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 двовимірних масивів на мові С ++"

ХАІ.301. 141. 319a. 16 ЛР

Виконав студент гр. \_\_\_\_\_\_319a\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Моісеєнко Євген*\_\_\_\_\_\_\_*

(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.

Matrix36 . Дана цілочисельна матриця розміру M × N, елементи якої можуть

набувати значень від 0 до 100. Різні рядки матриці назвемо схожими, якщо

збігаються безлічі чисел, що зустрічаються в цих рядках. Знайти кількість

рядків, схожих на перший рядок цієї матриці.

# ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Matrix36

Вхідні дані:

1)matrix,Тип: vector<vector<int>>,Опис: Матриця, що містить цілі числа.

Обмеження:

Матриця не може бути порожньою.

Кількість рядків та стовпчиків матриці має бути більше 0.

Всі елементи матриці мають бути цілими числами.

2)M,Тип: int,Опис: Кількість рядків матриці.

Обмеження:

M має бути більше 0.

3)N,Тип: int,Опис: Кількість стовпчиків матриці

Обмеження:

N має бути більше 0.

Вихідні дані:

similarRowsCount,Тип: int,Опис: Кількість рядків, схожих на перший рядок матриці.

Алгоритм вирішення показано на рис.1

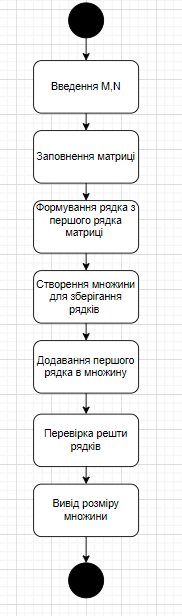


Рисунок 1 – Matrix36

Лістинг коду вирішення задачі Matrix36 наведено в дод. А (стор. 7-10).

Екран роботи програми показаний на рис. Б.1.

Завдання 2.

Вирішення задачі Matrix75

Вхідні дані:

1)m, n,Тип: int,Опис: Кількість рядків і стовпчиків у матриці відповідно.

Обмеження:

Ціле число більше 0

2)matrix,Тип: std::vector<std::vector<int>>.

Обмеження:

Немає обмежень на розмір матриці або значення її елементів.

Вихідні дані:

matrix,Тип: std::vector<std::vector<int>>,Опис: Матриця з змінними знаками локальних максимумів.

Алгоритм вирішення показано на рис.2

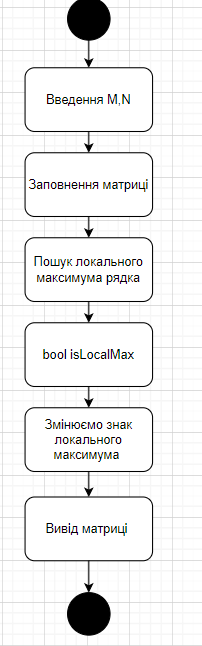


Рисунок 2 – Matrix75

Лістинг коду вирішення задачі Matrix75 наведено в дод. А (стор.7-10).

Екран роботи програми показаний на рис. Б.2.

# ВИСНОВКИ

У результаті вивчення теоретичного матеріалу з основ представлення двовимірних масивів (матриць) на мові C++ та їх практичної реалізації в середовищі Visual Studio можна зробити висновок про успішне засвоєння концепцій створення, введення, обробки та виведення матриць у програмуванні. Під час цього процесу було набуто навичок роботи з масивами, управління пам'яттю, введенням/виведенням даних у консольному інтерфейсі, а також розуміння основних принципів мови програмування С++.

ДОДАТОК А

Лістинг коду програми

*#include <iostream>*

*#include <vector>*

*#include <unordered\_set>*

*using namespace std;*

*// Глобальна змінна для матриці*

*vector<vector<int>> matrix;*

*// Функція для зчитування розміру матриці та її елементів*

*void readMatrix() {*

*int M, N;*

*cout << "Введіть розмірність матриці M та N: ";*

*cin >> M >> N;*

*// Заповнюємо матрицю елементами*

*matrix.resize(M, vector<int>(N));*

*cout << "Введіть елементи матриці:" << endl;*

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

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

*cin >> matrix[i][j];*

*}*

*}*

*}*

*// Функція, яка знаходить кількість рядків, схожих на перший рядок матриці*

*int findSimilarRows() {*

*int rows = matrix.size();*

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

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

*// Формуємо рядок з елементів першого рядка матриці*

*string firstRowString;*

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

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

*}*

*// Додаємо перший рядок в множину схожих рядків*

*similarRows.insert(firstRowString);*

*// Ідемо по решті рядків і додаємо ті, які схожі на перший рядок у множину*

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

*string currentRowString;*

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

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

*}*

*similarRows.insert(currentRowString);*

*}*

*// Повертаємо кількість унікальних схожих рядків*

*return similarRows.size();*

*}*

*// Функція для перевірки чи елемент (i, j) є локальним максимумом в матриці*

*bool isLocalMax(const std::vector<std::vector<int>>& matrix, int i, int j) {*

*int m = matrix.size();*

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

*// Перевіряємо всі сусідні елементи*

*for (int x = -1; x <= 1; ++x) {*

*for (int y = -1; y <= 1; ++y) {*

*int ni = i + x;*

*int nj = j + y;*

*if (ni >= 0 && ni < m && nj >= 0 && nj < n && !(x == 0 && y == 0)) {*

*if (matrix[ni][nj] >= matrix[i][j]) {*

*return false;*

*}*

*}*

*}*

*}*

*return true;*

*}*

*// Функція для зміни знаку всіх локальних максимумів у матриці*

*void changeLocalMax(std::vector<std::vector<int>>& matrix) {*

*int m = matrix.size();*

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

*// Перебираємо всі елементи матриці*

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

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

*if (isLocalMax(matrix, i, j)) {*

*// Якщо (i, j) - локальний максимум, змінюємо його знак*

*matrix[i][j] \*= -1;*

*}*

*}*

*}*

*}*

*// Функція для виведення матриці на екран*

*void printMatrix(const std::vector<std::vector<int>>& matrix) {*

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

*for (int element : row) {*

*std::cout << element << " ";*

*}*

*std::cout << std::endl;*

*}*

*}*

*// Основна функція*

*int main() {*

*int choice;*

*bool exit = false;*

*do {*

*cout << "Виберіть завдання:" << endl;*

*cout << "1. Знайти кількість рядків, схожих на перший рядок матриці." << endl;*

*cout << "2. Змінити знак всіх локальних максимумів у матриці." << endl;*

*cout << "3. Завершити програму." << endl;*

*cout << "Ваш вибір: ";*

*cin >> choice;*

*switch (choice) {*

*case 1: {*

*// Зчитуємо матрицю*

*readMatrix();*

*// Знаходимо кількість рядків, схожих на перший рядок*

*int similarRowsCount = findSimilarRows();*

*cout << "Кількість рядків, схожих на перший рядок: " << similarRowsCount << endl;*

*break;*

*}*

*case 2: {*

*int m, n;*

*cout << "Введіть розмір матриці (MxN): ";*

*cin >> m >> n;*

*// Читаємо елементи матриці з консолі*

*vector<vector<int>> matrix(m, vector<int>(n));*

*cout << "Введіть елементи матриці:\n";*

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

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

*cin >> matrix[i][j];*

*}*

*}*

*// Викликаємо функцію для зміни знаку локальних максимумів*

*changeLocalMax(matrix);*

*// Виводимо результати*

*cout << "Результат:\n";*

*printMatrix(matrix);*

*break;*

*}*

*case 3:*

*exit = true;*

*break;*

*default:*

*cout << "Неправильний вибір!" << endl;*

*}*

*} while (!exit);*

*return 0;*

*}*

ДОДАТОК Б

Скрін-шоти вікна виконання програми

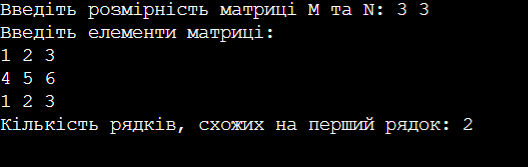


Рисунок Б.1 – Екран виконання програми для вирішення завдання   
Matrix36

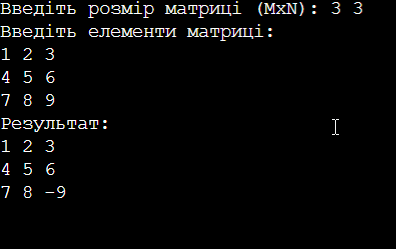


Рисунок Б.2 – Екран виконання програми для вирішення завдання   
Matrix75