Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

**До виконання лабораторних та практичних робіт блоку №4**

На тему : “Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами. ”

**З дисципліни :** “ Oснови програмування ”

До

ВНС Лабораторної Роботи №4

ВНС Лабораторної Роботи №5

Алготестер Лабораторної Роботи №2

Алготестер Лабораторної Роботи №3

Практичних Робіт до блоку 4

Виконала :

Студентка групи ШІ-12

Куць Софія Тарасівна

Львів 2024

Тема: Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.

Теоретичні відомості:

Тема №1 Класи памʼяті у С++

<https://acode.com.ua/urok-89-dynamichne-vydilennya-pam-yati/> <https://youtu.be/NyOjKd5Qruk?si=2aJy5iawzBdpQssC>

Тема №2 Масиви та Вказівники

<https://youtu.be/zopWRlYOXWw?si=UkUj0MEp2iAVY7r1>

<https://youtu.be/ULdbOaMBPYc?si=jRkgguAgqGVq7htD>

<https://youtu.be/btNOGePn778?si=s3_wQdQ5BCfqSynW>

<https://acode.com.ua/urok-77-masyvy/>

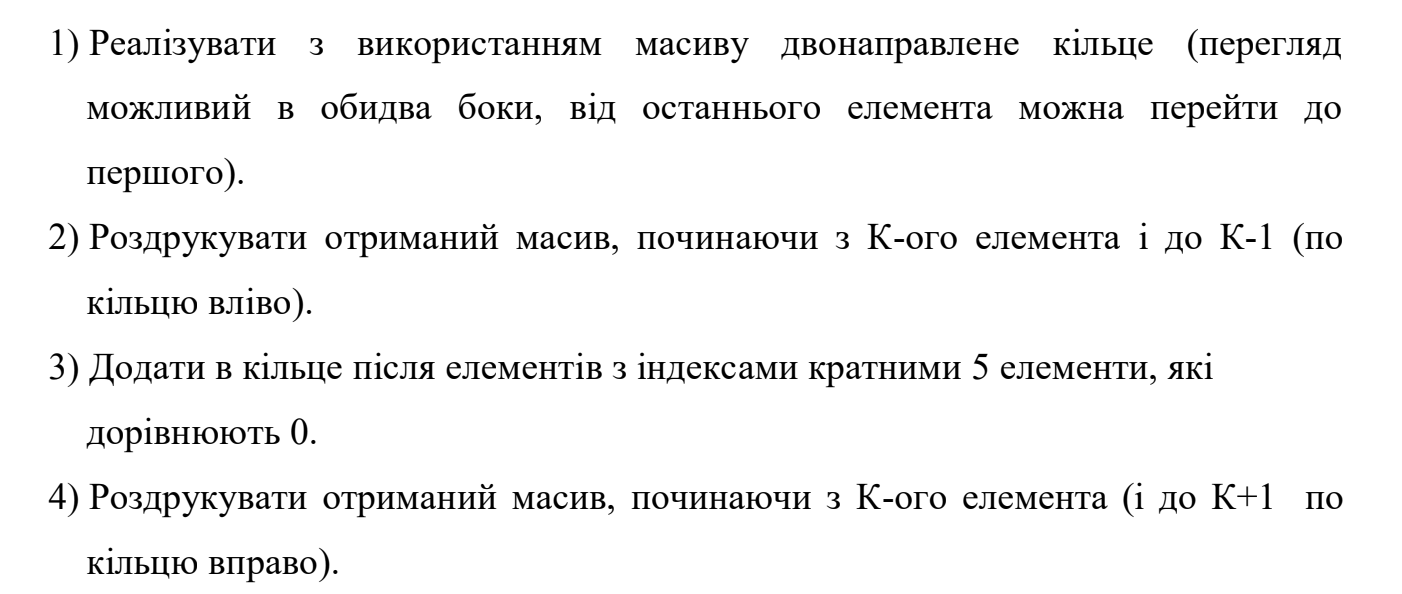
Тема №3 Структури даних, вкладені структури та використання структур

<http://cpp.dp.ua/vykorystannya-struktur/>

<https://youtu.be/999IE-6b7_s?si=SCdfZu-J5F82tuux>

**Виконання роботи :**

Завдання №1 ( VNS Lab 4, варіант 20 )



Завдання №2( VNS Lab 5, варіант 20 )

Використовувати функції . Масив повинен передаватися у функцію як параметр. Знайти мінімальний з неповторюваних елементів двовимірного масиву.

Завдання №3( Algotester Lab 2, варіант 1 )

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?

**Input**

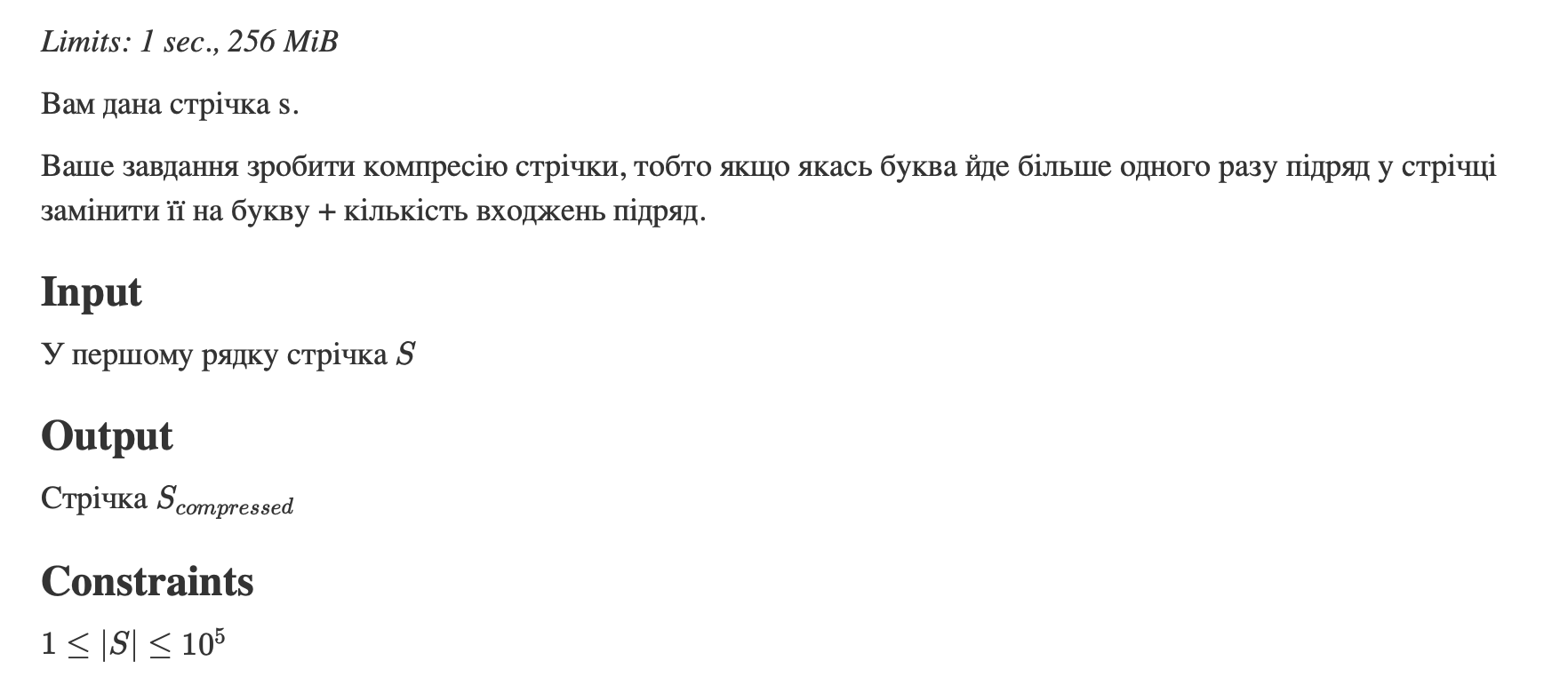
У першому рядку ціле число N - кількість чисел

У другому рядку масив r, який складається з N цілих чисел

**Output**

Єдине ціле число m - мінімальна втома, яку можна отримати

Завдання №4 (Algotester Lab 3, варіант 3)



Завдання №5 ( Class Practice Work)

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»).

### **Вимоги:**

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.

### **Кроки реалізації**

* Визначте та реалізуйте рекурсивну функцію isPalindrome для рядків.
* Визначте та реалізуйте перевантажену функцію isPalindrome для цілих чисел. Використати математичний підхід щоб перевірити чи число є паліндромом.

Завдання №6 ( Self Practice Work)

Вам дано 2 масиви розміром 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.

**Input**

У першому рядку ціле число N

у другому рядку N цілих чисел a1..an

У третьому рядку ціле число M

у четвертому рядку M цілих чисел b1..bn

# Output

У першому рялку одне ціле число - кількість елементів, які наявні в обох масивах одночасно.

У другому рядку кількість унікальних елементів в обох масивах (тобто кількість унікальних елементів у масиві, який буде об’єднанням двох даних).

**Constraints**

0≤N≤100

0≤ai≤100

0≤M≤100

0≤bi≤100

**Дизайн та планова оцінка часу виконання завдань**

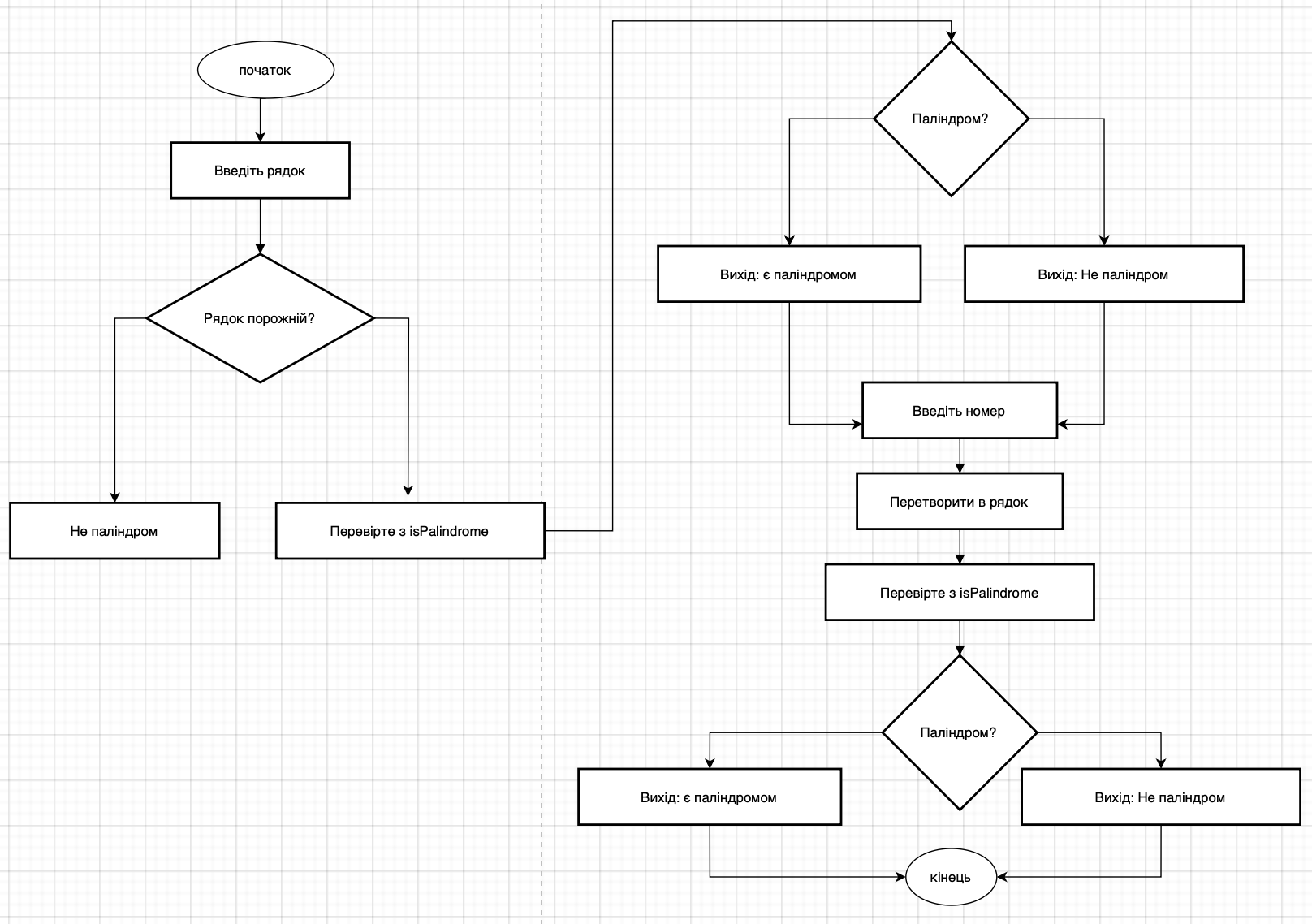
Завдання №1 ( VNS Lab 4, варіант 20 ) плановий час на виконання : 40хв

Завдання №2( VNS Lab 5, варіант 20 ) плановий час на виконання : 1 год

Завдання №3( Algotester Lab 2, варіант 1 ) плановий час на виконання : 35 хв

Завдання №4 (Algotester Lab 3, варіант 3) плановий час на виконання : 25 хв

Завдання №5 ( Class Practice Work) плановий час на виконання : 1 год



no

yes

no

yes

no

yes

Завдання №6 ( Self Practice Work) плановий час на виконання : 40 хв

Код програм

Завдання №1 ( VNS Lab 4, варіант 20 )

#include <iostream>

#include <vector>

using namespace std;

class CircularArray {

private:

vector<int> elements;

public:

CircularArray(const vector<int>& init) : elements(init) {}

int get(int index) const {

int n = elements.size();

return elements[(index % n + n) % n];

}

void insertAfter(int index, int value) {

int pos = (index + 1) % elements.size();

elements.insert(elements.begin() + pos, value);

}

void print() const {

for (size\_t i = 0; i < elements.size(); ++i) {

cout << elements[i] << " ";

}

cout << endl;

}

int size() const {

return elements.size();

}

int operator[](int index) const {

return get(index);

}

};

void printLeftFromK(const CircularArray& array, int k) {

vector<int> result;

int n = array.size();

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

result.push\_back(array[i]);

}

cout << "Кільце вліво від K: ";

for (size\_t i = 0; i < result.size(); ++i) {

cout << result[i] << " ";

}

cout << endl;

}

void addZerosAfterMultiplesOfFive(CircularArray& array) {

int shift = 0; // Зсув через додані елементи

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

if ((i % 5) == 0) {

array.insertAfter(i + shift, 0);

++shift;

}

}

}

void printRightFromK(const CircularArray& array, int k) {

vector<int> result;

int n = array.size();

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

result.push\_back(array[i]);

}

cout << "Кільце вправо від K: ";

for (size\_t i = 0; i < result.size(); ++i) {

cout << result[i] << " ";

}

cout << endl;

}

int main() {

vector<int> init;

for (int i = 1; i <= 20; ++i) {

init.push\_back(i);

}

CircularArray circularArray(init);

cout << "Початковий масив: ";

circularArray.print();

int K = 5;

printLeftFromK(circularArray, K);

addZerosAfterMultiplesOfFive(circularArray);

cout << "Після додавання нулів: ";

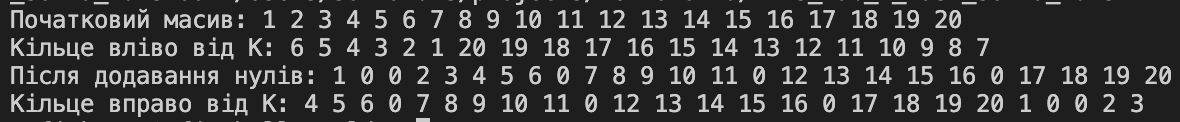
circularArray.print();

printRightFromK(circularArray, K);

return 0;

}

Результат:



Витрачений час : 55 хв

Завдання №2( VNS Lab 5, варіант 20 )

#include <iostream>

#include <vector>

using namespace std;

bool isUnique(int element, int rows, int cols, int\*\* array) {

int count = 0;

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

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

if (array[i][j] == element) {

++count;

if (count > 1) return false;

}

}

}

return true;

}

int findMinUnique(int rows, int cols, int\*\* array) {

int minUnique = INT\_MAX;

bool found = false;

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

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

if (isUnique(array[i][j], rows, cols, array)) {

found = true;

if (array[i][j] < minUnique) {

minUnique = array[i][j];

}

}

}

}

return found ? minUnique : -1;

}

int main() {

int rows, cols;

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

cin >> rows >> cols;

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

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

array[i] = new int[cols];

}

cout << "Enter the elements of the array:" << endl;

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

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

cin >> array[i][j];

}

}

int minUnique = findMinUnique(rows, cols, array);

if (minUnique == -1) {

cout << "No unique elements found in the array." << endl;

} else {

cout << "The minimum unique element is: " << minUnique << endl;

}

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

delete[] array[i];

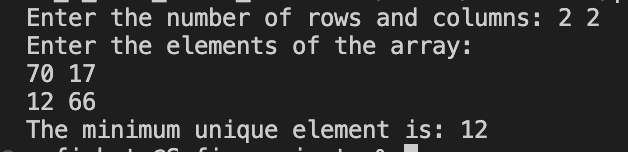
}

delete[] array;

return 0;

}

Резульат:



Час виконання: 40 хв

Завдання №3( Algotester Lab 2, варіант 1 )

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

int N;

cin >> N;

vector<int> r(N);

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

cin >> r[i];

}

if (N == 1) {

cout << 0 << endl;

return 0;

}

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

int removeMinFatigue = r[N - 2] - r[0];

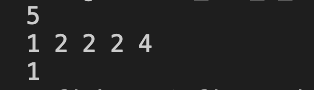
int removeMaxFatigue = r[N - 1] - r[1];

cout << min(removeMinFatigue, removeMaxFatigue) << endl;

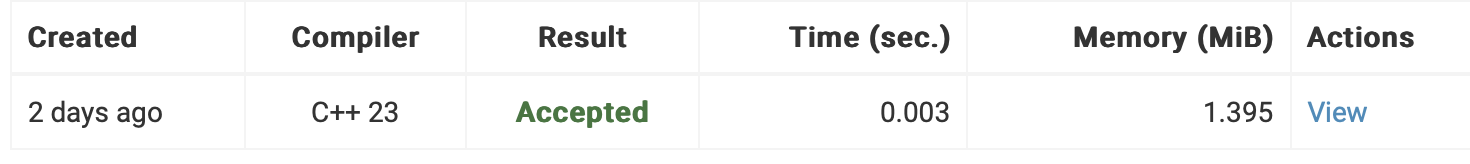
return 0;

}

Результат:



Час виконання : ~20 хв



Завдання №4 (Algotester Lab 3, варіант 3)

#include <iostream>

#include <string>

using namespace std;

int main() {

string s;

cin >> s;

string result = "";

int count = 1;

for (size\_t i = 1; i <= s.length(); ++i) {

if (i < s.length() && s[i] == s[i - 1]) {

++count;

} else {

result += s[i - 1];

if (count > 1) {

result += to\_string(count);

}

count = 1;

}

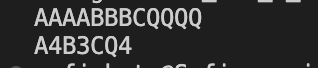
}

cout << result << endl;

return 0;

}

Результат:



Час виконання : ~20 хв



Завдання №5 ( Class Practice Work)

#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(int num) {

string str = to\_string(num);

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

}

int main() {

string testString;

cout << "Enter a string to check if it's a palindrome: ";

cin >> testString;

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

cout << "The string \"" << testString << "\" is a palindrome." << endl;

} else {

cout << "The string \"" << testString << "\" is not a palindrome." << endl;

}

int testNumber;

cout << "Enter a number to check if it's a palindrome: ";

cin >> testNumber;

if (isPalindrome(testNumber)) {

cout << "The number " << testNumber << " is a palindrome." << endl;

} else {

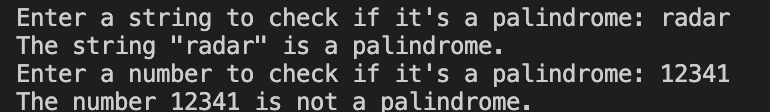
cout << "The number " << testNumber << " is not a palindrome." << endl;

}

return 0;

}

Результат:



Час виконання : 40 хв

Завдання №6 ( Self Practice Work)

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

int N, M;

cin >> N;

vector<int> a(N);

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

cin >> a[i];

}

cin >> M;

vector<int> b(M);

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

cin >> b[i];

}

int common\_count = 0;

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

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

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

++common\_count;

break;

}

}

}

vector<int> combined;

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

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

}

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

combined.push\_back(b[i]);

}

vector<int> unique\_elements;

for (size\_t i = 0; i < combined.size(); ++i) {

bool is\_unique = true;

for (size\_t j = 0; j < unique\_elements.size(); ++j) {

if (combined[i] == unique\_elements[j]) {

is\_unique = false;

break;

}

}

if (is\_unique) {

unique\_elements.push\_back(combined[i]);

}

}

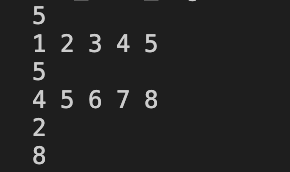
cout << common\_count << endl;

cout << unique\_elements.size() << endl;

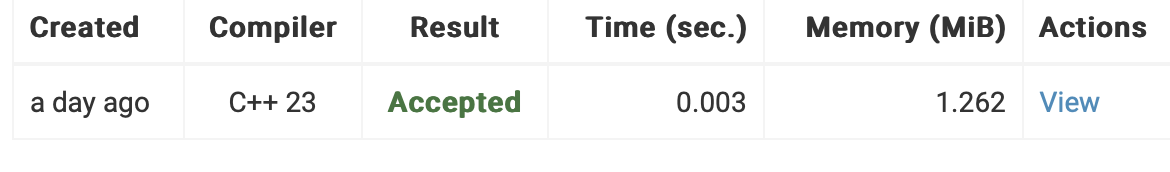
return 0;

}

Результат:

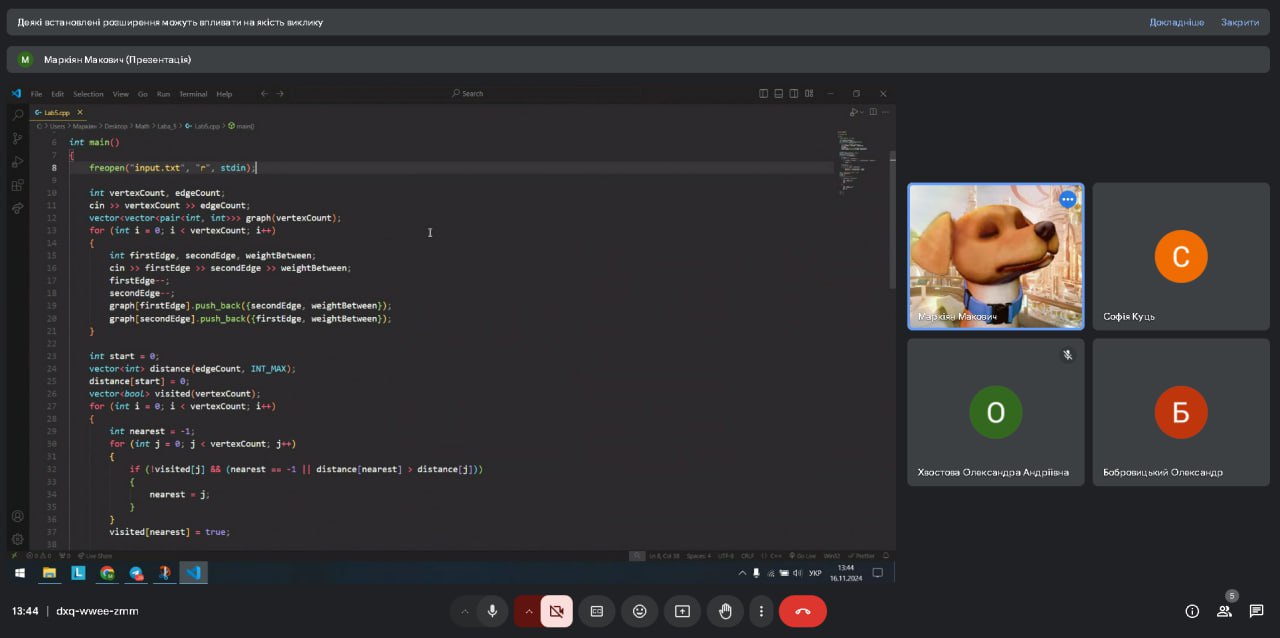


Час виконання : 45 хв



**Кооперація з командою**

Скрін з першої зустрічі по обговоренню завдань :



Скрін з другої зустрічі по обговоренню завдань:

