Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічнийінститут 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 29

Виконав студент ІП-13 Романюк Діана Олексіївна
(шифр, прізвище, ім'я, по батькові)

Перевірив Вєчерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Лабораторна робота 9

ДОСЛІДЖЕННЯ АЛГОРИТМІВ ОБХОДУ МАСИВІВ

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

Варіант 29

Постановка задачі:

Задати матрицю обходом по стовпцях, визначити чи присутнє задане число на побчній діагоналі, порівняти це число із середнім арифметичним елементів, що знаходяться під побічною діагоналлю.

Побудова математичної моделі:

Таблиця імен змінnних

Змінна	Тип	Ім’я	Призначення
Розмір масиву	цілий	size	Вхідні дані
Матриця	дійсний	Array	Вихідні дані
Лічильник	цілий	i	Проміжні дані
Лічильник	цілий	j	Проміжні дані
Задане число	дійсний	X	Проміжні дані
Мінімальне число у проміжку генерації	цілий	MIN	Проміжні дані
Максимальне число у проміжку генерації	цілий	MAX	Проміжні дані
Зміна порядку елементів	цілий	dir	Проміжні дані
Визначення місцезнаходження X	цілий	non	Проміжні дані

Сума елементів під побічною діагоналлю	дійсний	Sum	Проміжні дані
Лічильник	цілий	k	Проміжні дані
Середнє арифметичне елементів під побічною діагоналлю	дійсний	Avg	Проміжні дані

inArray() – підпрограма для заповнення масиву.
Out() – підпрограма для виводу масиву.
findX() – підпрограма для визначення місцезнаходження числа X
Avg() – підпрограма для пошуку і порівнння значення середнього арифметичного елементів під побічною діагоналлю**.**
Задати у головній програмі розмір динамічного масиву, ініціалізувати динамічний масив, визвати функцію заповнення масиву за допомогою вкладених циклів. У головній програмі задати деяке число X. Визвати функцію для пошуку цього числа на побічній діагоналі за допомогою двох вкладених циклів «повторити для» та двох циклів «якщо, то». Визвати функцію для пошуку середнього арифметичного, за допомогою обходу матриці під побічною діагоналлю та використати вкладені цикли для виводу результату порівняння.

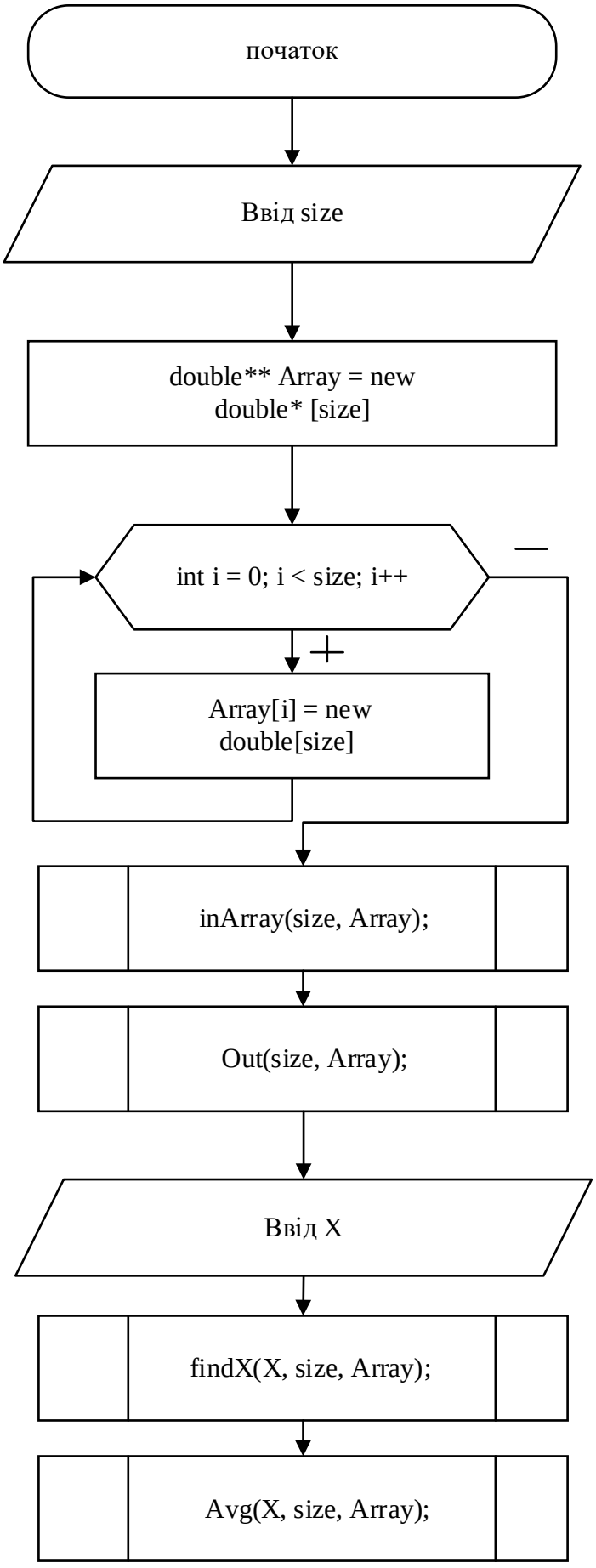
Задача:

Задати матрицю обходом по стовпцях, визначити чи присутнє задане число на побчній діагоналі, порівняти це число із середнім арифметичним елементів, що знаходяться під побічною діагоналлю.

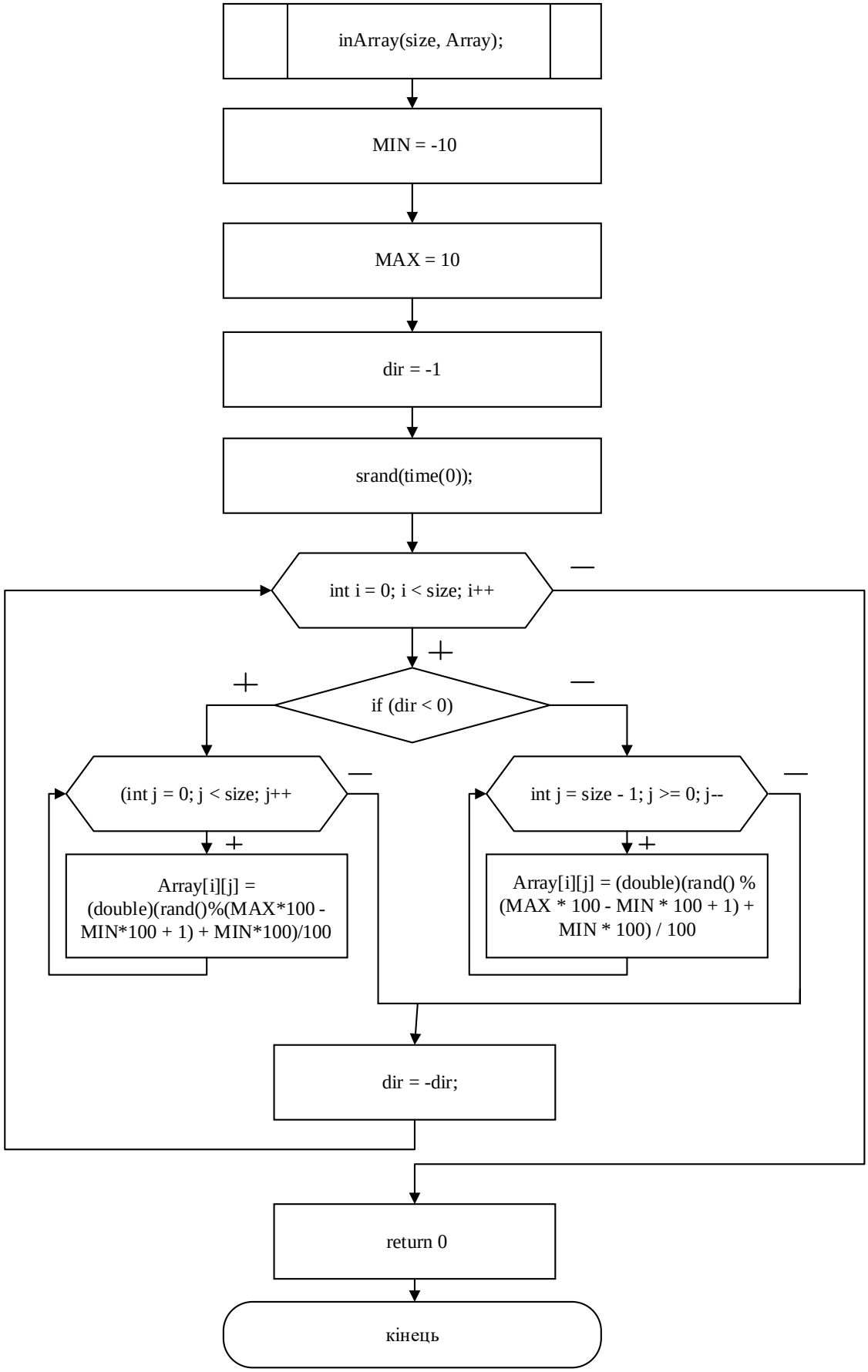
Псевдокод алгоритму:

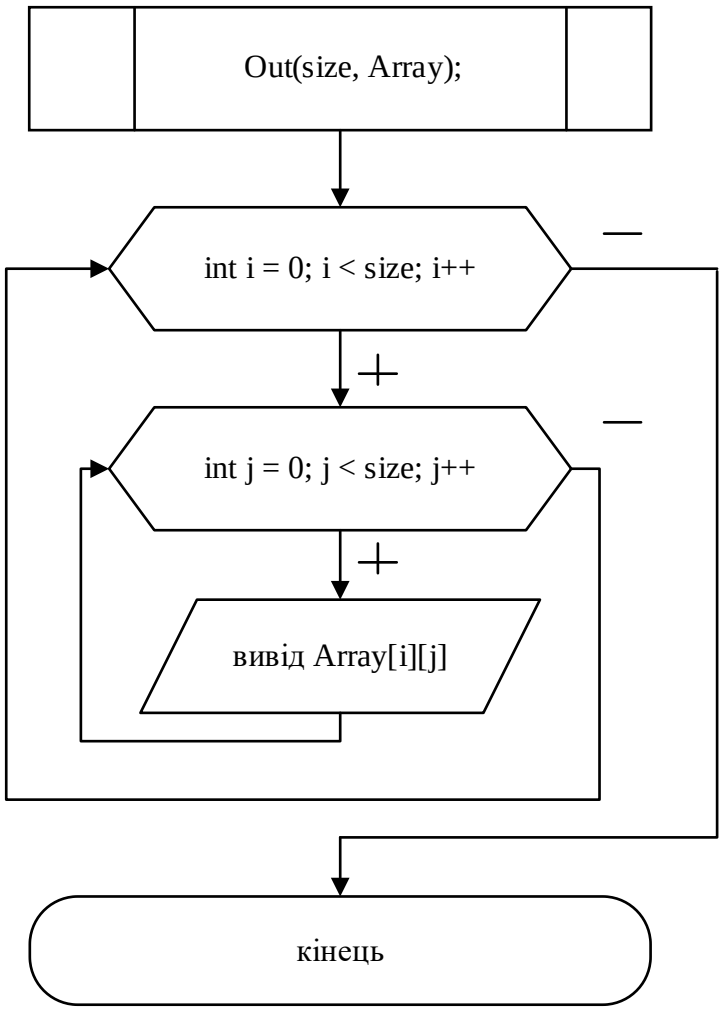
```
початок
    введення size
    double** Array = new double* [size];
    повторити для i = 0; i < size; i++
       Array[i] = new double[size]
    все повторити
    inArray(size, Array)
    Out(size, Array)
    введення X
    findX(X, size, Array)
     Avg(X, size, Array)
кінець
```

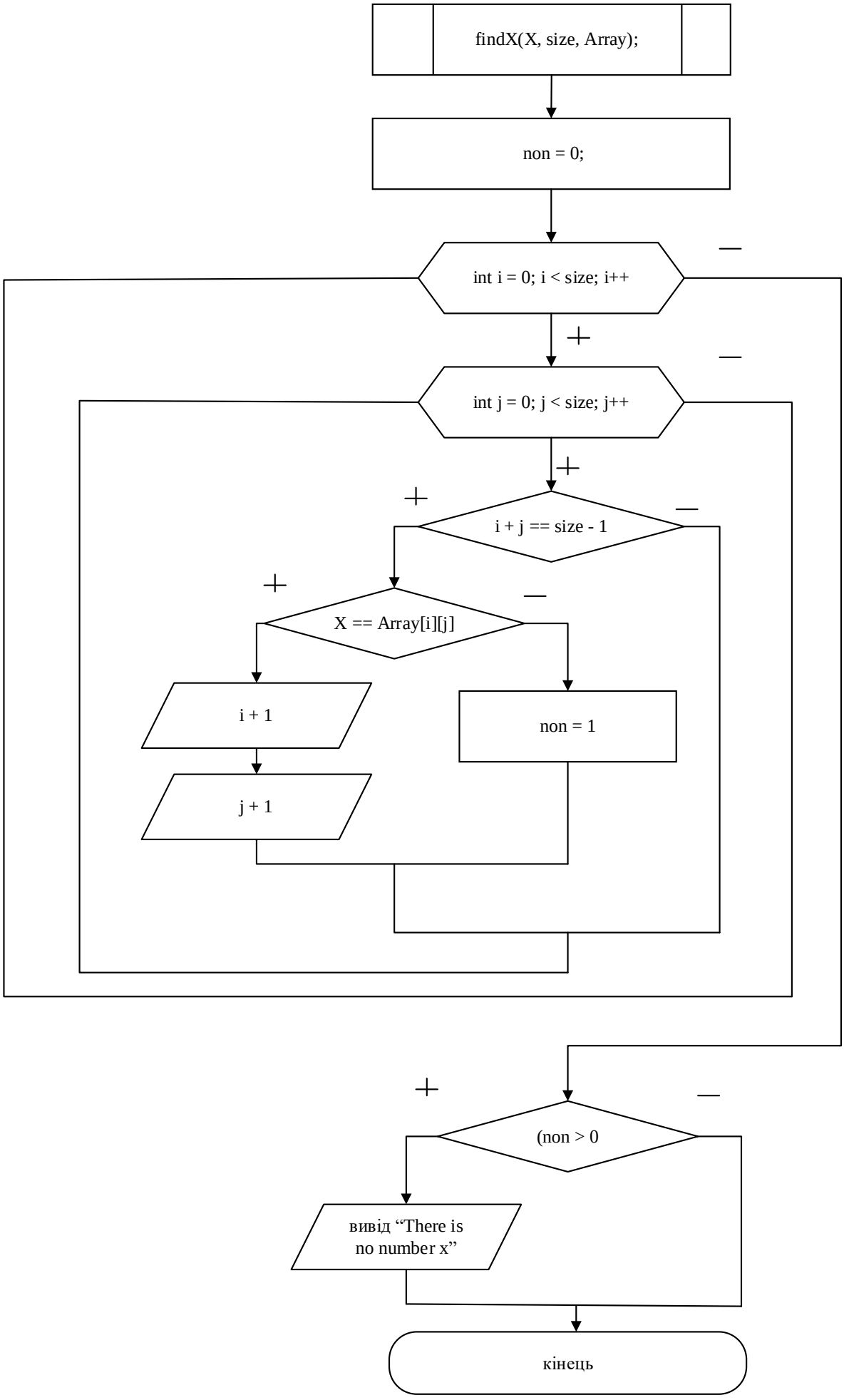
Блок-схема програми:

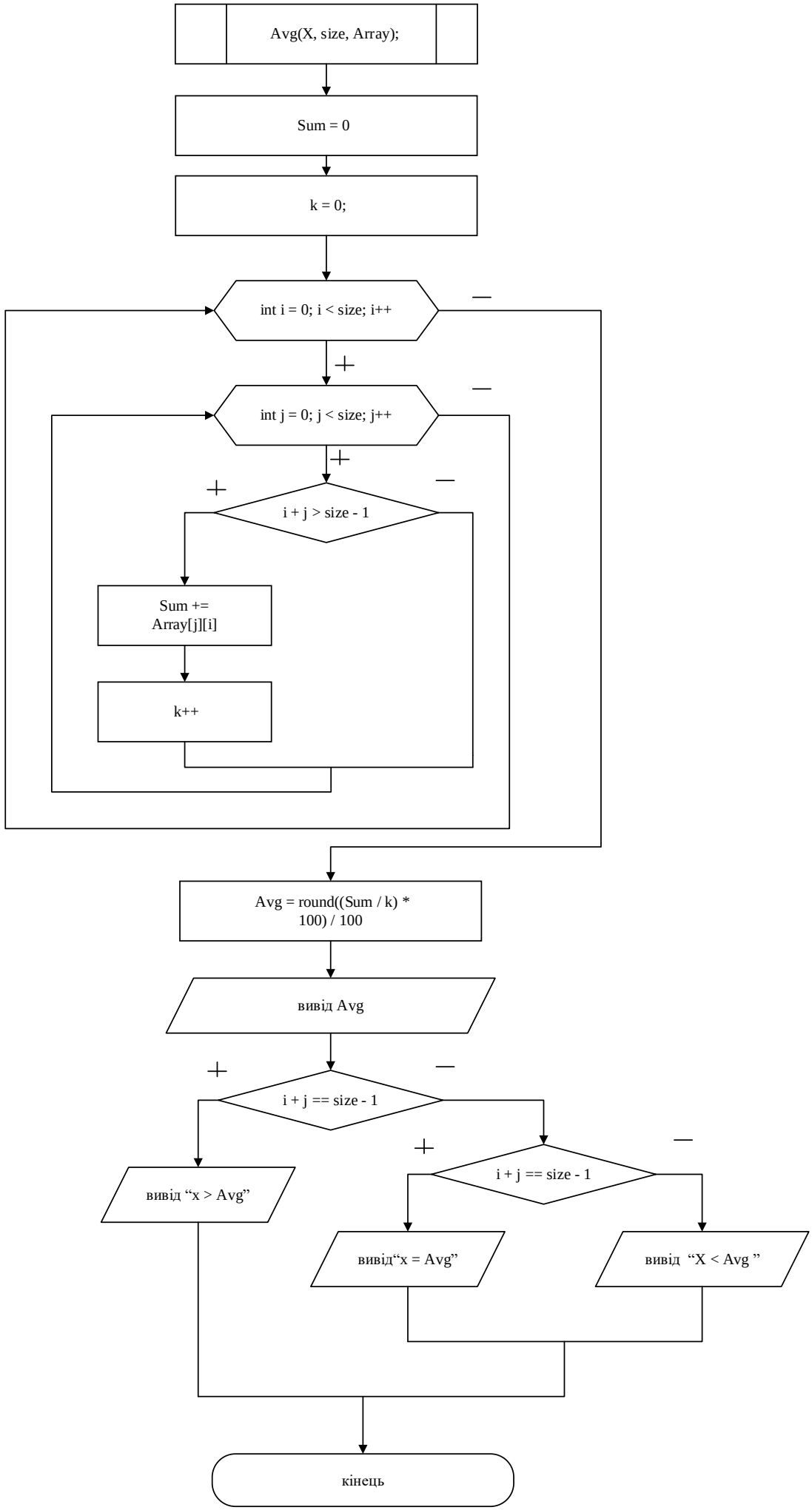


Блок-схема підпрограми:




Out(size, Array);
int i = 0; i < size; i++
int j = 0; j < size; j++
вивід Array[i][j]
кінець


findX(X, size, Array);
non = 0;
int i = 0; i < size; i++
int j = 0; j < size; j++
i + j == size - 1
X == Array[i][j]
i + 1
j + 1
non = 1
(non > 0
вивід “There is no number x”
кінець


Avg(X, size, Array);
Sum = 0
k = 0;
int i = 0; i < size; i++
int j = 0; j < size; j++
i + j > size - 1
Sum += Array[j][i]
k++
Avg = round((Sum / k) * 100) / 100
вивід Avg
i + j == size - 1
вивід “x > Avg”
i + j == size - 1
вивід“x = Avg”
вивід “X < Avg ”
кінець

Код програми:

```cpp
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

double inArray(int size, double** Array) {
    int MIN = -10;
    int MAX = 10;
    int dir = -1;
    srand(time(0));

    for (int i = 0; i < size; i++) {
        if (dir < 0) {
            for (int j = 0; j < size; j++) {
                Array[i][j] = (double)(rand()%(MAX*100 - MIN*100 + 1) + MIN*100)/100;
            }
        }
        else {
            for (int j = size - 1; j >= 0; j--) {
                Array[i][j] = (double)(rand() % (MAX * 100 - MIN * 100 + 1) + MIN * 100) / 100;
            }
        }

        dir = -dir;
    }
    return 0;
}
void Out(int size, double** Array) {

    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            cout << Array[i][j] << "\t";
        }
        cout << endl;
    }

}
void findX(double X, int size, double** Array) {
    int non = 0;
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (i + j == size - 1) {
                if (X == Array[i][j]) {
                    cout << "X is on the " << i + 1 << " row and ";
                    cout << j + 1 << " column ";
                }
                else { non = 1; }
            }
        }
    }
    if (non > 0) {
        cout << "There is no number x on the side diagonal";
    }
}
void Avg(double X, int size, double** Array) {
    double Sum = 0;
    int k = 0;
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (i + j > size - 1) {
                Sum += Array[j][i];
                k++;
            }
        }
    }
    double Avg = round((Sum / k) * 100) / 100;
    cout << "\n" << "Average of the elements under the secondary diagonal of the matrix = " << Avg << "\n";
    if (X > Avg) {
        cout << "X > Average";
    }
    else if (X == Avg) {
        cout << "X = Average";
    }
    else {
        cout << "X < Average";
    }

}
```

```
int main(){
        int size;
        cout << "Enter the dimension of the matrix ";
        cin >> size;
        double** Array = new double* [size];
        for (int i = 0; i < size; i++) {
            Array[i] = new double[size];
        }
        inArray(size, Array);
        Out(size, Array);
        double X;
        cout << "Enter x = ";
        cin >> X;
        findX(X, size, Array);
        Avg(X, size, Array);
}
```

Консоль отладки Microsoft Visual Studio

```
Enter the dimension of the matrix 3
-2.84   -8.59   7.21
-7.8    8.87    7.31
-8.96   5.6     -7.72
Enter x = -7.8
There is no number x on the side diagonal
Average of the elements under the secondary diagonal of the matrix = 1.73
X < Average
C:\Users\Asus\source\repos\ConsoleApplication1\Debug\ConsoleApplication1.exe (процесс 3984) заве
```

Выбрать Консоль отладки Microsoft Visual Studio

```
Enter the dimension of the matrix 4
7.02    -8.73   9.87    -6.41
-7.42   7.98    -5.59   9.28
2.38    7.32    -5.05   -0.31
7.53    -4.7    2.68    8.94
Enter x = 7.32
X is on the 3 row and 2 column
Average of the elements under the secondary diagonal of the matrix = 1.81
X > Average
```

Висновки:

Під час виконання лабораторної роботи досліджено подання операторів повторення дій. Отримано практичні навички їх використання підчас складання лінійних програмних специфікацій. Побудовано математичну модель задачі та таблицю імен змінних. Розроблено псевдокод вирішення даної математичної задачі. Умовно розбито виконання коду на кроки, а також описано його виконання за допомогою створення відповідної блок-схеми. Перевірено умовне виконання коду за допомогою випробування алгоритму.