МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів

Кафедра систем управління літальних апаратів

**Лабораторна робота № 7**

з дисципліни «Алгоритмізація та програмування»

на тему «Реалізація алгоритмів обробки одновимірних масивів мовою С ++

»

ХАІ.301. 175. 318.14 ЛР

Виконав студент гр. \_\_\_\_\_\_318\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_ Катерина ОВЧАРЕНКО*\_\_\_\_\_\_\_*

(підпис, дата) (П.І.Б.)

Перевірив

\_\_\_\_\_\_\_\_\_ к.т.н., доц. Олена  ГАВРИЛЕНКО

(підпис, дата) (П.І.Б.)

2023

# МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ представлення двовимірних масивів

(матриць) на мові С ++ і реалізувати оголошення, введення з консолі,

обробку і виведення в консоль матриць на мові C ++ в середовищі Visual

Studio.

# ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на аналіз і виведення елементів матриці.

Введення і виведення даних здійснити в командному вікні. Варіанти

завдань Matrix 36.

Завдання 2. Перетворити матрицю відповідно до свого варіанту завдання

Matrix 59, розмір матриці і його елементи ввести з консолі. Вивести

результати у консоль.

# ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Matrix 36

Вхідні дані (ім’я, опис, тип, обмеження):

matrix - Опис: Матриця, що містить цілі числа, яку необхідно перевірити

на наявність схожих рядків. Тип: Вектор векторів цілих чисел Обмеження:

Матриця може бути порожньою або мати різні розміри рядків. Всі е

лементи матриці повинні бути цілими числами.

Вихідні дані (ім’я, опис, тип):  
 count - Опис: Кількість рядків, які схожі на перший рядок матриці.

Тип:Ціла змінна

uniqueRows - Опис: Множина унікальних рядків, що схожі на перший рядок. Тип: Множина рядків

Алгоритм вирішення:  
 1.Перевірка, чи не є матриця порожньою. Якщо так, повертаємо 0 як результат.

2.Ініціалізуємо лічильник схожих рядків та множину для збереження унікальних рядків .

3.Конвертуємо перший рядок матриці в рядок символів для подальшого порівняння.

4.Проходимо по решті рядків матриці і порівнюємо їх із першим рядком.

5.Виводимо унікальні схожі рядки.

6.Повертаємо кількість схожих рядків.

Лістинг коду вирішення задачі Matrix 36 наведено в додатку А (стор.)

Екран роботи програми показаний на рис. .

Завдання 2.

Вирішення задачі Array 65

Вхідні дані (ім’я, опис, тип, обмеження):

matrix (матриця): Матриця цілих чисел, яку потрібно дзеркально

відобразити. Тип: Вектор векторів цілих чисел. Обмеження: немає.

Вихідні дані (ім’я, опис, тип):  
 Алгоритм вирішення:

1.Перевірка, чи не є матриця порожньою. Якщо так, функція завершується.

2.Отримання кількості рядків (m) та кількості стовпців (n) у матриці.

3.Проходимо по першій половині рядків матриці (до m / 2):

# ВИСНОВКИ

Після вивчення та використання функції mirrorMatrix було закріплено розуміння роботи з двовимірними векторами та операціями над ними в мові програмування C++.Програмний код демонструє ефективну реалізацію операції дзеркального відображення матриці безпосередньо у C++, що сприяє поглибленню навичок у розробці програмного забезпечення.

ДОДАТОК А

Лістинг коду програми

*#include <iostream>*

*#include <vector>*

*#include <unordered\_set>*

*using namespace std;*

*int countSimilarRows(const vector<vector<int>>& matrix) {*

*if (matrix.empty() || matrix[0].empty()) {*

*return 0; // Перевірка на порожню матрицю*

*}*

*int m = matrix.size();*

*int n = matrix[0].size();*

*int count = 0; // Лічильник схожих рядків*

*unordered\_set<string> uniqueRows; // Множина для збереження унікальних рядків*

*// Конвертуємо перший рядок в рядок символів для порівняння*

*string firstRow = "";*

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

*firstRow += to\_string(matrix[0][j]) + ",";*

*}*

*// Перевіряємо решту рядків на схожість з першим*

*for (int i = 1; i < m; ++i) {*

*string currentRow = "";*

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

*currentRow += to\_string(matrix[i][j]) + ",";*

*}*

*if (currentRow == firstRow) {*

*// Якщо рядок схожий, збільшуємо лічильник та додаємо його до множини*

*++count;*

*uniqueRows.insert(currentRow);*

*}*

*}*

*cout << "Схожі рядки: ";*

*for (const string& row : uniqueRows) {*

*cout << row << " ";*

*}*

*cout << endl;*

*return count;*

*}*

*int main() {*

*// Приклад використання*

*vector<vector<int>> matrix = {*

*{1, 2, 3},*

*{4, 5, 6},*

*{1, 2, 3},*

*{7, 8, 9}*

*};*

*int result = countSimilarRows(matrix);*

*cout << "Кількість схожих рядків: " << result << endl;*

*return 0;*

*#include <iostream>*

*#include <vector>*

*using namespace std;*

*void mirrorMatrix(vector<vector<int>>& matrix) {*

*if (matrix.empty() || matrix[0].empty()) {*

*return; // Перевірка на порожню матрицю*

*}*

*int m = matrix.size();*

*int n = matrix[0].size();*

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

*swap(matrix[i], matrix[m - 1 - i]); // Поміняти рядки місцями*

*}*

*}*

*int main() {*

*// Приклад використання*

*vector<vector<int>> matrix = {*

*{1, 2, 3},*

*{4, 5, 6},*

*{7, 8, 9}*

*};*

*cout << "Оригінальна матриця:" << endl;*

*for (const auto& row : matrix) {*

*for (int num : row) {*

*cout << num << " ";*

*}*

*cout << endl;*

*}*

*mirrorMatrix(matrix);*

*cout << "Дзеркально відображена матриця:" << endl;*

*for (const auto& row : matrix) {*

*for (int num : row) {*

*cout << num << " ";*

*}*

*cout << endl;*

*}*

*return 0;*

*}*

ДОДАТОК Б

Скрін-шоти вікна виконання програми

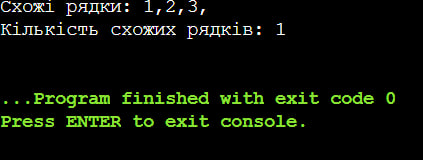


Рисунок Б.1 – Екран виконання програми для вирішення завдання   
Matrix 36

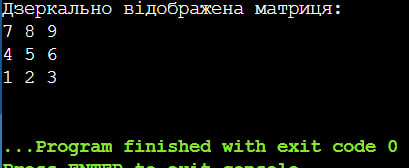


Рисунок Б.2 – Екран виконання програми для вирішення завдання   
Matrix 59