Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 9 з дисципліни

«Алгоритми та структури даних.

Основи алгоритмізації»

«Дослідження алгоритмів обходу масивів»

Варіант 10

Виконав студент ІП-12 Горобець Олексій Сергійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 202 1

**Лабораторна робота 8**

**Дослідження алгоритмів обходу масивів**

**Мета** – дослідити алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

**Варіант 10**

***Задача 1.10***.

Задано матрицю дійсних чисел A[m,n], ініціалізувати матрицю обходом по стовбцях. У кожному стовпчику матриці знайти останній максимальний елемент і його місцезнаходження. Обміняти знайдене значення Х з елементом першого стовбця

***Постановка задачі*.** Результатом є позиція останнього максимального елементу стовпця матриці, та матриця у якій останній максимальний елемент стовпця обміняний з елементом першого стовпця. Для визначення цієї матриці нам необхідно задані розміри матриці, сама матриця. Інших початкових даних для розв’язку не потрібно.

***Математична побудова.*** Складемо таблицю змінних.

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| **Кількість рядків матриці m** | Цілий | m | Початкове дане |
| **Кількість стовпців матриці n** | Цілий | n | Початкове дане |
| **Двовимірний масив matrix** | Індексований | matrix | Початкове дане |
| **Максимальний елемент у стовпці матриці max** | Дійсний | max | Проміжне значення |
| **Позиція по рядках макс елементу у стовпці indexI** | Цілий | indexI | Результат |
| **Позиція по стовпцях макс елементу у стовпці maxJ** | Цілий | maxJ | Результат |
| **Результуючий максимальний елементу у стовпці resMax** | Дійсний | resMax | Результат |
| **Тимчасова змінна temp** | Дійсний | temp | Проміжне значення |

Таким чином, математичне формулювання задачі зводиться до ініцілізування матриці по стовпцям, потім розрахування максимальних елементів та їх позицій у стовпцях матриці. Далі необхідно обміняти останній максимальний елемент стовпця з елементом першого стовбця. Задля покращення читабельності псевдокоду введемо таку заміну **randInput** =rand%201-100, тобто **randInput** – генерує випадковім чином числа в діапазоні від 100 до -100 включно.

***Розв’язання:***

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

***Крок 1.*** Визначимо основні дії.

***Крок 2.*** Деталізуємо ініціалізації початкових змінних ***n, m, matrix***

***Крок 3.*** Деталізуємо пошук останніх максимальних елементів стовпців матриці

***Крок 4.*** Деталізуємо пошук позиції максимального елементу стовпця

***Крок 5.*** Деталізуємо обмін останнього максимального елементу стовпця з елементом першого стовбця

***Псевдокод***

*крок 1*

**початок**

Ініціалізації початкових змінних ***n, m, matrix***

Деталізуємо дію пошуку останніх максимальних елементів стовпців матриці Деталізуємо дію пошуку позиції максимального елементу стовпця

Деталізуємо дію обміну останнього максимального елементу стовпця з елементом першого стовбця

**Кінець**

*крок 2*

**початок**

**Ввід** m, n

**randMatrixInput(matrix, m, n)**

Деталізуємо дію пошуку останніх максимальних елементів стовпців матриці Деталізуємо дію пошуку позиції максимального елементу стовпця

Деталізуємо дію обміну останнього максимального елементу стовпця з елементом першого стовбця

**Кінець**

**початок функції randMatrixInput (matrix, m, n)**

**повторити**

**для j від 0 до n**

**повторити**

**для і від 0 до m**

matrix[i][j]:=randInput

**все повторити**

**все повторити**

**кінець функції randMatrixInput**

*крок 3*

**Ввід** m, n

**randMatrixInput(matrix, m, n)**

**findColumnMaxAndSwap(matrix, m, n)**

**Кінець**

**початок функції randMatrixInput (matrix, m, n)**

**повторити**

**для j від 0 до n**

**повторити**

**для і від 0 до m**

matrix[i][j]:=randInput

**все повторити**

**все повторити**

**кінець функції randMatrixInput**

**початок функції findColumnMaxAndSwap(matrix, m, n)**

**повторити**

**для j від 0 до n**

indexI:= 0

max:= matrix[0][j]

**повторити**

**для і від 0 до m**

**якщо** matrix[i][j]>= max

**то**

max:=matrix[i][j]

indexI:=i

**все якщо**

**все повторити**

Деталізуємо дію пошуку позиції максимального елементу стовпця

Деталізуємо дію обміну останнього максимального елементу стовпця з елементом першого стовбця

**все повторити**

**кінець функції findColumnMaxAndSwap**

Крок 4

**Ввід** m, n

**randMatrixInput(matrix, m, n)**

**findColumnMaxAndSwap(matrix, m, n)**

**Кінець**

**початок функції randMatrixInput (matrix, m, n)**

**повторити**

**для j від 0 до n**

**повторити**

**для і від 0 до m**

matrix[i][j]:=randInput

**все повторити**

**все повторити**

**кінець функції randMatrixInput**

**початок функції findColumnMaxAndSwap(matrix, m, n)**

**повторити**

**для j від 0 до n**

indexI:= 0

max:= matrix[0][j]

**повторити**

**для і від 0 до m**

**якщо** matrix[i][j]>= max

**то**

max:=matrix[i][j]

indexI:=i

**все якщо**

**все повторити**

maxJ:=j

resMax:=max

**Вивід** resMax, indexI, maxJ

Деталізуємо дію обміну останнього максимального елементу стовпця з елементом першого стовбця

**все повторити**

**кінець функції findColumnMaxAndSwap**

**Крок 5.**

**Ввід** m, n

**randMatrixInput(matrix, m, n)**

**findColumnMaxAndSwap(matrix, m, n)**

**Кінець**

**початок функції randMatrixInput (matrix, m, n)**

**повторити**

**для j від 0 до n**

**повторити**

**для і від 0 до m**

matrix[i][j]:=randInput

**все повторити**

**все повторити**

**кінець функції randMatrixInput**

**початок функції findColumnMaxAndSwap (matrix, m, n)**

**повторити**

**для j від 0 до n**

indexI:= 0

max:= matrix[0][j]

**повторити**

**для і від 0 до m**

**якщо** matrix[i][j]>= max

**то**

max:=matrix[i][j]

indexI:=i

**все якщо**

**все повторити**

maxJ:=j

resMax:=max

**Вивід** resMax, indexI, maxJ

temp:= matrix[indexI][j]

matrix[indexI][j] = matrix[indexI][0];

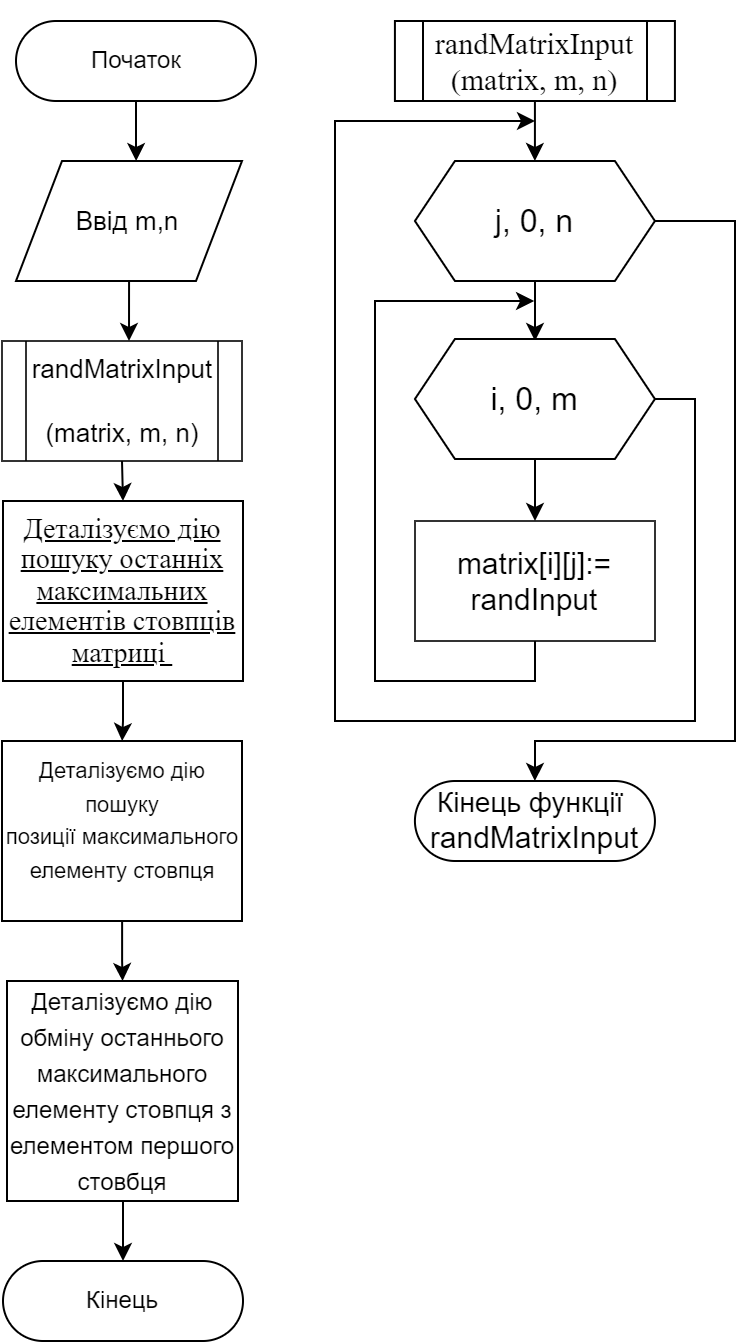
matrix[indexI][0] = temp;

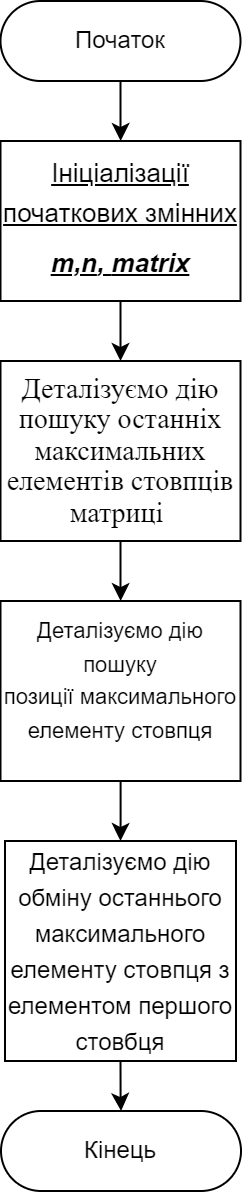
**все повторити**

**кінець функції findColumnMaxAndSwap**

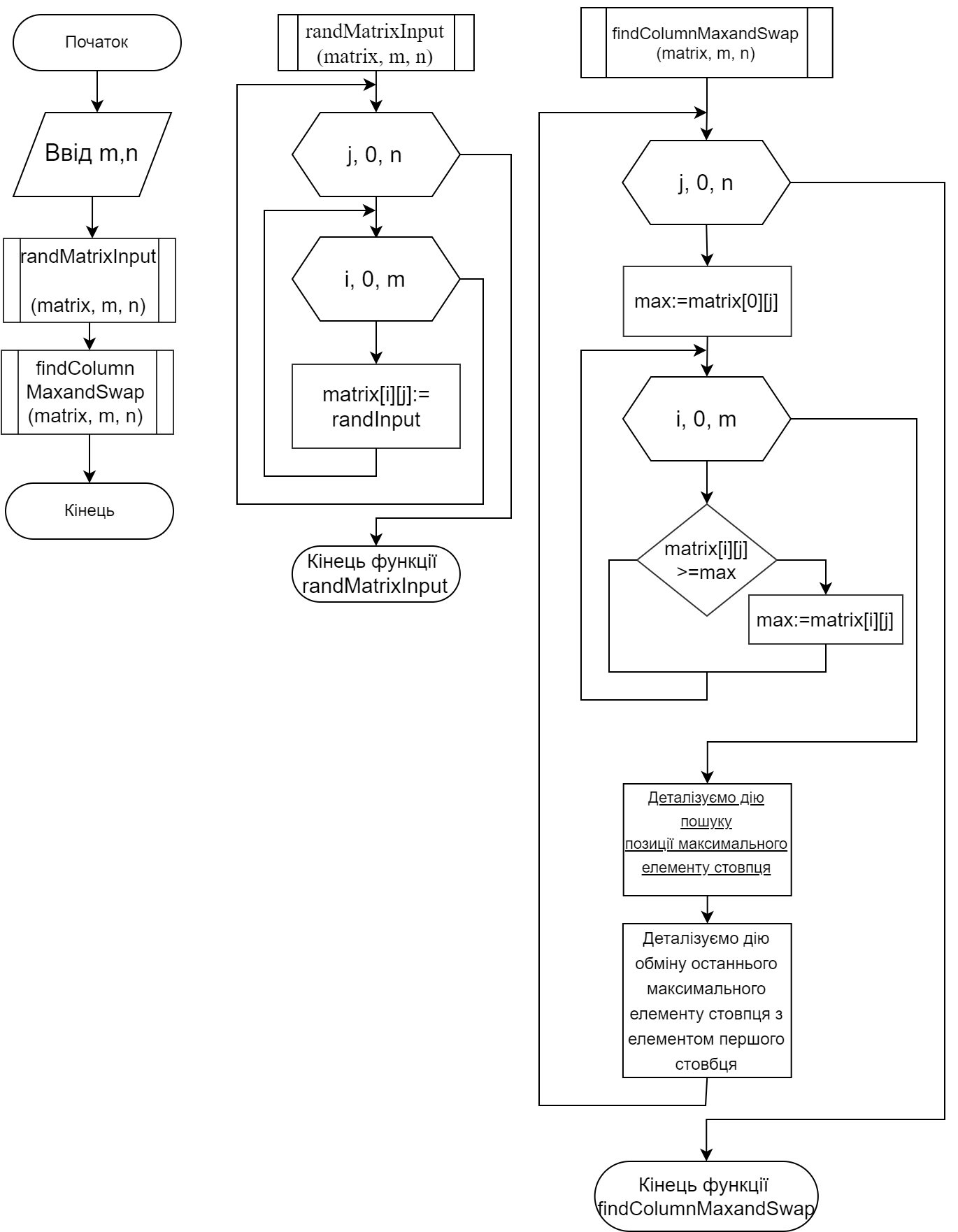
***Блок-схема***

***Крок 1 Крок 2***

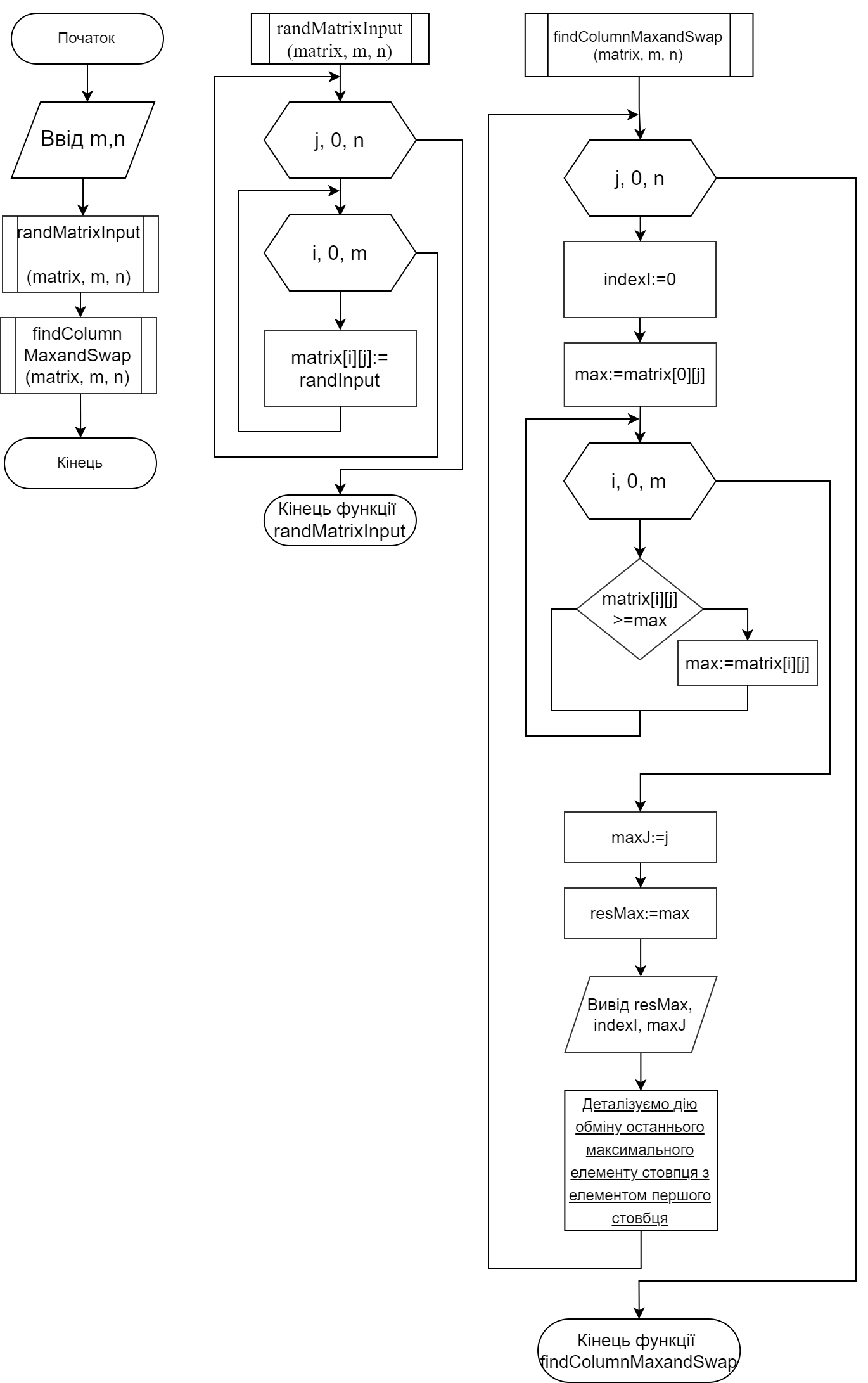
**

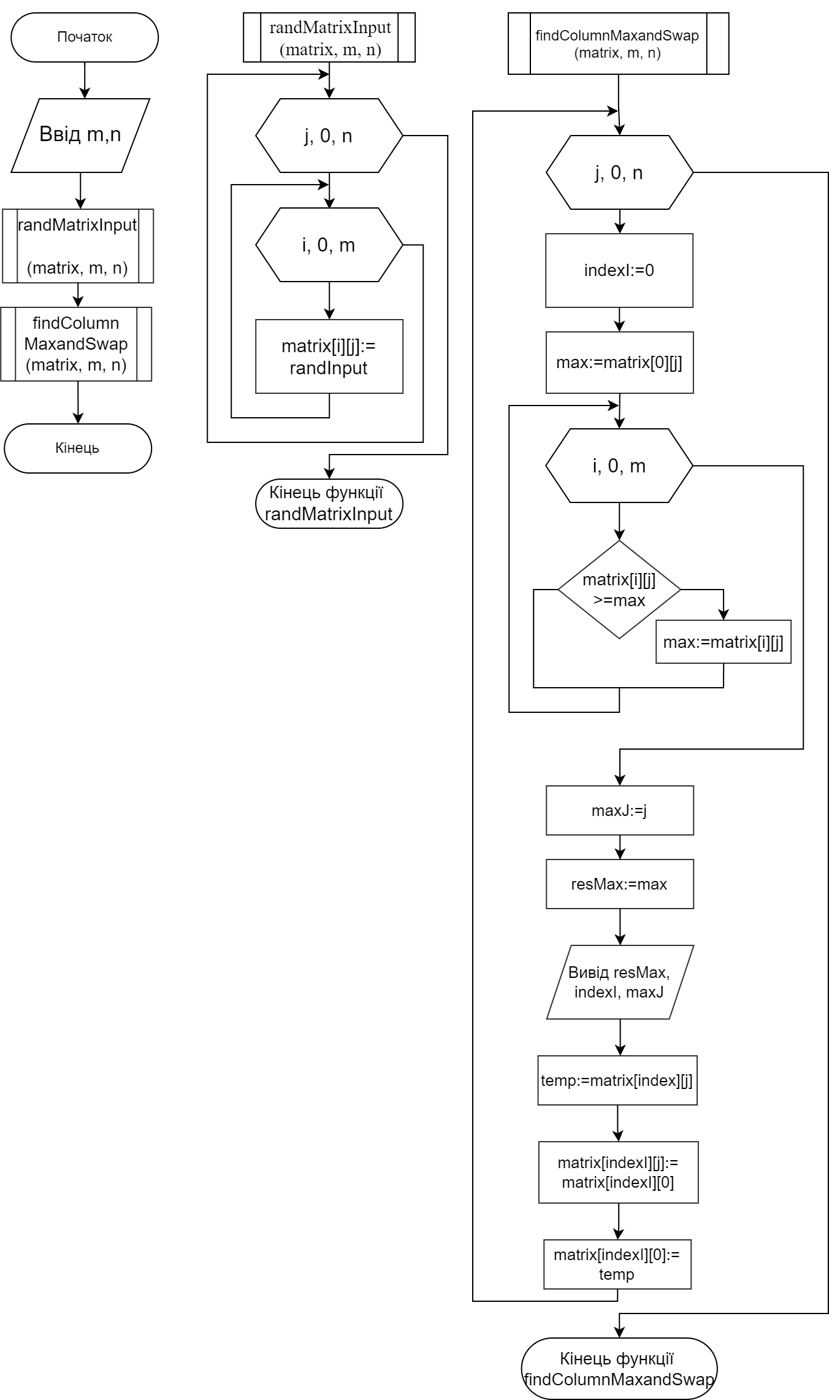
**

***Крок 3***

******

***Крок 4***

******

***Крок 5***

***Код програми c++***

#include <iostream>

#include <ctime>

#include <iomanip>

using namespace std;

int intInput(string);

double\*\* createMatrix(int, int);

void displayMatrix(double\*\*, int, int, string);

void randMatrixInput(double\*\*, int, int);

void findColumnMaxAndSwap(double\*\*, int, int);

int main()

{

setlocale(LC\_ALL, "rus");

int m, n;

m = intInput("Enter m: ");

n = intInput("Enter n: ");

double\*\* matrix = createMatrix(m,n);

randMatrixInput(matrix, m, n);

displayMatrix(matrix, m, n, "\nInitial matrix:\n");

findColumnMaxAndSwap(matrix, m, n);

displayMatrix(matrix, m, n, "\nResult matrix:\n");

}

int intInput(string msg)

{

int n;

cout << msg;

cin >> n;

return n;

}

double\*\* createMatrix(int m, int n)

{

double\*\* matrix = new double\* [m];

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

matrix[i] = new double[n];

}

return matrix;

}

void displayMatrix(double\*\* matrix, int m, int n, string msg)

{

cout << msg;

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

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

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

}

cout << endl;

}cout << endl;

}

void randMatrixInput(double\*\* matrix, int m, int n)

{

srand(time(NULL));

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

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

matrix[i][j] = rand() % (201) - 100;

}

}

}

void findColumnMaxAndSwap(double\*\* matrix, int m, int n) {

int indexI;

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

indexI = 0;

double max = matrix[0][j];

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

if (matrix[i][j] >= max) {

max = matrix[i][j];

indexI = i;

}

}

int maxJ = j;

double resMax = max;

cout << "Max element of "<< maxJ <<" column: " << resMax << "\tPosition: " << indexI << " " << maxJ << endl;

double temp = matrix[indexI][j];

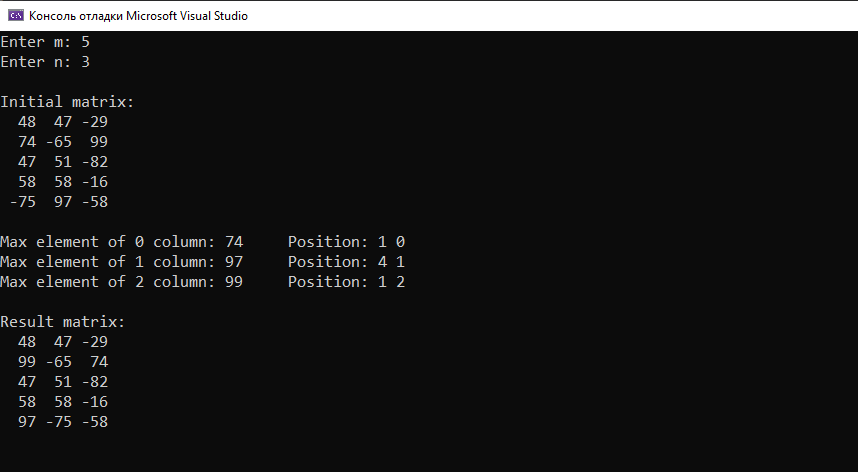
matrix[indexI][j] = matrix[indexI][0];

matrix[indexI][0] = temp;

}

}

***Тест програми***



***Висновок***

Протягом лабораторної роботи було досліджено алгоритми обходу масивів та набуто практичних навичок їх використання під час складання програмних специфікацій. Особливістю роботи було:

1. Розробка функції пошуку максимальних елементів у колонках матриці та їх позицій.
2. Розробка підпрограми для обміну максимальних елементів стовпців з елементами першого стовпця.