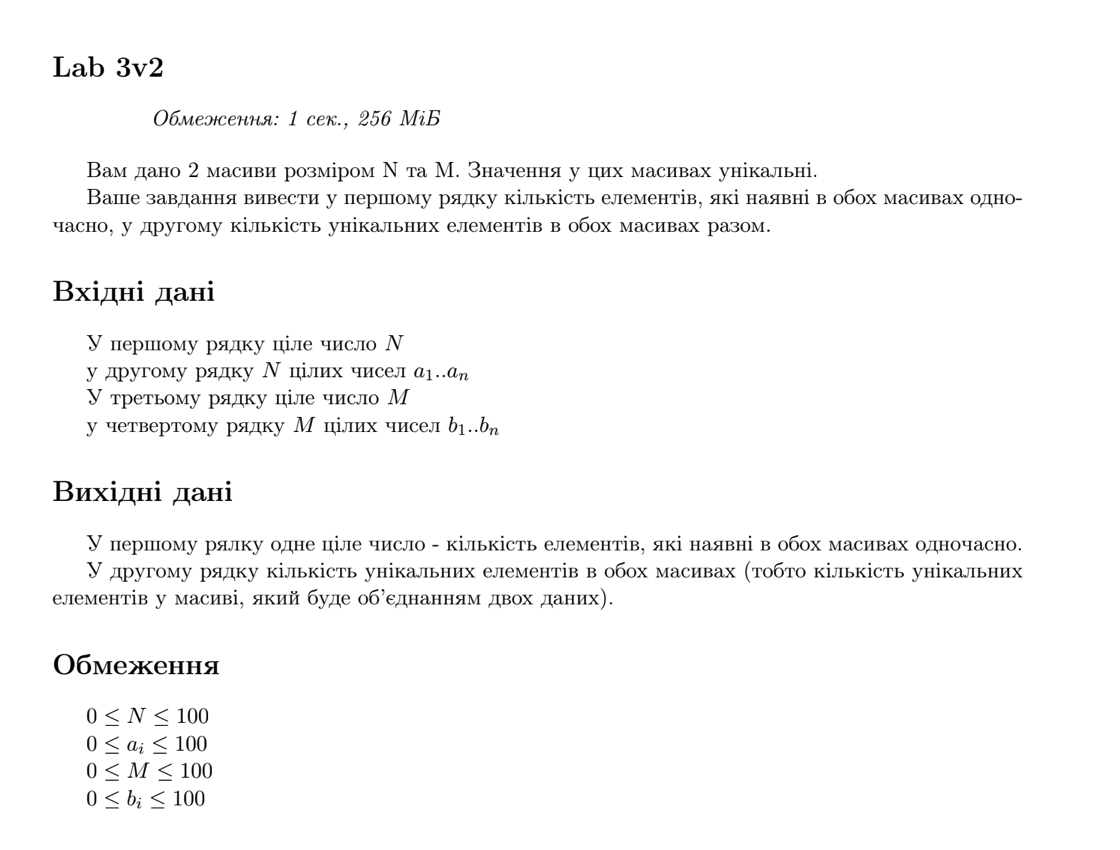
* Деталі завдання
* 

Рисунок 4 - Умова завдання №4

* Важливі деталі для врахування в імплементації програми: Користуватися std::set та std::map та їх похiдними ЗАБОРОНЕНО.

Завдання №5 **Class Practice Work**

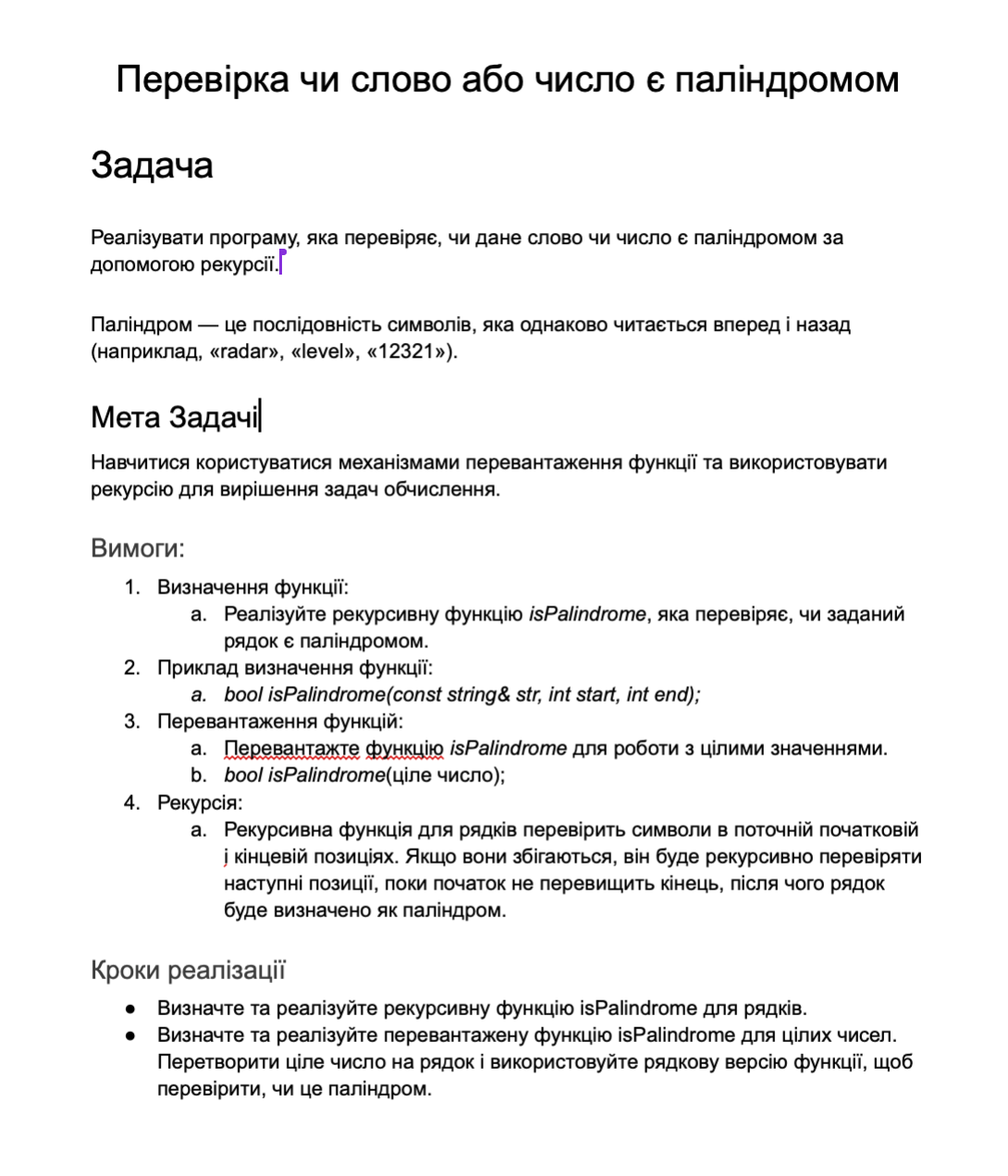
* Деталі завдання
* 

Рисунок 5 - Умова завдання №5

Завдання №6 **Алготестер 0182 – Зуби**

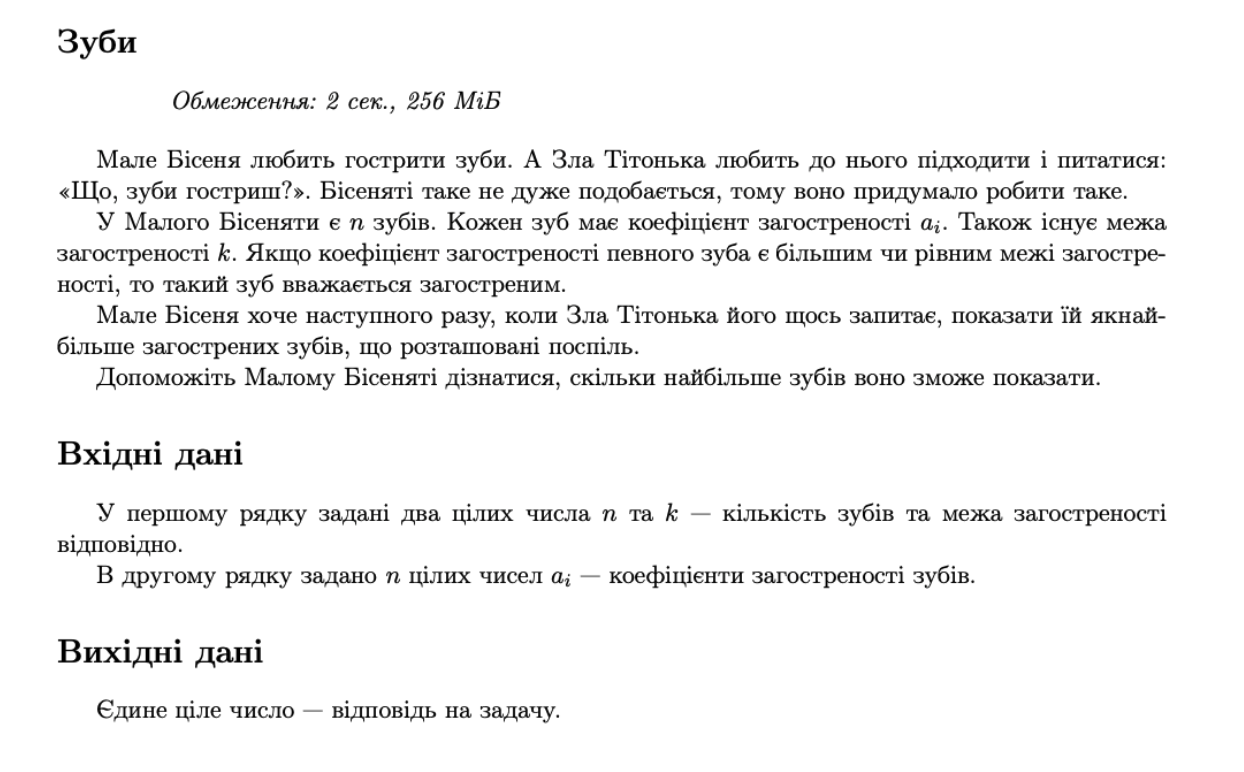
* Деталі завдання
* 

Рисунок 6 - Умова завдання №6

# **Завдання №7 Алготестер – Lab 2v2 | NULP\_LABS\_MPP\_2023**

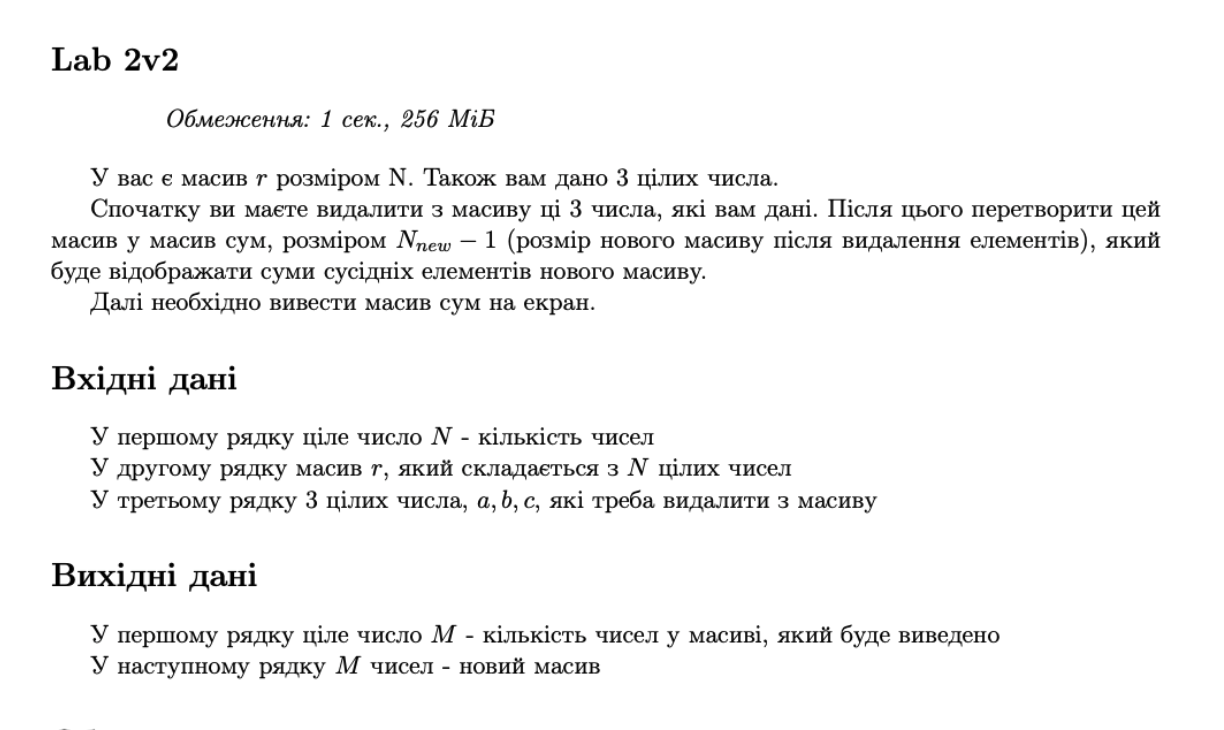
* Деталі завдання
* 

Рисунок 7 - Умова завдання №7

# **Завдання №8 Алготестер – Lab 2v3 | NULP\_LABS\_MPP\_2023**

* Деталі завдання

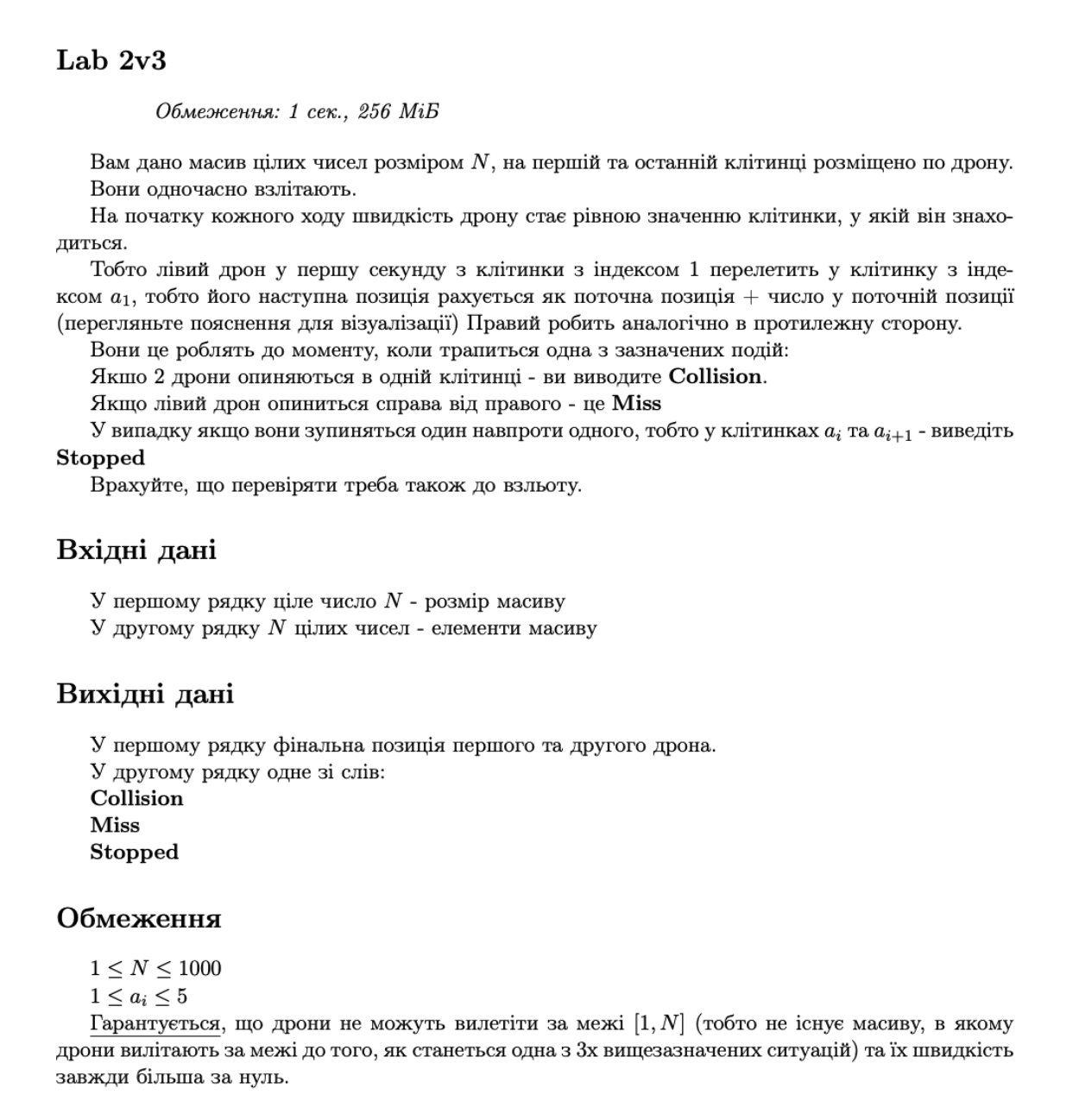


Рисунок 8 - Умова до завдання №8

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

Програма №1 **VNS Lab 4 - Task 1-17**

* Планований час на реалізацію: 120хв
* Важливі деталі для врахування в імплементації: необхідно використати circular doubly linked list.

Програма №2 **VNS Lab 5 - Task 1-17**

**Блок-схема:**

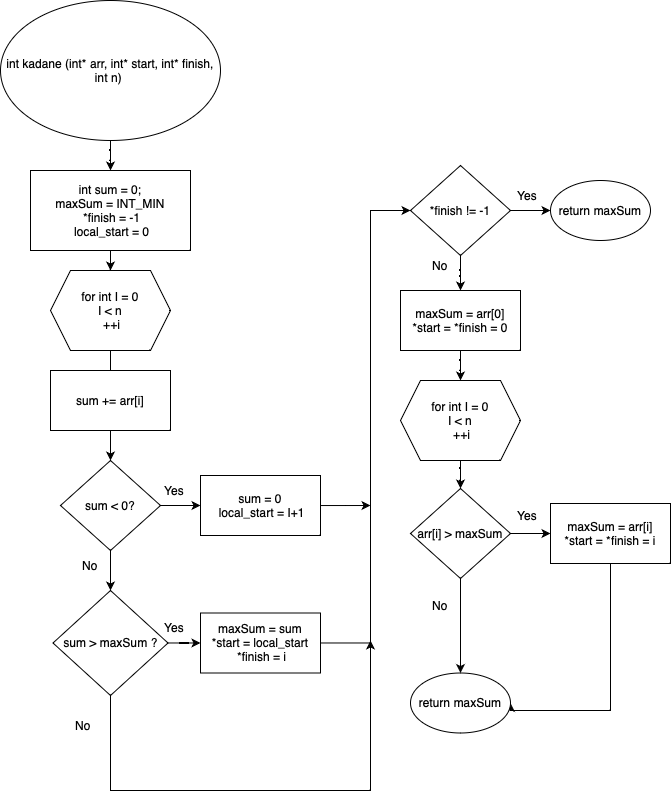


Рисунок 9 -- VNS Lab 5 - Task 1-17, int kadane

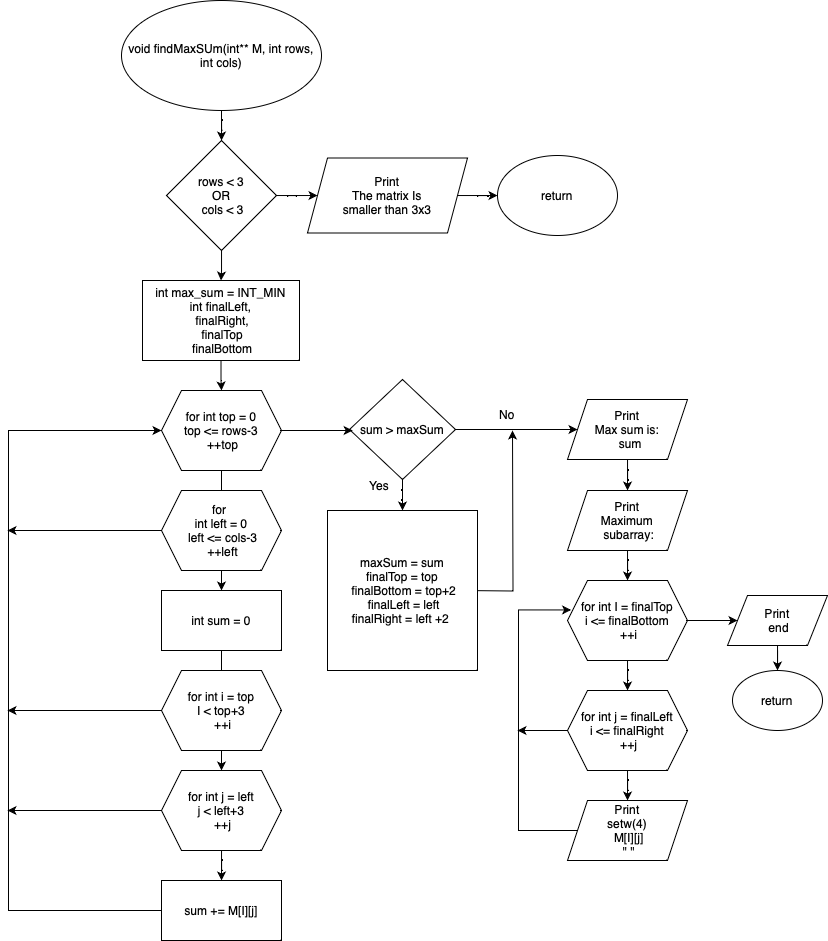


Рисунок 10 - VNS Lab 5 - Task 1-17, void findMaxSum

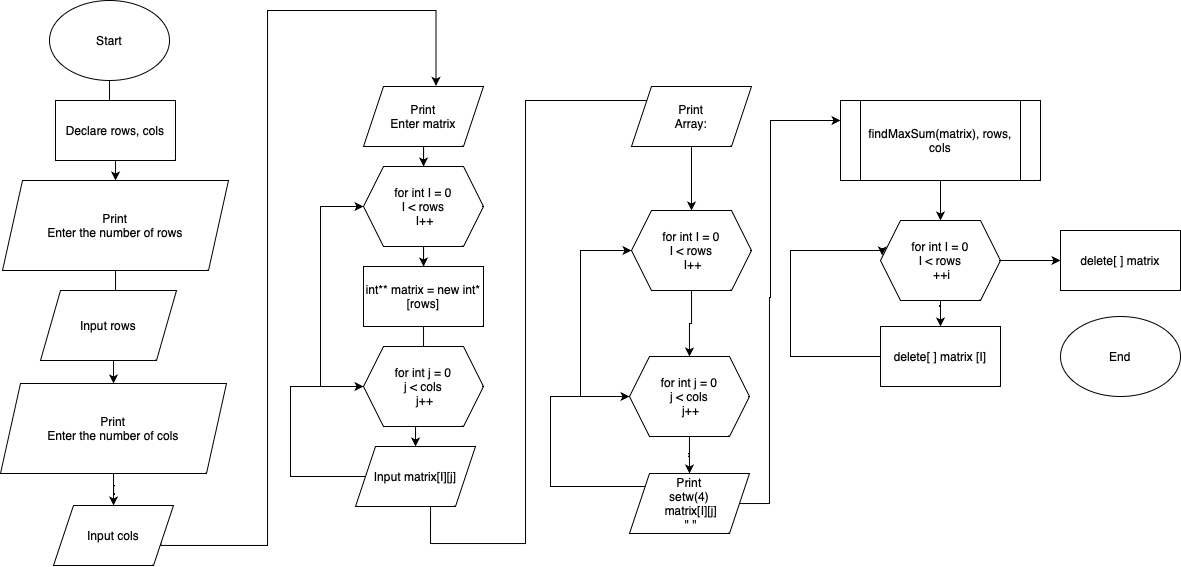


Рисунок 11 - VNS Lab 5 - Task 1-17, main

* Планований час на реалізацію: 150хв.
* Важливі деталі для врахування в імплементації:

Матриця може бути не кратна 3, складатися з додатніх та від’ємних чисел.

Програма №3 **Algotester Lab 2**

Планований час на реалізацію: 60хв.

* Важливі деталі для врахування в імплементації програми: мінімальна втома може бути отримана як викиданням з масиву найбільшого, так і найменшого елементів.

Програма №4 **Algotester Lab 3**

* Планований час на реалізацію 60 хв.
* Важливі деталі для врахування в імплементації програми: Користуватися std::set та std::map та їх похiдними ЗАБОРОНЕНО.

Програма №5 **Class Practice Work**

* Планований час на реалізацію: 35хв.

Програма №6 **Алготестер 0182 – Зуби**

* Планований час на реалізацію: 30хв.

Програма №7 **Алготестер – Lab 2v2 | NULP\_LABS\_MPP\_2023**

* Планований час на реалізацію: 60хв.

Програма №8 **Алготестер – Lab 2v3 | NULP\_LABS\_MPP\_2023**

* Планований час на реалізацію: 90хв.

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

Без змін

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

Завдання №1 **VNS Lab 4 - Task 1-17**

Посилання на Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_artem_laskavyi/ai_13/artem_laskavyi/Epic%204/vns_lab_4_task_1_variant_17_artem_laskavyi.cpp>

#include <iostream>

using namespace std;

struct Node{

struct Node \* prev;

int data;

struct Node \* next;

};

void addToEnd(Node\* & head, int value){

Node\* n = new Node{nullptr, value, nullptr};

if (!head) { // якщо кільце пусте

head = n;

head->prev = head;

head->next = head;

} else {

// поперeдній елемент <- новий елемент

n->prev = head->prev;

n->next = head;

// попередній елемент -> новий

head->prev->next = n;

head->prev = n;

}

}

void addToStart(Node\* & head, int value){

Node\* n = new Node{nullptr, value, nullptr};

if (!head) { // якщо кільце пусте

head = n;

head->prev = head;

head->next = head;

} else {

n->next = head->next;

head->next = n;

n->next->prev = n;

n->prev = head;

}

}

void displayClockwise(const Node\* head) { // по кільцю вліво

const Node\* current = head;

do {

cout << current->data << " ";

current = current->next;

} while (current != head);

cout << endl;

}

void displayAntiClockwise(const Node\* head) { // по кільцю вправо

const Node\* current = head;

do {

cout << current->data << " ";

current = current->prev;

} while (current != head);

cout << endl;

}

int main()

{

Node\* myList = nullptr;

addToEnd(myList, 5);

addToEnd(myList, 10);

addToEnd(myList, 3);

addToEnd(myList, 7);

addToEnd(myList, 1);

displayClockwise(myList);

addToStart(myList, 25);

addToEnd(myList, 50);

displayAntiClockwise(myList);

return 0;

}

Рисунок 12 - Код програми №1

Завдання №2 **VNS Lab 5 - Task 1-17**

Посилання на Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_artem_laskavyi/ai_13/artem_laskavyi/Epic%204/vns_lab_5_task_1_variant_17_artem_laskavyi.cpp>

#include <iomanip>

using namespace std;

int kadane(int\* arr, int\* start, int\* finish, int n);

void findMaxSum(int\*\* M, int rows, int cols);

int main() {

int rows, cols;

cout << "Enter the number of rows: ";

cin >> rows;

cout << "Enter the number of columns: ";

cin >> cols;

cout << "Enter Matrix: ";

int\*\* matrix = new int\*[rows];

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

matrix[i] = new int[cols];

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

cin >> matrix[i][j];

}

}

cout << "Array:\n";

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

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

cout << setw(4) << matrix[i][j] << " ";

}

cout << endl;

}

findMaxSum(matrix, rows, cols);

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

delete[] matrix[i];

}

delete[] matrix;

return 0;

}

int kadane(int\* arr, int\* start, int\* finish, int n)

{

int sum = 0, maxSum = INT\_MIN, i; // int min - minimal value that can be held by int

\*finish = -1;

int local\_start = 0;

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

sum += arr[i];

if (sum < 0) {

sum = 0;

local\_start = i + 1;

}

else if (sum > maxSum) {

maxSum = sum;

\*start = local\_start;

\*finish = i;

}

}

if (\*finish != -1)

return maxSum;

// When all numbers

// in arr[] are negative

maxSum = arr[0];

\*start = \*finish = 0;

// Find the maximum element in array

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

if (arr[i] > maxSum) {

maxSum = arr[i];

\*start = \*finish = i;

}

}

return maxSum;

}

void findMaxSum(int\*\* M, int rows, int cols) {

if (rows < 3 || cols < 3) {

cout << "The matrix is smaller than 3x3." << endl;

return;

}

int maxSum = INT\_MIN;

int finalLeft, finalRight, finalTop, finalBottom;

for (int top = 0; top <= rows - 3; ++top) { // rows - 3 and cols-3 to ensure matrix is never smaller than 3x3

for (int left = 0; left <= cols - 3; ++left) {

int sum = 0; // Initialize sum for each 3x3 submatrix

for (int i = top; i < top + 3; ++i) {

for (int j = left; j < left + 3; ++j) {

sum += M[i][j];

}

}

// Compare the sum with the maximum sum so far

if (sum > maxSum) {

maxSum = sum;

finalTop = top;

finalBottom = top + 2; // 3 rows

finalLeft = left;

finalRight = left + 2; // 3 columns

}

}

}

cout << "\nMax sum is: " << maxSum << endl;

cout << "\nMaximum sum subarray:" << endl;

for (int i = finalTop; i <= finalBottom; ++i) {

for (int j = finalLeft; j <= finalRight; ++j) {

cout << setw(4) << M[i][j] << " ";

}

cout << endl;

}

}

13 - Код програми №2

Завдання №3 **Algotester Lab 2**

Посилання на Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_artem_laskavyi/ai_13/artem_laskavyi/Epic%204/algotester_lab_2_task_1_variant_1_artem_laskavyi.cpp>

#include <iostream>

#include <algorithm>

using namespace std;

int main(){

int N, m1, m2, result; //N - к-сть чисел масиву, m - втома

cin >> N;

int r[N];

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

cin >> r[i];

}

sort(r, r+N);

m1 = r[N-2] - r[0]; // with second greatest value

m2 = r[N-1] - r[1]; // with second smallest value

if(N==1 || N==2){

result = 0;

}else{

result = (m1<m2)? m1 : m2;

}

cout << result << endl;

return 0;

}

Рисунок 14 - Код програми №3

Завдання №4 **Algotester Lab 3**

Посилання на Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_artem_laskavyi/ai_13/artem_laskavyi/Epic%204/algotester_lab_3_task_1_variant_2_artem_laskavyi.cpp>

#include <iostream>

#include <vector>

using namespace std;

void findIntersection( vector<int> &a, vector<int> &b, vector<int> &c){

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

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

if(a[i] == b[j]){

c.push\_back(a[i]);

break;

}

}

}

}

int main(){

int N, M, num;

vector<int> a, b, intersec;

cin >> N;

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

cin >> num;

a.push\_back(num);

}

cin >> M;

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

cin >> num;

b.push\_back(num);

}

findIntersection(a, b, intersec);

cout << intersec.size() << endl;

cout << (M + N) - intersec.size() << endl;

return 0;

}

Рисунок 15 - Код програми №4

Програма №5 **Class Practice Work**

Посилання на Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_artem_laskavyi/ai_13/artem_laskavyi/Epic%204/practice_work_team_tasks_artem_laskavyi.cpp>

#include <iostream>

#include <vector>

#include <iostream>

#include <vector>

#include <algorithm>

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(const vector<int>& vec, int start, int end){

if (start >= end) return true;

if (vec[start] != vec[end]) {

return false;

}

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

}

int main(){

string str1 = "radar";

string str2 = "level";

string str3 = "forest";

vector<int> vec1 = {1, 2, 3, 4, 3, 2, 1};

vector<int> vec2 = {1, 1, 1};

vector<int> vec3 = {12, 17, 99};

vector<int> vec4 = {12, 21};

if (isPalindrome(str1, 0, str1.length() - 1)) {

cout << "Palindrome.\n";

} else {

cout << "Not a palindrome.\n";

}

if (isPalindrome(vec1, 0, vec1.size() - 1)) {

cout << "Palindrome.\n";

} else {

cout << "Not a palindrome.\n";

}

if (isPalindrome(str2, 0, str2.length() - 1)) {

cout << "Palindrome.\n";

} else {

cout << "Not a palindrome.\n";

}

if (isPalindrome(vec2, 0, vec2.size() - 1)) {

cout << "Palindrome.\n";

} else {

cout << "Not a palindrome.\n";

}

if (isPalindrome(str3, 0, str3.length() - 1)) {

cout << "Palindrome.\n";

} else {

cout << "Not a palindrome.\n";

}

if (isPalindrome(vec3, 0, vec3.size() - 1)) {

cout << "Palindrome.\n";

} else {

cout << "Not a palindrome.\n";

}

if (isPalindrome(vec4, 0, vec3.size() - 1)) {

cout << "Palindrome.\n";

} else {

cout << "Not a palindrome.\n";

}

return 0;

}

Рисунок 16 - Код програми №5

Програма №6 **Алготестер 0182 – Зуби**

Посилання на Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_artem_laskavyi/ai_13/artem_laskavyi/Epic%204/practice_work_self_algotester_task_1_artem_laskavyi.cpp>

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

int main() {

// count counts consecutive sharp teeth

int n, k, sharpness, count = 0;

cin >> n >> k;

vector<int> seq;

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

cin >> sharpness;

if(sharpness >= k){

count++;

}else{

seq.push\_back(count);

count = 0;

}

}

seq.push\_back(count);

cout << \*max\_element(seq.begin(), seq.end()) << endl;

return 0;

}

Рисунок 17 - Код програми №6

Програма №7 **Алготестер – Lab 2v2 | NULP\_LABS\_MPP\_2023**

Посилання на Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_artem_laskavyi/ai_13/artem_laskavyi/Epic%204/practice_work_self_algotester_task_2_artem_laskavyi.cpp>

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main(){

int N, num;

vector<int> arr, del;

vector<int>::iterator it;

cin >> N;

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

cin >> num;

arr.push\_back(num);

}

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

cin >> num;

del.push\_back(num);

}

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

auto it = find(arr.begin(), arr.end(), del[i]);

if (it != arr.end()){

arr.erase(it);

}

}

int new\_size = (arr.size() != 0) ? arr.size()-1 : 0;

int arrnew[new\_size];

for (int i=0; i < new\_size; i++){

arrnew[i] = arr[i]+ arr[i+1];

}

cout << new\_size << endl;

for (int i=0; i < new\_size; i++){

cout << arrnew[i] << " ";

}

cout << endl;

return 0;

}

Рисунок 18 - Код програми №7

Програма №8 **Алготестер – Lab 2v3 | NULP\_LABS\_MPP\_2023**

Посилання на Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_artem_laskavyi/ai_13/artem_laskavyi/Epic%204/practice_work_self_algotester_task_3_artem_laskavyi.cpp>

#include <iostream>

#include <vector>

using namespace std;

int main(){

int N, num;

vector<int> a;

cin >> N;

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

cin >> num;

a.push\_back(num);

}

int drone1 = 0;

int drone2 = a.size()-1;

bool flying = true;

while(flying == true){

if(drone1 == drone2){

flying = false;

cout << drone1 + 1 << " " << drone2 + 1 << endl;

cout << "Collision" << endl;

break;

}else if(drone1 == drone2 - 1){

flying = false;

cout << drone1 + 1 << " " << drone2 + 1 << endl;

cout << "Stopped" << endl;

break;

}else if (drone1 > drone2){

flying = false;

cout << drone1 + 1 << " " << drone2 + 1 << endl;

cout << "Miss" << endl;

break;

}

drone1 += a[drone1];

drone2 -= a[drone2];

}

return 0;

}

Рисунок 19 - Код програми №8

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

Завдання №1 **VNS Lab 4 - Task 1-17**

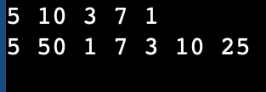


Рисунок 20 - Тестування програми №1

Час затрачений на виконання завдання: 30хв.

Завдання №2 **VNS Lab 5- Task 1-17**

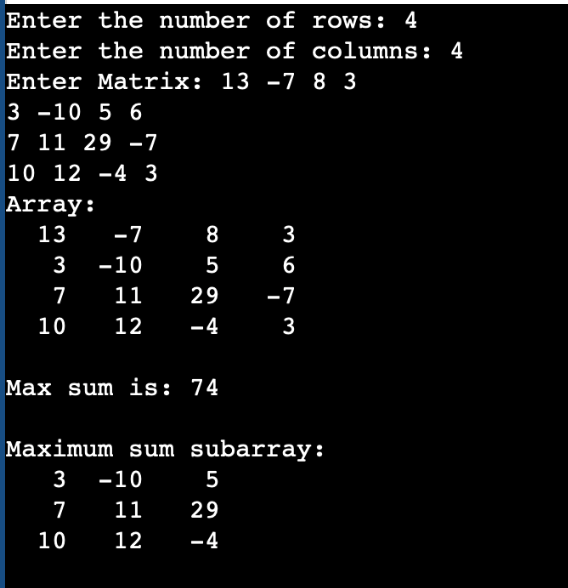


Рисунок 21 - Тестування програми №2

Час затрачений на виконання завдання: 80хв.

Завдання №3 **Algotester Lab 2**

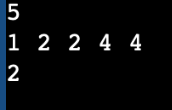


Рисунок 22 - Тестування програми №3\_1



Рисунок 23 - Тестування програми №3\_2

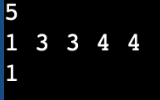


Рисунок 24 - Тестування програми №3\_3



Рисунок 25 - Тестування програми №3\_3

Час затрачений на виконання завдання: 150хв.

Завдання №4 **V Algotester Lab 3**



Рисунок 26 - Тестування програми №4

Час затрачений на виконання завдання: 20хв.

Програма №5 **Class Practice Work**

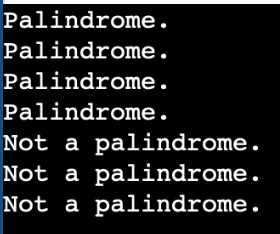


Рисунок 27 - Тестування програми №5

Час затрачений на виконання завдання: 60хв.

Програма №6 **Алготестер 0182 – Зуби**

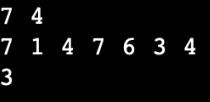


Рисунок 28 - Тестування програми №6

Час затрачений на виконання завдання: 5хв.

Програма №7 **Алготестер – Lab 2v2 | NULP\_LABS\_MPP\_2023**

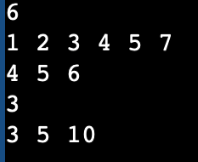


Рисунок 29 - Тестування програми №7\_1

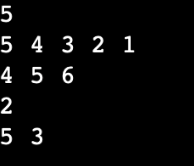


Рисунок 30 - Тестування програми №7\_2

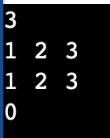


Рисунок 31 - Тестування програми №7\_3

Час затрачений на виконання завдання: 5хв.

Завдання №8 **Алготестер – Lab 2v3 | NULP\_LABS\_MPP\_2023**

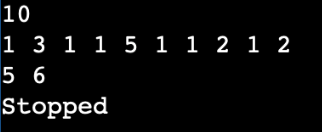


Рисунок 32 - Тестування програми №8\_1

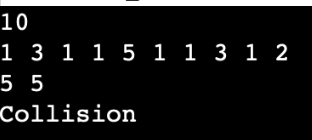


Рисунок 33 - Тестування програми №8\_2

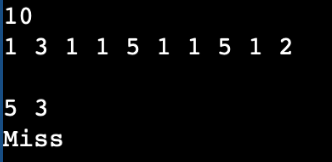


Рисунок 34 - Тестування програми №8\_3

Час затрачений на виконання завдання: 180хв.

## **6. Кооперація з командою:**

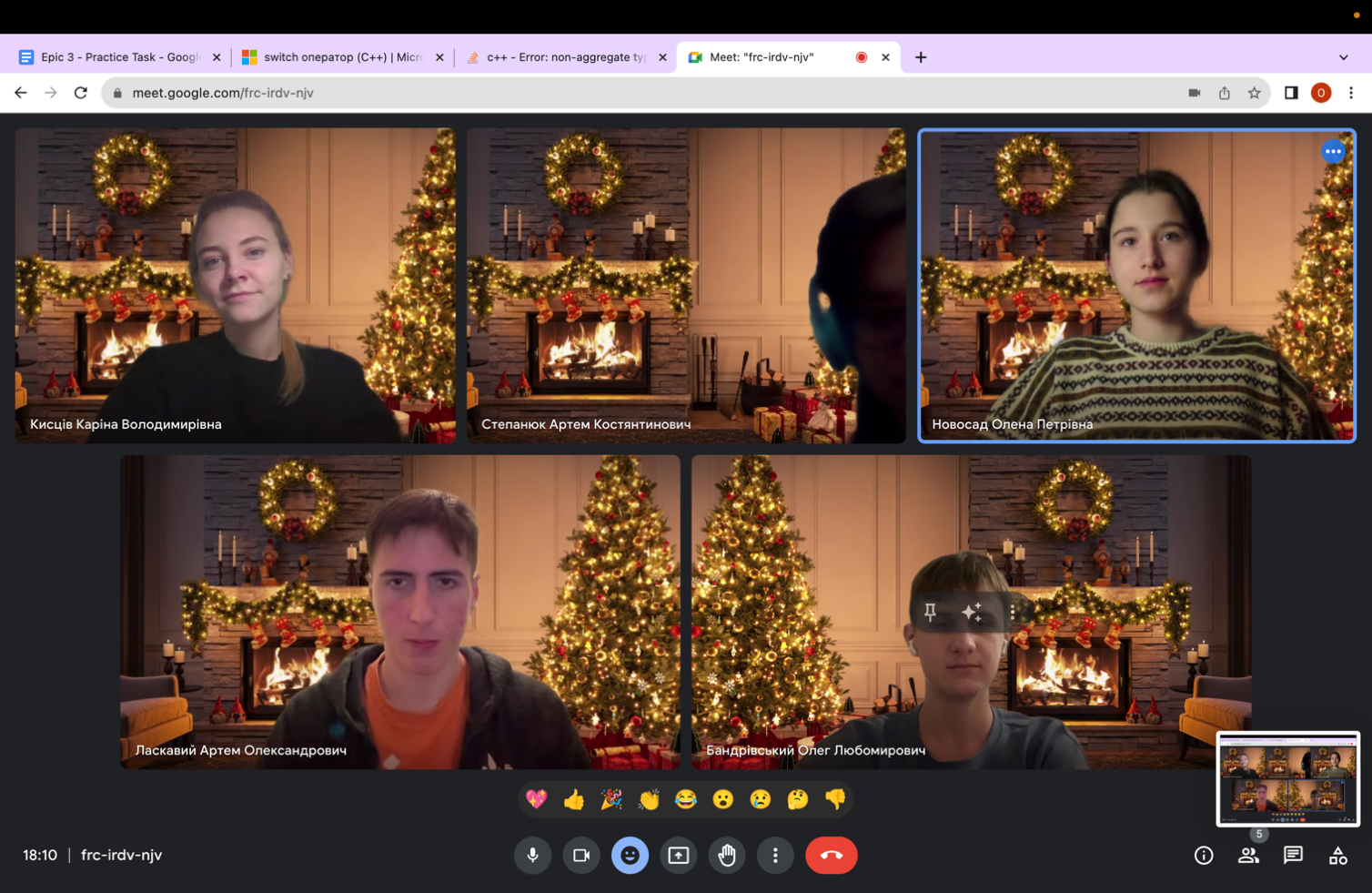
* Скрін з 1-ї зустрічі по обговоренню задач Епіку:
* 

Рисунок 35 - Скріншот зустрічі №1 команди

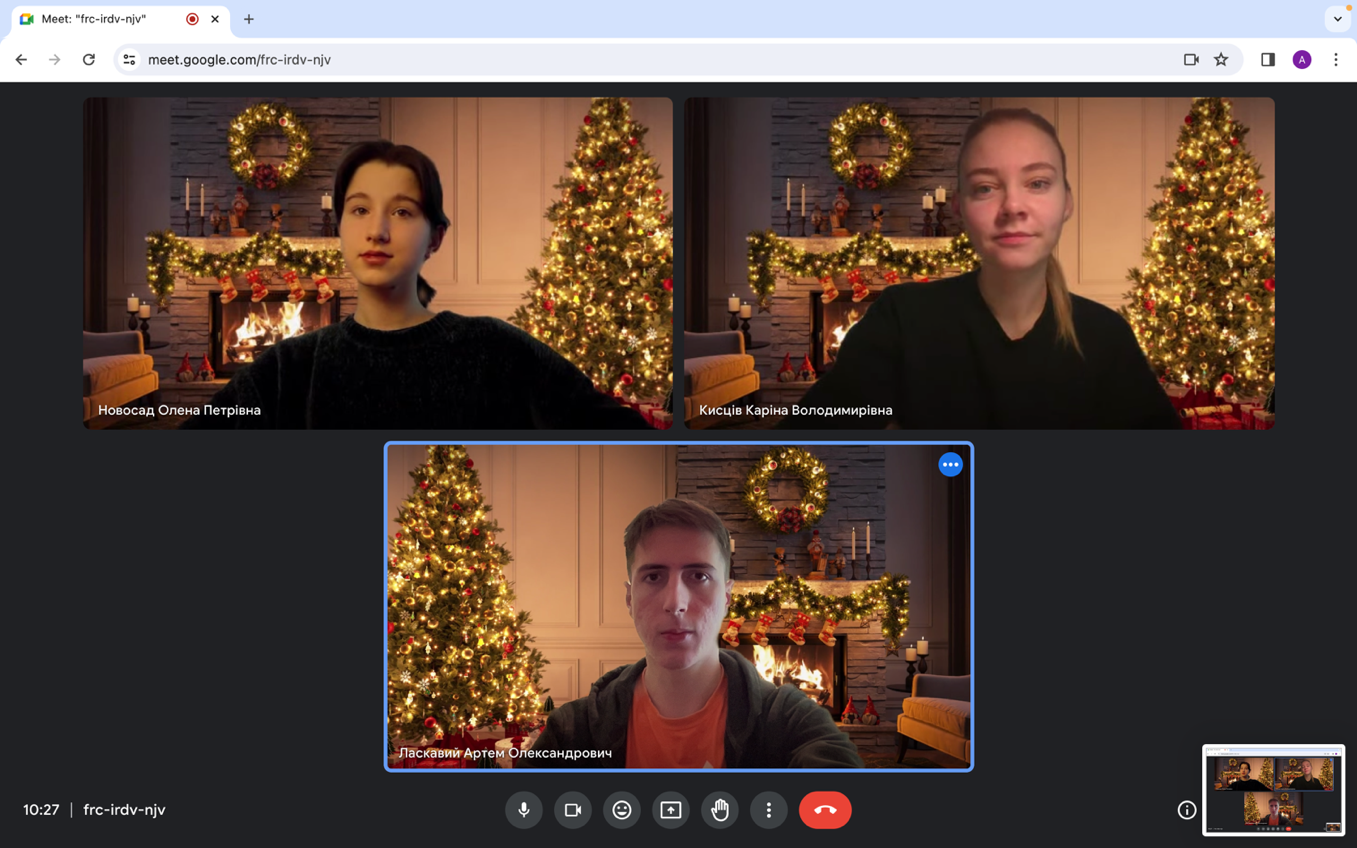
* Скрін з 2-ї зустрічі по обговоренню задач Епіку
* 

Рисунок 36 - Скріншот зустрічі №2 команди

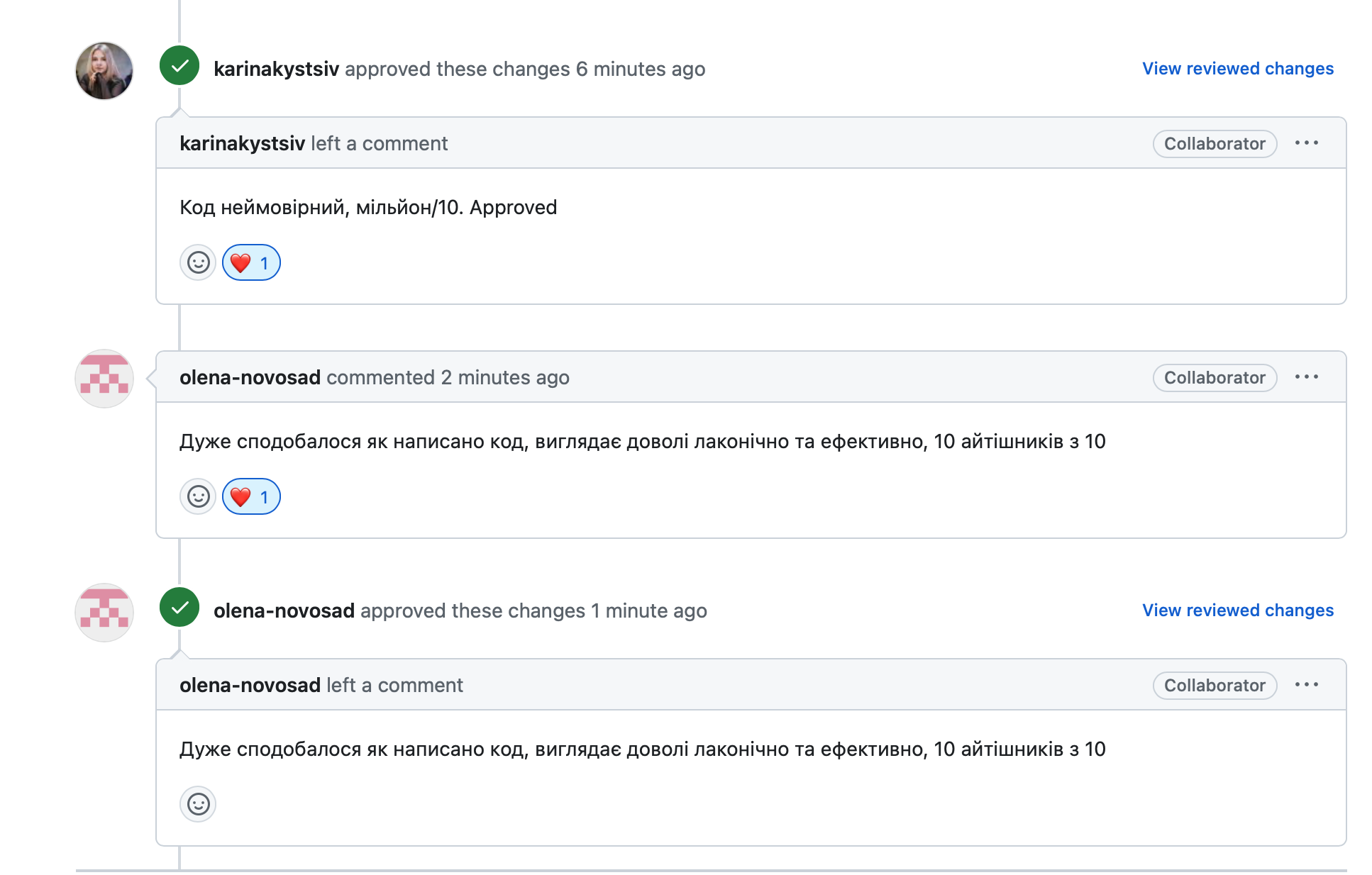
* Скрін з 2-мa коментарями від учасників команди на пул реквесті з Ревю Роботи
* 

Рисунок 37 - скріншот з коментарями від учасників команди

# **Висновки:**

Під час виконання епіку 3 я ознайомився з структурами даних та їх видами, одно- та двовимірними масивами, алгоритмами обробки (пошуку, сортування, обробки рядків), алгоритмом Кадейна з пошуку найбільшої неперевної послідовності в масиві, структурою даних двозв’язне кільце. Реалізував програми для пошуку найбільшої підматриці 3х3 та перевірки чи є слово паліндромом.