Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Катедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 9 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження алгоритмів обходу масивів»

Варіант 18

Виконав студент ІП-11 Лесів Владислав Ігорович

Перевірив Мартинова О.П.

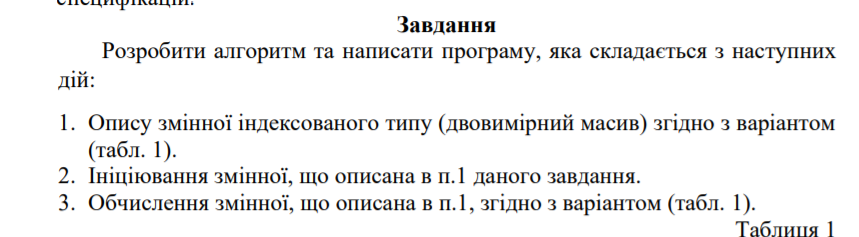
Київ 2021

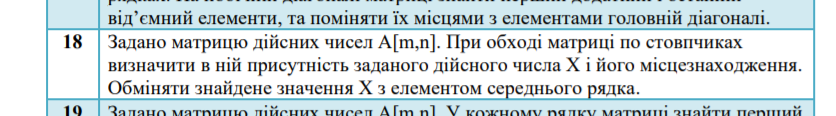
**Лабораторна робота 9**

**Дослідження алгоритмів обходу масивів**

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

**Варіант №18.**





**Постановка задачі.** Результатом розв’язку є матриця, утворена з заданої та змінена відповідно до умови: знайдене значення необхідно обміняти з елементом середнього рядка. Для визначення результату повинна бути задана матриця m\*n з дійсних елементів, а також елемент Х, який ми будемо шукати в матриці. Інших початкових даних для розв’язку не потрібно.

**Побудова математичної моделі.** Складемо таблицю імен змінних.

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім'я | Призначення |
| Кількість рядків матриці | Цілий | m | Початкове дане |
| Кількість стовпців матриці | Цілий | n | Початкове дане |
| Матриця | Дійсний | matr | Початкове дане, результат |
| Заданий елемент | Дійсний | x | Початкове дане |
| Значення елементу середнього рядка, з яким обмінюватимемо Х | Дійсний | tmp | Проміжний результат |
| Перевірка першого входження елементу Х на поточній позиції | Логічний | k | Проміжний результат |
| Індекс рядка для обходу змійкою | Цілий | h | Проміжний результат |

Математичне формулювання задачі зводиться до знаходження першого входження заданого елементу під час обходу за стовпцями змійкою і подальшого обміну його значення з елементом першого рядка матриці того ж стовпця, що і знайдений елемент.

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

*Крок 1.* Визначимо основні дії.

*Крок 2.* Деталізуємо дію ініціалізації заданої в умові задачі матриці за допомогою підпрограми.

*Крок 3.* Деталізуємо дію виведення матриці за допомогою підпрограми.

*Крок 4.* Деталізуємо дію обробки матриці відповідно до умови задачі за допомогою підпрограми.

*Псевдокод*

*крок 1*

**початок**

введення m, n, x

ініціалізація матриці

виведення матриці

обробка матриці

виведення матриці

**кінець**

*крок 2*

**початок**

введення m, n, x

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

inputMatrix(matr, m, n);

виведення матриці

обробка матриці

виведення матриці

**кінець**

**підпрограма** inputMatrix(\*\*matr, m, n)

srand(time(NULL))

**повторити**

**для** і **від** 0 **до** m-1

matr[i] = new double[n];

**все повторити**

**повторити**

**для** і **від** 0 **до** m-1

**повторити**

**для** j **від** 0 **до** n-1

matr[i][j] = rand() % 30 - 15;

**все повторити**

**все повторити**

**кінець підпрограми**

*крок 3*

**початок**

введення m, n, x

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

inputMatrix(matr, m, n);

outputMatrix(matr, m, n);

обробка матриці

outputMatrix(matr, m, n);

**кінець**

**підпрограма** inputMatrix(\*\*matr, m, n)

srand(time(NULL))

**повторити**

**для** і **від** 0 **до** m-1

matr[i] = new double[n];

**все повторити**

**повторити**

**для** і **від** 0 **до** m-1

**повторити**

**для** j **від** 0 **до** n-1

matr[i][j] = rand() % 30 - 15;

**все повторити**

**все повторити**

**кінець підпрограми**

**підпрограма** outputMatrix(\*\*matr, m, n)

**повторити**

**для** і **від** 0 **до** m-1

**повторити**

**для** j **від** 0 **до** n-1

виведення matr[i][j]

**все повторити**

виведення “\n”

**все повторити**

**кінець підпрограми**

*крок 4*

**початок**

введення m, n, x

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

inputMatrix(matr, m, n);

outputMatrix(matr, m, n);

processMatrix(matr, m, n, x);

outputMatrix(matr, m, n);

**кінець**

**підпрограма** inputMatrix(\*\*matr, m, n)

srand(time(NULL))

**повторити**

**для** і **від** 0 **до** m-1

matr[i] = new double[n];

**все повторити**

**повторити**

**для** і **від** 0 **до** m-1

**повторити**

**для** j **від** 0 **до** n-1

matr[i][j] = rand() % 30 - 15;

**все повторити**

**все повторити**

**кінець підпрограми**

**підпрограма** outputMatrix(\*\*matr, m, n)

**повторити**

**для** і **від** 0 **до** m-1

**повторити**

**для** j **від** 0 **до** n-1

виведення matr[i][j]

**все повторити**

виведення “\n”

**все повторити**

**кінець підпрограми**

**підпрограма** processMatrix(\*\*matr, m, n, x)

k:=0

**повторити**

**для** j **від** 0 **до** n-1

**повторити**

**для** i **від** 0 **до** m-1

**якщо** j%2==0

**то**

h=i;

**інакше**

h=m-i-1;

**все якщо**

**якщо (**matr[h][j] == x && k==0)

**то**

виведення «Елемент Х знайдено в», i+1, «рядку й», j+1, «стовпці»

tmp = matr[m/2][j];

matr[m/2][j] = matr[h][j];

matr[h][j] = tmp;

k = 1;

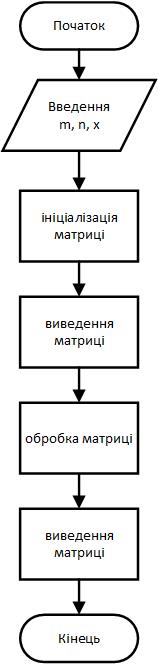
**все якщо**

**все повторити**

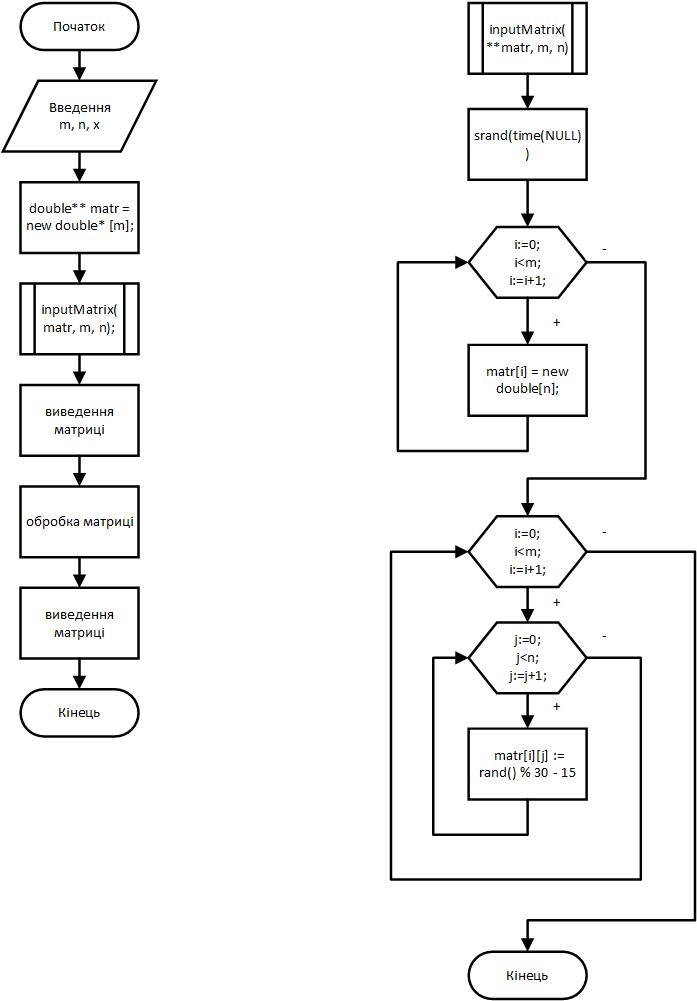
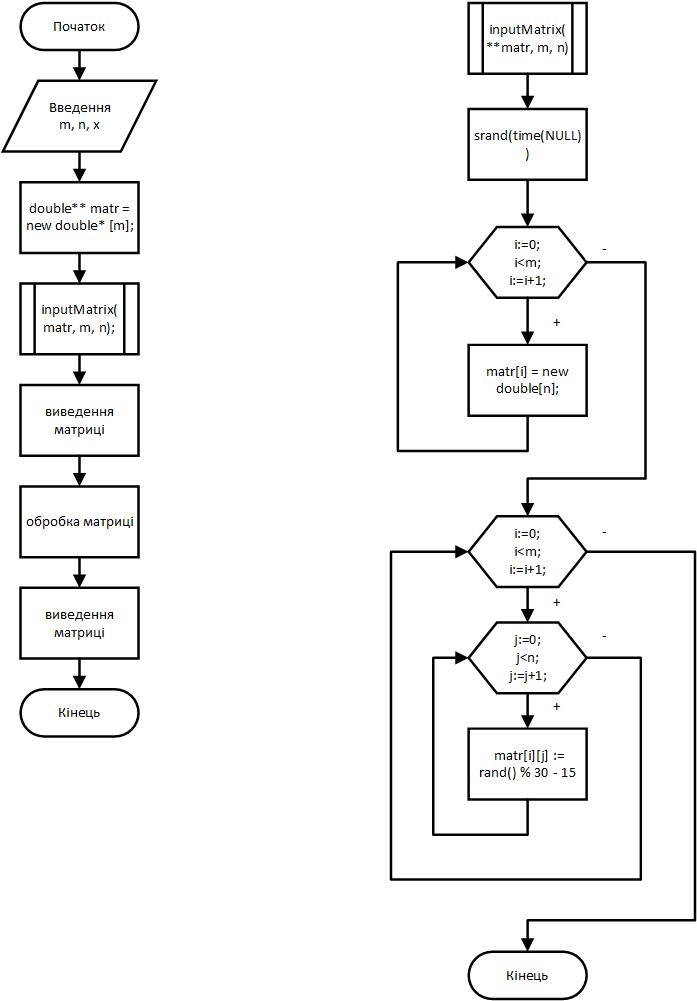
**все повторити**

*Блок-схема*

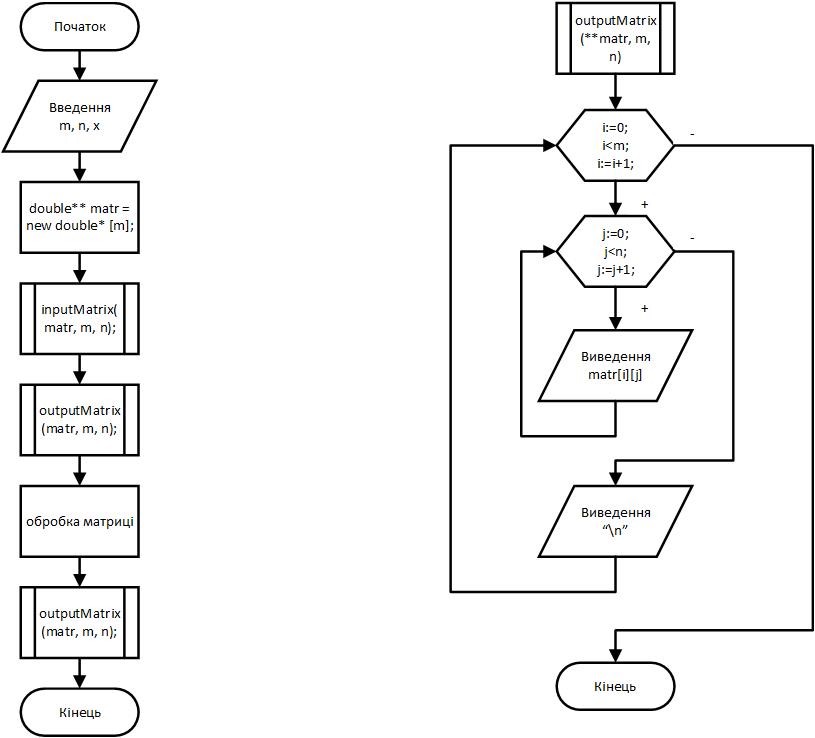
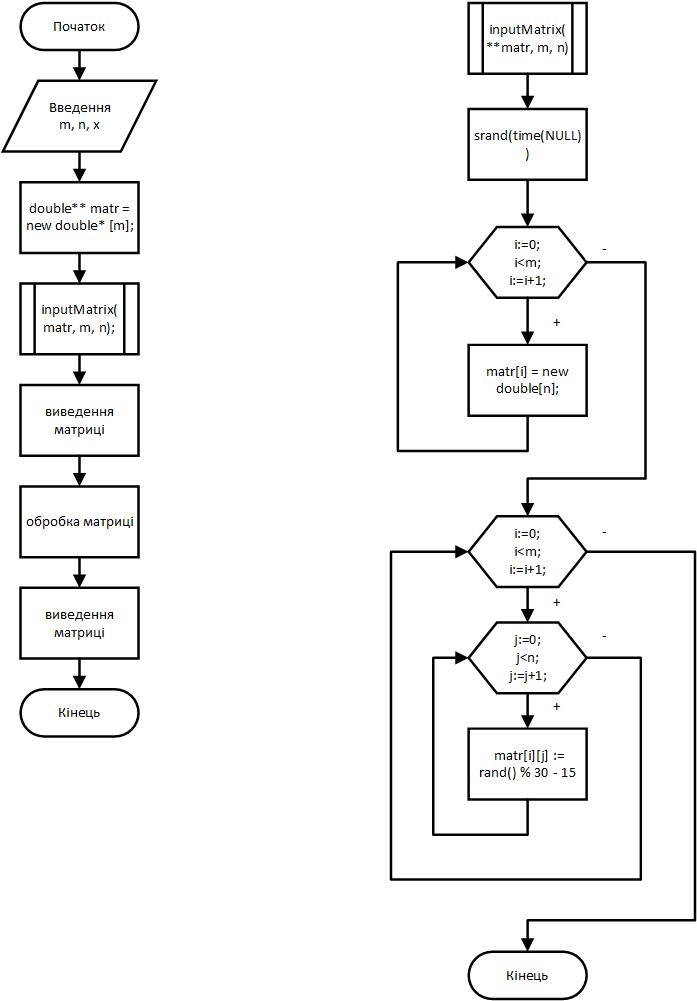
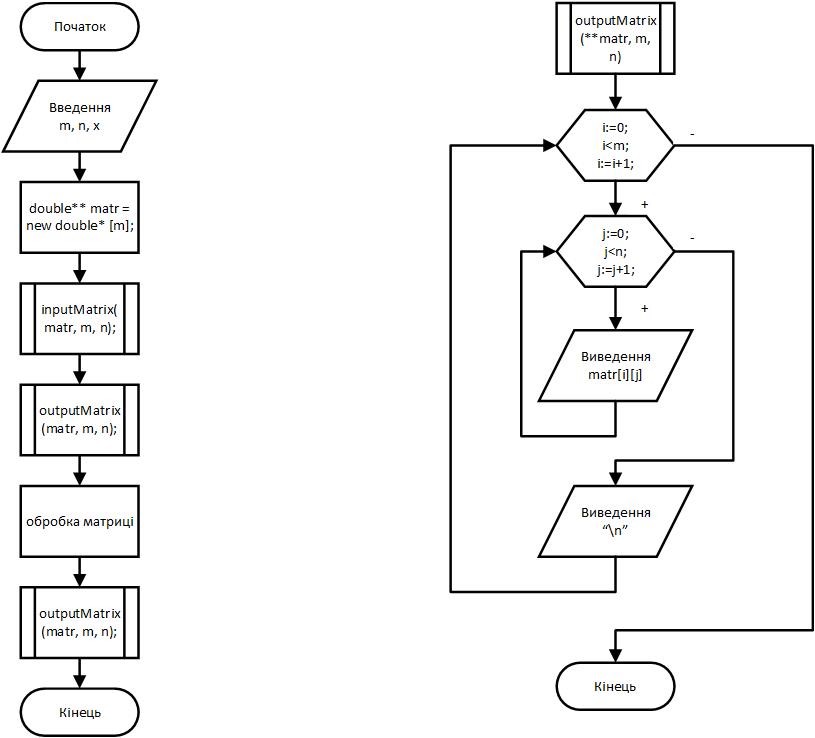
*Крок 1*



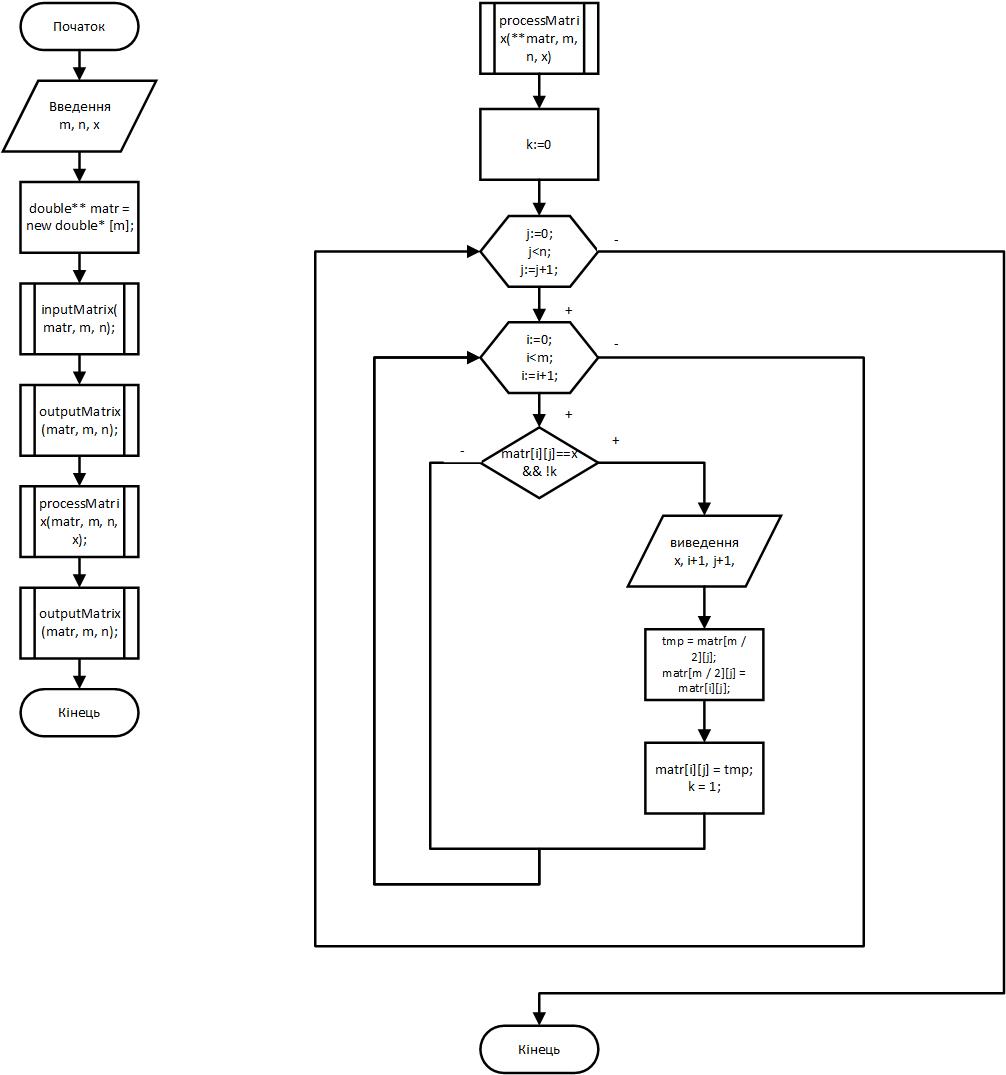
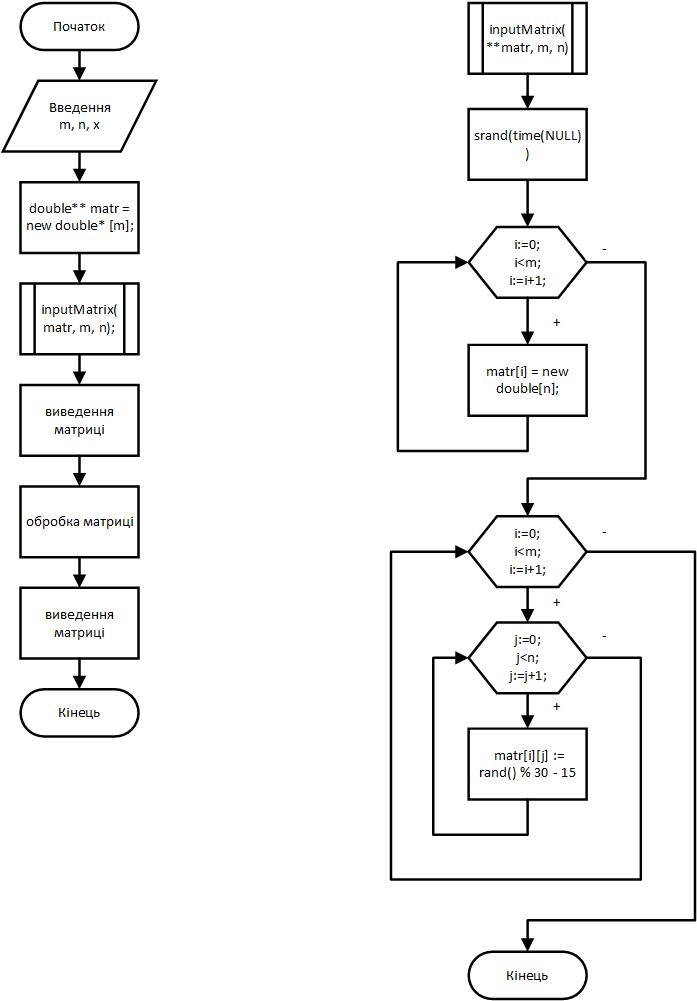
*Крок 2*

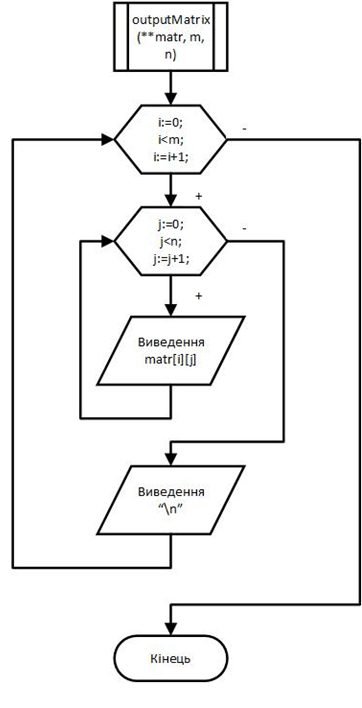
 

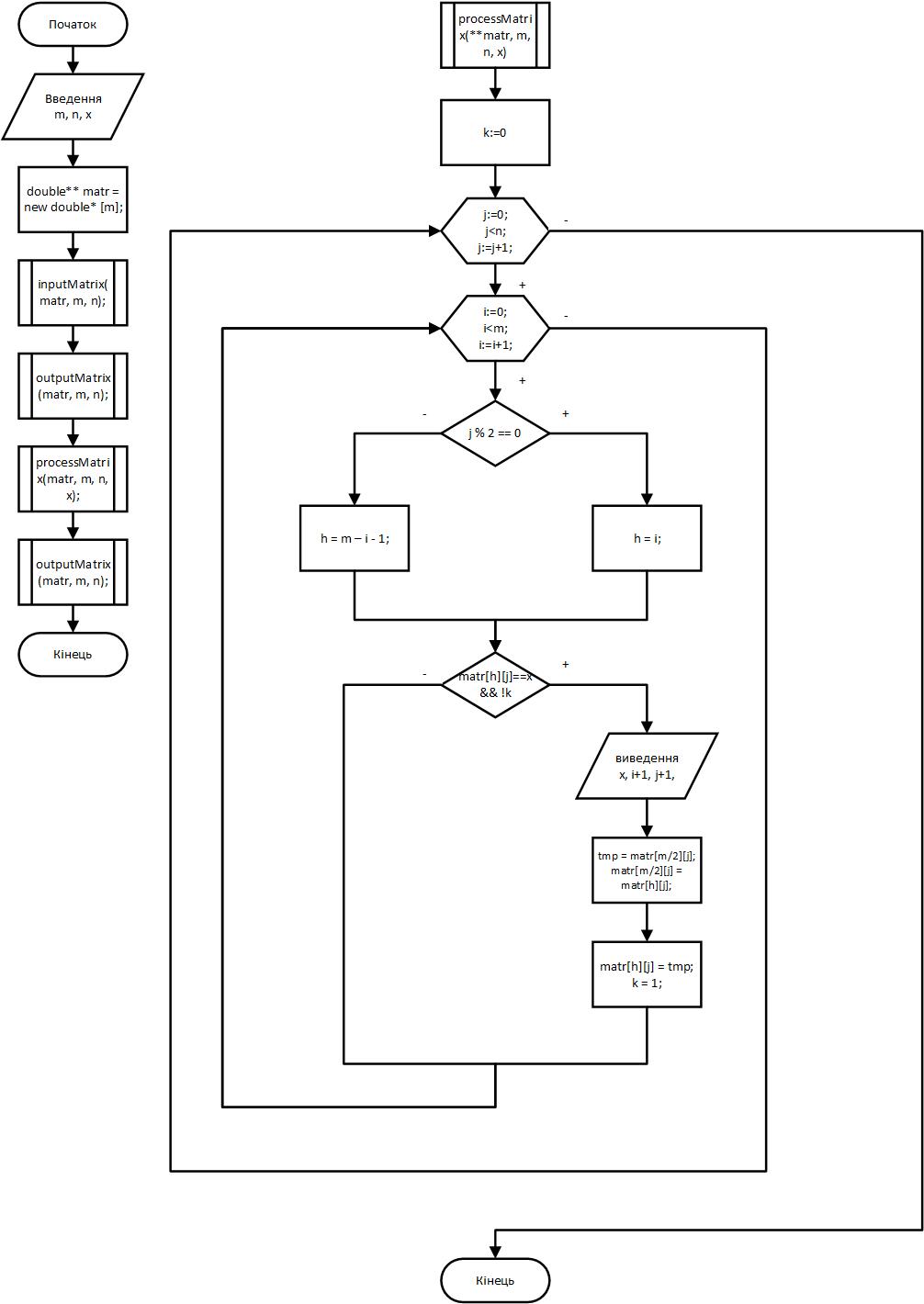
*Крок 3*

*Крок 4*

**



**Виконання мовою C++.**

**Код програми:**

#include <iostream>

#include <ctime>

using namespace std;

void inputMatrix(double\*\*, int, int); //Прототипи функцій: введення матриці

void outputMatrix(double\*\*, int, int); //Виведення матриці

void processMatrix(double\*\*, int, int, double); //Обробка згідно з умовою

int main()

{

setlocale(LC\_ALL, "ukr");

int m, n;

double x;

cout << "Введiть кiлькiсть рядкiв i стовпцiв матрицi: ";

cin >> m >> n;

double\*\* matr = new double\* [m]; //Ініціалізація динамічного масиву вказівників

inputMatrix(matr, m, n);

cout << "\nПочаткова матриця:\n";

outputMatrix(matr, m, n);

cout << "\nВведiть дiйсне число x: ";

cin >> x;

processMatrix(matr, m, n, x);

cout << "\nЗмiнена матриця:\n";

outputMatrix(matr, m, n);

}

void inputMatrix(double \*\*matr, int m, int n) {

srand(time(NULL));

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

matr[i] = new double[n];

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

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

matr[i][j] = rand() % 30 - 15; //Заповнення матриці випадковим чином від -15 до 15

}

}

void outputMatrix(double\*\* matr, int m, int n) {

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

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

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

}

cout << endl;

}

}

void processMatrix(double\*\*matr, int m, int n, double x) {

bool k = 0;

int h;

for (int j = 0; j < n; j++) //Обхід матриці по стовпцях

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

if (j % 2 == 0)

h = i;

else

h = m - i-1;

if (matr[h][j] == x && !k) { //Перше входження елементу X по стовпцях

cout << "\nЕлемент X = " << x << " знайдено в " << h + 1 << " рядку й " << j + 1 << " стовпцi." << endl;

double tmp = matr[m/2][j];

matr[m/2][j] = matr[h][j]; //Обмінюємо Х з елементом середнього рядка того ж стовпця

matr[h][j] = tmp;

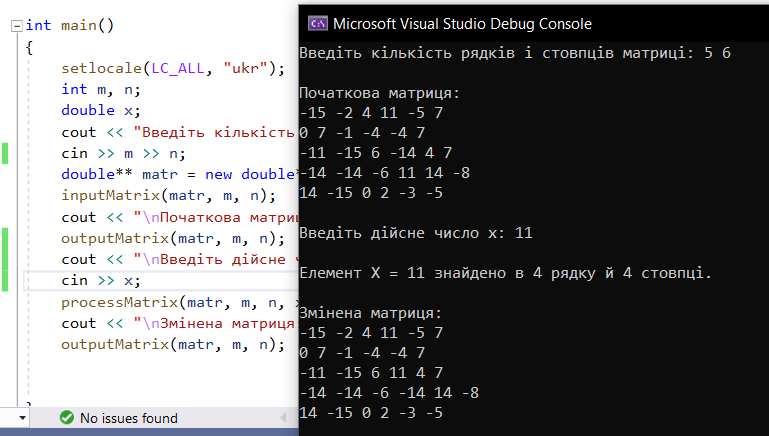
k = 1;

}

}

}

**Випробування алгоритму.**



**Перевірка алгоритму.**

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | Введення m=5, n=6, x=11 |
| 2 | inputMatrix(matr, m, n) – заповнення випадковим чином |
| 3 | outputMatrix(matr,m,n)–виведення матриці (на скриншоті) |
| 4 | processMatrix(matr,m,n,x) – обробка матриці – пошук елементу Х – результат «Елемент Х знайдено в 4 рядку і 4 стовпці» (matr[3][3]==11). Заміна його з елементом matr[2][3] – результат matr[2][3]=11, matr[3][3]=-14 |
| 5 | outputMatrix(matr,m,n)–виведення матриці (на скриншоті) |
|  | Кінець |

**Висновок.** Отже, у цій роботі я дослідив алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій. У результаті лабораторної роботи було розроблено математичну модель, що відповідає постановці задачі; псевдокод та блок-схеми, які пояснюють логіку алгоритму, а також програму, що виконує задачу відповідно до постановки. Використовуючи три підпрограми для роботи з масивами – одна для введення матриці, одна – для її виведення, одна – для обробки масиву відповідно до задачі, використовуючи алгоритм обходу за стовпцями змійкою - отримуємо коректний результат.