Основи програмування – 1. Алгоритми та структури даних

Додаток 1

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-1.

Основи алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант 18

Виконав студент ІП-13 Король Валентин Олегович

Перевірив Вечерковська Анастасія Сергіївна

Київ 2021

Алгоритми та структури даних. Основи алгоритмізації

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

18	Задано матрицю дійсних чисел A[m,n]. При обході матриці по стовпчиках визначити в ній присутність заданого дійсного числа X і його місцезнаходження. Обміняти знайдене значення X з елементом середнього рядка.

- **Постановка задачі**

Результатом розв’язку є двовимірний масив, в якому елемент x заміняється елементом з середнього рядка.

- **Побудова математичної модель**

Змінна	Тип	Ім’я	Призначення
Двовимірний масив	Дійсний	Matrix	Проміжне і Кінцеве дане

Підпрограма **fill_array**

Змінна	Тип	Ім’я	Призначення
Двовимірний масив	Дійсний	matrix	Проміжне дане
Рядочки	Цілий	m	Вхідні дані
Стовпці	Цілий	n	Вхідні дані
Ітератор	Цілий	i	Проміжні дані
Ітератор	Цілий	j	Проміжні дані

Підпрограма **create_matrix**

Змінна	**Тип**	**Ім’я**	**Призначення**
Двовимірний масив	Дійсний	matrix	Проміжне дане
Рядочки	Цілий	m	Вхідні дані
Стовпці	Цілий	n	Вхідні дані
Ітератор	Цілий	i	Проміжні дані
Ітератор	Цілий	j	Проміжні дані

Підпрограма **change_matrix**

Змінна	**Тип**	**Ім’я**	**Призначення**
Двовимірний масив	Дійсний	matrix	Проміжне дане
Рядочки	Цілий	m	Вхідні дані
Стовпці	Цілий	n	Вхідні дані
Задане число масива	Дійсний	x	Проміжні дані
Ітератор	Цілий	i	Проміжні дані
Ітератор	Цілий	j	Проміжні дані

Спочатку генеруємо матрицю за допомогою рандому в підпрограмі **fill_array**. Виводимо відповідну матрицю в підпрограмі **create_matrix**.

Вибираємо число x, з масива, заміняємо його на значення з відповідного середнього рядка за допомогую підпрограми **change_matrix**. Виводимо змінену матрицю.

- **Розв’язання**

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

Крок 1. Визначити основні дії.

Крок 2. Згенерувати матрицю.

Крок 3. Вивести матрицю.

Крок 4. Замінити x елемент на відповідний елемент з середнього рядка.

Крок 5. Вивести змінену матрицю.

● **Псевдокод алгоритму** Крок 1.

підпрограма fill_array (Matrix, m, n)

Заповнення двовимірного масиву.

все підпрограма

підпрограма create_matrix (Matrix, m, n)

Виведення двовимірного масиву.

все підпрограма

підпрограма change_matrix (Matrix, m, n)

Замінити x елемент на відповідний елемент з середнього рядка.

все підпрограма

підпрограма create_matrix (Matrix, m, n);

Виведення зміненого масиву.

все підпрограма

початок

введення m, n

double** Matrix = new double* [m]

Matrix = **fill_array** (Matrix, m, n)

create_matrix (Matrix, m, n)

введення x

change_matrix (Matrix, m, n)

create_matrix (Matrix, m, n)

кінець

Крок 2.

підпрограма fillMatrix(matrix, m, n)

повторити для i **від** 0 **до** m**, збільшувати на 1**

Matrix[i] = new double[n]

повторити для j **від** 0 **до** n**, збільшувати на 1**

matrix[i][j] = (double)(rand() % 100) **все повторити** **все повторити** return matrix

все підпрограма

підпрограма create_matrix (matrix, m, n)

повторити для i **від** 0 **до** m**, збільшувати на 1**

повторити для j **від** 0 **до** n**, збільшувати на 1**

виведення Matrix[i][j] **все повторити**

все повторити

все підпрограма

підпрограма change_matrix (matrix, m, n)
повторити для j **від** 0 **до** n**, збільшувати на 1**

якщо j % 2 == 1

то

повторити для i **від** 0 **до** m**, збільшувати на 1**

Якщо Matrix[i][j] == x

то

Matrix[i][j] = Matrix[l][j]
Matrix[l][j] = x
все якщо **все**

повторити **все**

повторити все

підпрограма

підпрограма create_matrix (Matrix, m, n);

повторити для i **від** 0 **до** m**, збільшувати на 1**

повторити для j **від** 0 **до** n**, збільшувати на 1**

виведення Matrix[i][j]

все повторити

все повторити все підпрограма

початок

введення m, n

double** Matrix = new double* [m]

Matrix = **fill_array** (Matrix, m, n)

create_matrix (Matrix, m, n)

введення x

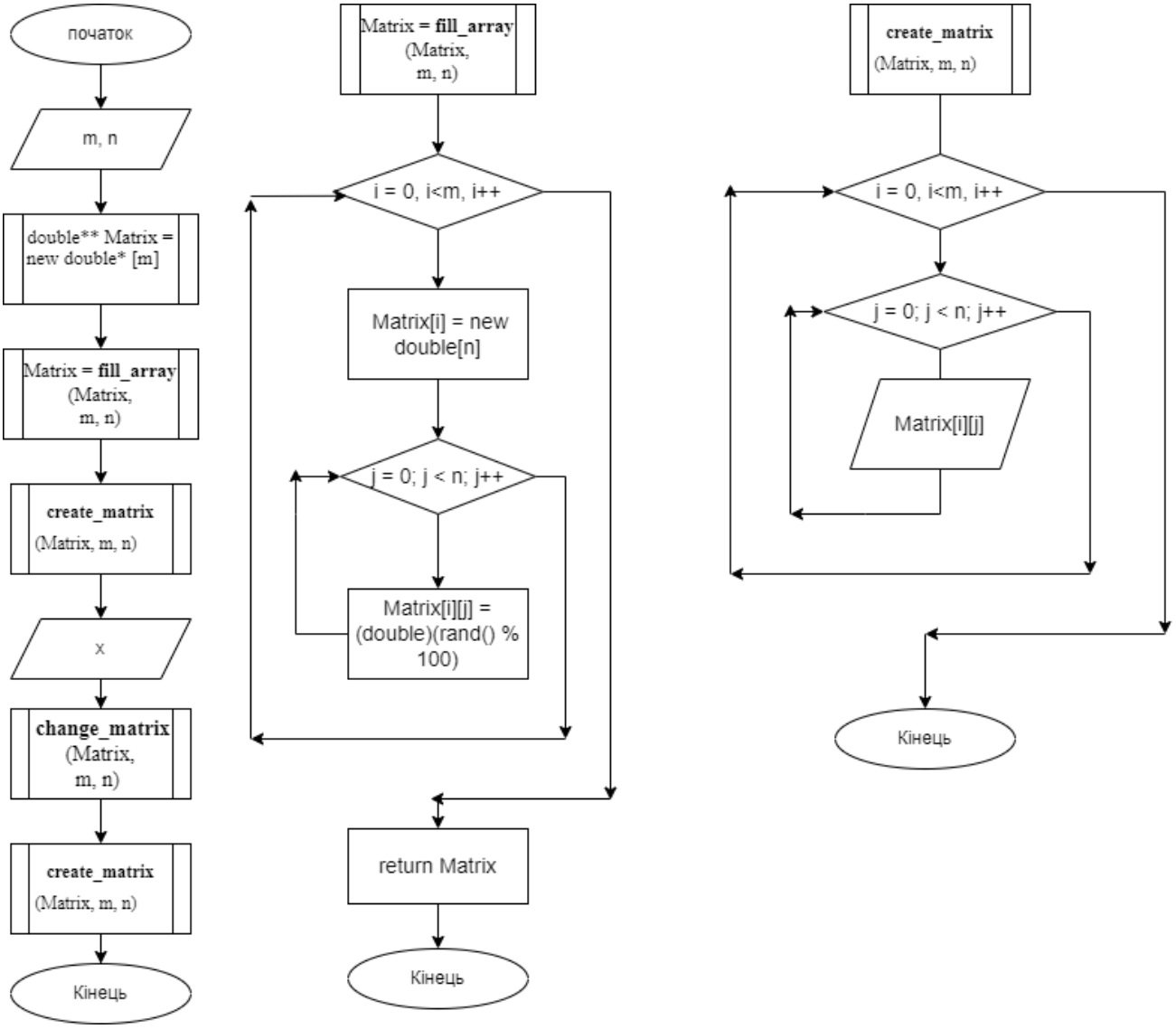
change_matrix (Matrix, m, n)

create_matrix (Matrix, m, n)

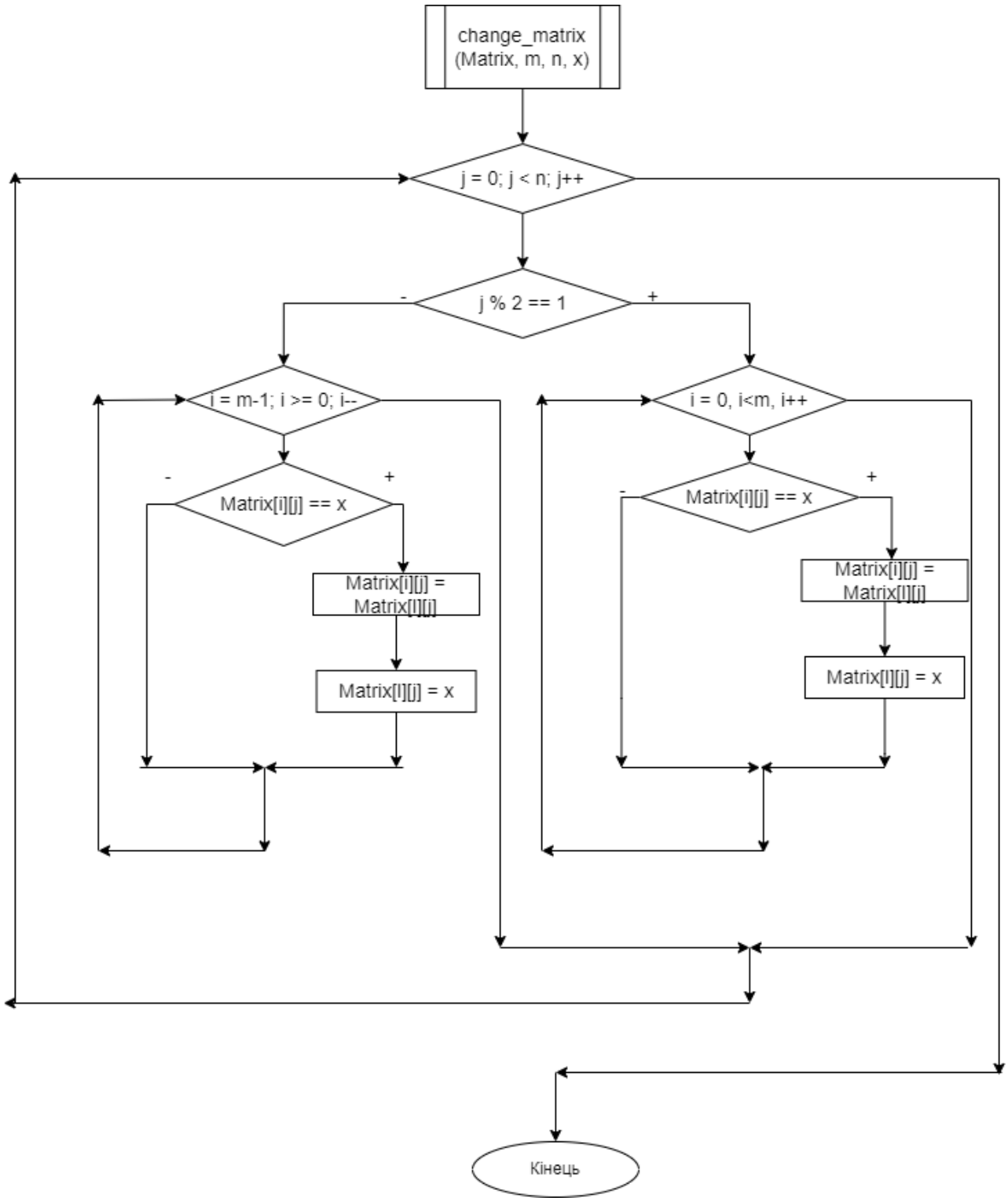
кінець

● Блоксхема алгоритму

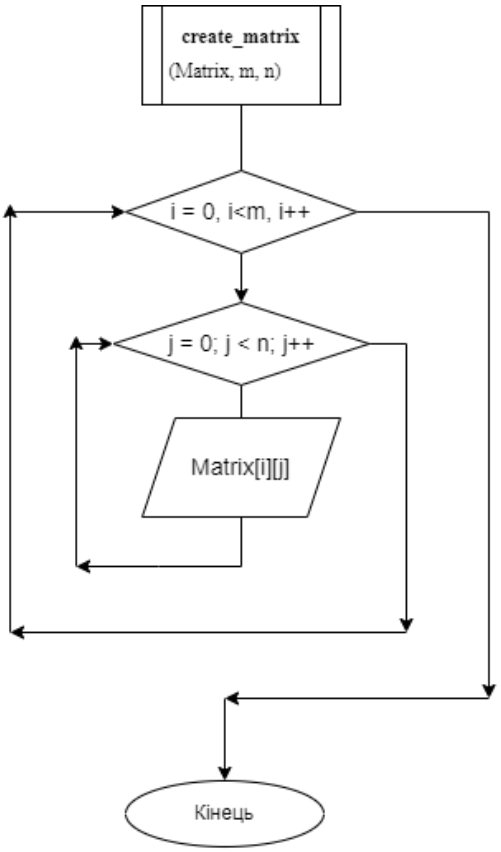
Крок 1. Крок 2. Крок 3.


початок
m, n
double** Matrix = new double* [m]
Matrix = fill_array (Matrix, m, n)
create_matrix (Matrix, m, n)
x
change_matrix (Matrix, m, n)
create_matrix (Matrix, m, n)
Кінець
Matrix = fill_array (Matrix, m, n)
i = 0, i<m, i++
Matrix[i] = new double[n]
j = 0; j < n; j++
Matrix[i][j] = (double)(rand() % 100)
return Matrix
Кінець
create_matrix (Matrix, m, n)
i = 0, i<m, i++
j = 0; j < n; j++
Matrix[i][j]
Кінець

Крок 4.


change_matrix
(Matrix, m, n, x)
j = 0; j < n; j++
j % 2 == 1
-
+
i = m-1; i >= 0; i--
Matrix[i][j] == x
-
+
Matrix[i][j] =
Matrix[l][j]
Matrix[l][j] = x
i = 0, i<m, i++
Matrix[i][j] == x
-
+
Matrix[i][j] =
Matrix[l][j]
Matrix[l][j] = x
Кінець

Крок 5.



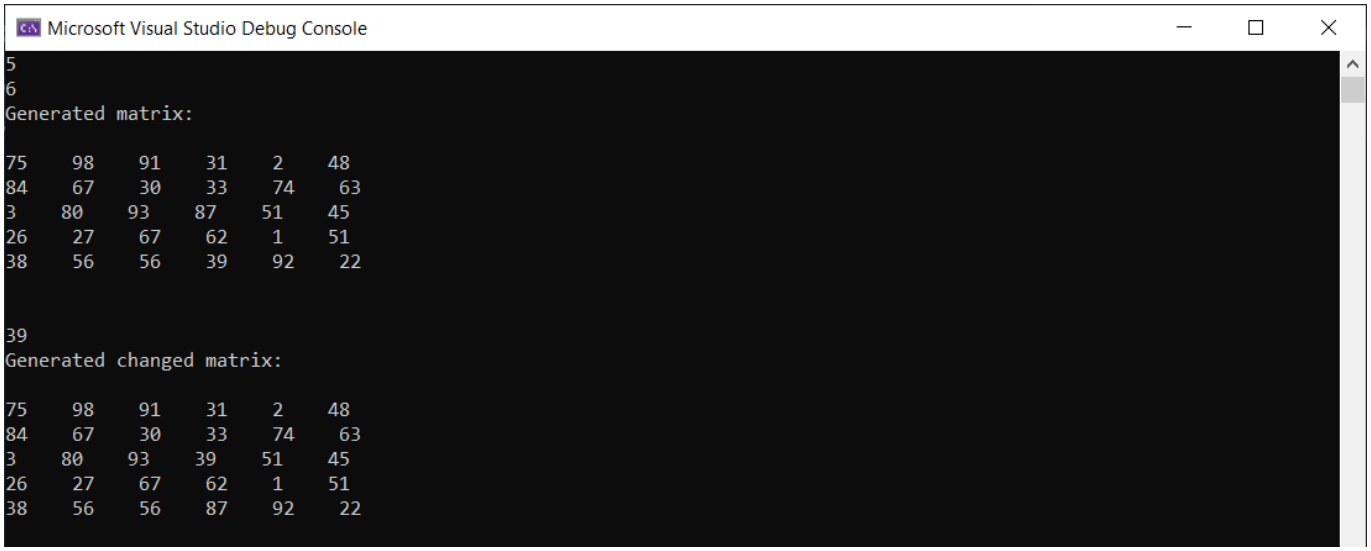
- **Код програми на C++**

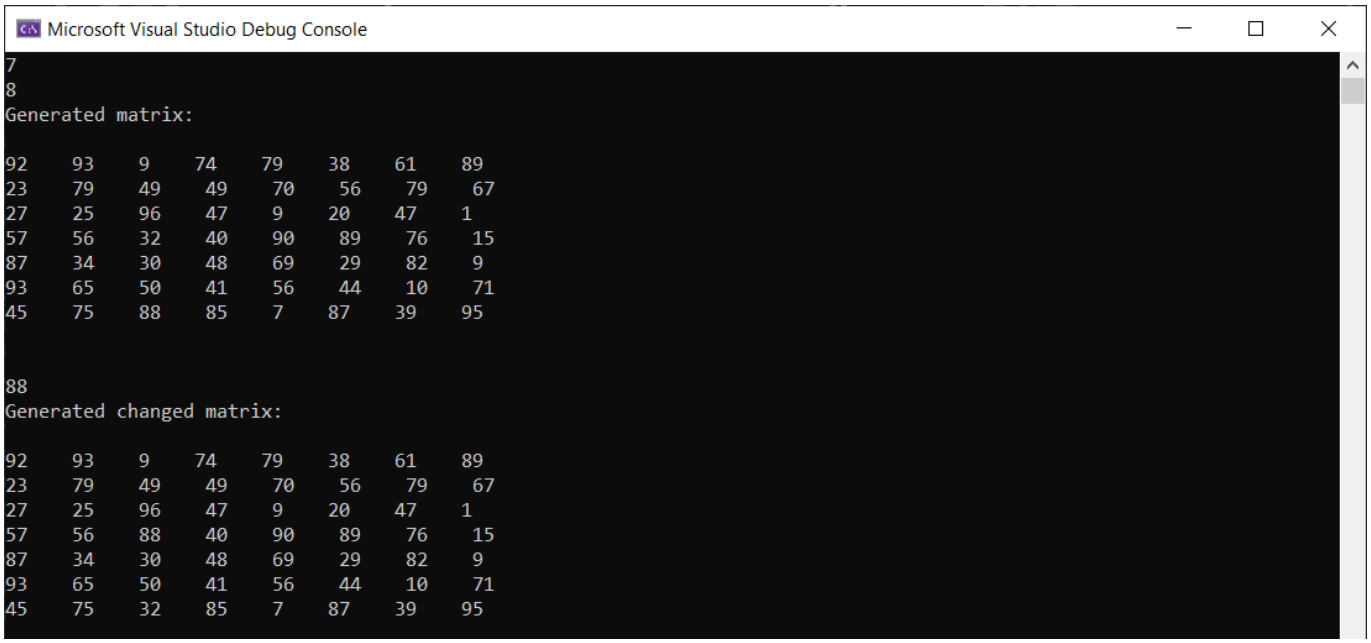
```
(Global Scope)                                  change_matrix(double ** M
#include <iomanip>
#include <iostream>
#include <ctime>

using namespace std;

double** fill_array(double**, int, int);
void create_matrix(double**, int, int);
void change_matrix(double**, int, int, double);

double** fill_array(double** Matrix, int m, int n){
    for (int i = 0; i < m; i++) {
        Matrix[i] = new double[n];
        for (int j = 0; j < n; j++) {
            Matrix[i][j] = (double)(rand() % 100);
        }
    }
    return Matrix;
}

void create_matrix(double** Matrix, int m, int n) {
    cout << endl;
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            cout << Matrix[i][j] << "    ";
        }
        cout << endl;
    }
    cout << endl;
}
```

```
void change_matrix(double** Matrix, int m, int n, double x) {
  for (int j = 0; j < n; j++) {
    if (j % 2 == 1) {
        for (int i = 0; i < m; i++) {
            if (Matrix[i][j] == x) {
                int l = 0;
                l = m / 2;
                Matrix[i][j] = Matrix[l][j];
                Matrix[l][j] = x;
            }
        }
    }
        else {
              for (int i = m-1; i >= 0; i--) {
                  if (Matrix[i][j] == x) {
                      int l = 0;
                      l = m / 2;
                      Matrix[i][j] = Matrix[l][j];
                      Matrix[l][j] = x;
                  }
              }
        }

  }

}
```

```cpp
int main()
{

    int m; cin >> m;
    int n; cin >> n;

    double** Matrix = new double* [m];

    srand(time(NULL));
    cout << "Generated matrix: \n";
    Matrix = fill_array(Matrix, m, n);
    create_matrix(Matrix, m, n);
    cout << endl;
    double x; cin >> x;
    change_matrix(Matrix, m, n, x);
    cout << "Generated changed matrix: \n";
    create_matrix(Matrix, m, n);



    return 0;

```

- **Випробування алгоритму**





- **Висновок**

На цій лабораторній роботі я дослідив методи обходу масивів. Навчився писати псевдокод, складати блок схему, написав код для вирішення своєї задачі.