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-14

Мруць Остап Мар’янович

# **Тема роботи:**

Прості структури даних у С++. Ознайомлення з одновимірними та двовимірними масивами. Знайомство з базовими алгоритми обробки даних

# **Мета роботи:**

Ознайомитися з одновимірними та двовимірними у С++. Опрацювати та застосувати алгоритми обробки для масивів.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Одновимірні масиви.
* Тема №2: Двовимірні масиви.
* Тема №3: Алгоритми обробки.

1. Індивідуальний план опрацювання теорії:

* Тема: Одновимірні масиви.
  + Джерела Інформації:

<https://acode.com.ua/urok-81-bagatovymirni-masyvy/>

<https://www.w3schools.com/cpp/cpp_function_overloading.asp>

<https://www.w3schools.com/cpp/cpp_function_param.asp>

<https://www.w3schools.com/cpp/cpp_functions_recursion.asp>

<https://www.guru99.com/cpp-dynamic-array.html>

* + Що опрацьовано:
    - Одновимірні, двовимірні масиви, параметри функцій, динамічні масиви, перевантаження функції, рекурсії.
  + Статус: Ознайомлений

# **Виконання роботи:**

**VNS LAB4**

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

Завдання №1 VNS Lab4

19.

1) Реалізувати з використанням масиву однонаправлене кільце (перегляд

можливий зправа наліво, від першого елемента можна перейти до

останнього).

2) Роздрукувати отриманий масив, починаючи з К-ого елемента і до К+1.

3) Додати в кільце перший і останній елементи.

4) Знищити з кільця непарні елементи.

5) Роздрукувати отриманий масив, починаючи з К-ого елемента й до К+1.

## **2. Код програми:**

#include <iostream>

#include <vector>

using namespace std;

int main() {

vector<int> circular\_array = {1, 5, 12, 97, 18, 36, 77, 19, 56, 23};

cout << "Початкове кільце: ";

for (int i = 0; i < circular\_array.size(); ++i) {

cout << circular\_array[i] << " ";

}

cout << endl;

int K = 3;

cout << "Елементи від K до K+1: ";

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

cout << circular\_array[i % circular\_array.size()] << " ";

}

cout << endl;

circular\_array.push\_back(circular\_array[0]);

circular\_array.push\_back(circular\_array[circular\_array.size() - 2]);

vector<int> updated\_array;

for (int num : circular\_array) {

if (num % 2 == 0) {

updated\_array.push\_back(num);

}

}

cout << "Оновлене кільце без непарних елементів: ";

for (int i = 0; i < updated\_array.size(); ++i) {

cout << updated\_array[i] << " ";

}

cout << endl;

cout << "Елементи від K до K+1 після змін: ";

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

cout << updated\_array[i % updated\_array.size()] << " ";

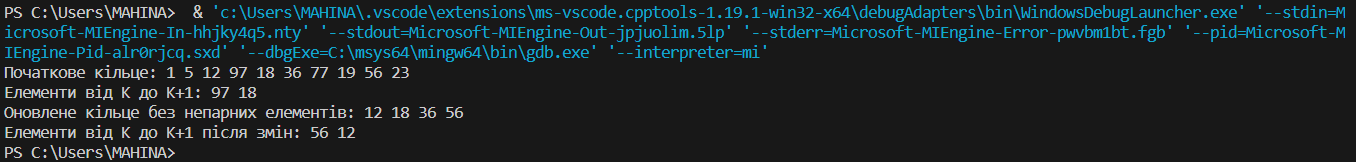
}

cout << endl;

return 0;

}

## **3. Результати виконання завдань та тестування:**



**VNS LAB5**

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

Завдання №2 VNS Lab5

19. Задано рядок з N^2 цифр. Встановити чи можна, розбивши рядок на

підстрічки довжиною N, записати їх у рядки двовимірного масиву N x N по

одній цифрі в одному елементі так, щоб вони в першому стовпці

розташувалися в порядку зростання.

## **2. Код програми:**

#include <iostream>

#include <cmath>

using namespace std;

bool sort(int arr[], int size) {

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

if (arr[i] > arr[i + 1]) {

return false;

}

}

return true;

}

void M(int arr[], int size) {

int sqr = sqrt(size);

if (!sort(arr, size)) {

cout << "Error" << endl;

return;

}

int matrix[sqr][sqr];

int index = 0;

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

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

matrix[j][i] = arr[index++];

}

}

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

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

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

}

cout << endl;

}

}

int main() {

int arr\_ordered[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};

int size\_ordered = sizeof(arr\_ordered) / sizeof(arr\_ordered[0]);

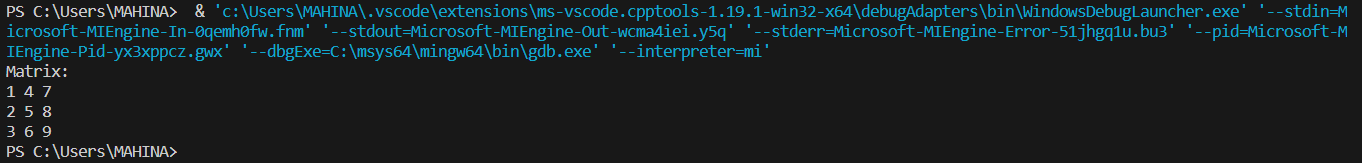
cout << "Matrix:" << endl;

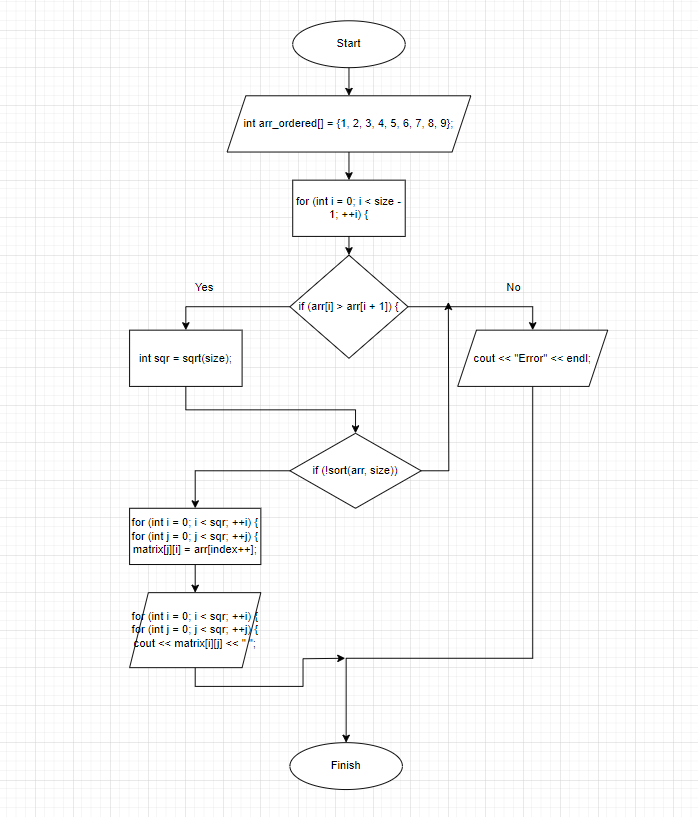
M(arr\_ordered, size\_ordered);

return 0;

}

## **3. Результати виконання завдань:**





**Algotester Lab3V2**

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

Завдання 3 Algotester Lab 3v2

**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.**

## **2. Код програми:**

## #include <iostream>

## #include <unordered\_set>

## using namespace std;

## void readArray(unordered\_set<int>& arr) {

## int n;

## cin >> n;

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

## int element;

## cin >> element;

## arr.insert(element);

## }

## }

## void compareArrays() {

## unordered\_set<int> arr, arr1;

## 

## readArray(arr);

## readArray(arr1);

## int similar = 0;

## for (int element : arr) {

## if (arr1.count(element)) {

## similar++;

## }

## }

## int totalUnique = arr.size() + arr1.size() - similar;

## cout << similar << endl << totalUnique << endl;

## }

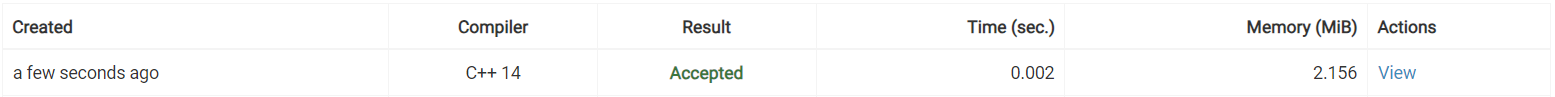
## int main() {

## compareArrays();

## return 0;

## }

## **3. Результати виконання завдання:**



**Practice Task # 4**

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

Завдання №4 Class-Practice Task # 4

# **Перевірка чи слово або число є паліндромом**

# **Задача**

## **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»).**

## **Мета Задачі**

Навчитися користуватися механізмами перевантаження функції та використовувати рекурсію для вирішення задач обчислення.

### 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. Перетворити ціле число на рядок і використовуйте рядкову версію функції, щоб перевірити, чи це паліндром.

## **2. Код програми:**

## **#include <iostream>**

## **#include <string>**

## **using namespace std;**

## **bool isPalindrome(const string &word) {**

## **int length = word.length();**

## **for (int i = 0; i < length / 2; ++i) {**

## **if (word[i] != word[length - 1 - i]) {**

## **return false;**

## **}**

## **}**

## **return true;**

## **}**

## **int main() {**

## **string inputWord;**

## **cout << "Введіть слово: ";**

## **cin >> inputWord;**

## **if (isPalindrome(inputWord)) {**

## **cout << "паліндром." << endl;**

## **} else {**

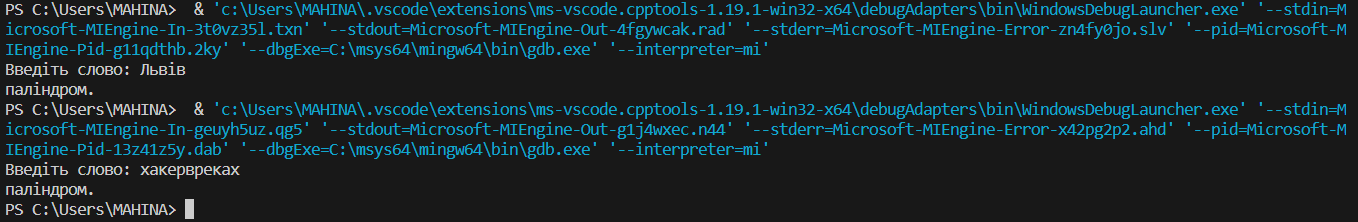
## **cout << "не паліндром." << endl;**

## **}**

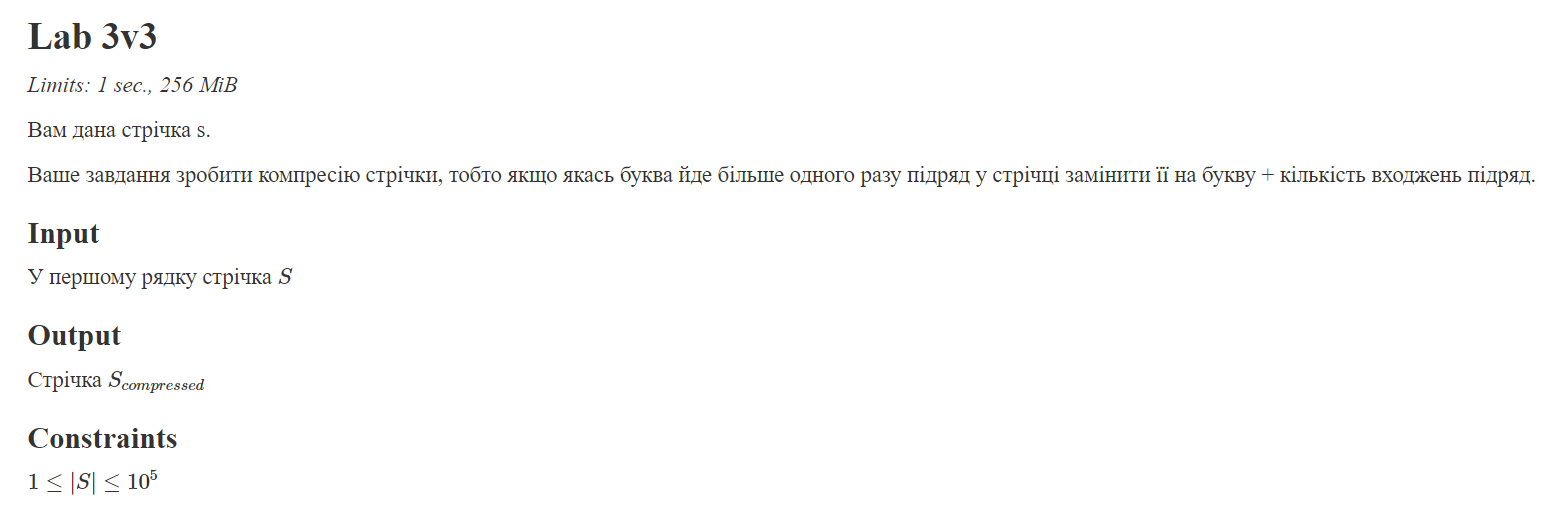
## **return 0;**

## **}**

## **3 . Результати виконання:**



**Self-Practice(Algotester Lab3V3)**



## **2. Код програми:**

#include <iostream>

using namespace std;

void readArray(int arr[], int& n) {

cin >> n;

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

cin >> arr[i];

}

}

int countSimilar(int arr[], int n, int arr1[], int n1) {

int similar = 0;

for (int i = 0; i < n1; i++) {

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

similar += (arr1[i] == arr[j]);

}

}

return similar;

}

void compareArrays() {

int arr[100], arr1[100];

int n, n1;

readArray(arr, n);

readArray(arr1, n1);

int similar = countSimilar(arr, n, arr1, n1);

int unique = n + n1 - similar;

cout << similar << endl << unique << endl;

}

int main() {

compareArrays();

return 0;

}

## **3. Результати виконання завдання:**



**Algotester Lab2V2**

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

Завдання 3 Algotester Lab 2v2

У вас є масив

r розміром N. Також вам дано 3 цілих числа.

Спочатку ви маєте видалити з масиву ці 3 числа, які вам дані. Після цього перетворити цей масив у масив сум, розміром

N new − 1

(розмір нового масиву після видалення елементів), який буде відображати суми сусідніх елементів нового масиву.

Далі необхідно вивести масив сум на екран.

## **2. Код програми:**

**#include <iostream>**

**#include <vector>**

**using namespace std;**

**int main() {**

**int N;**

**cin >> N;**

**vector<int> r(N);**

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

**cin >> r[i];**

**}**

**int a, b, c;**

**cin >> a >> b >> c;**

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

**if (r[i] == a || r[i] == b || r[i] == c) {**

**r.erase(r.begin() + i);**

**N--;**

**i--;**

**}**

**}**

**vector<int> sums;**

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

**sums.push\_back(r[i] + r[i + 1]);**

**}**

**cout << sums.size() << endl;**

**for (int sum : sums) {**

**cout << sum << " ";**

**}**

**cout << endl;**

**return 0;**

**}**

## **Результати виконання завдання:**

****

## **Кооперація з командою:**



# **Висновки:**

Отже, в ході роботи над епіком 4 я отримав знання про те, що таке структури даних у С++. Ознайомився з одновимірними та двовимірними масивами та з базовими алгоритми обробки даних

Pull-Request: