

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 алгоритмів обходу масивів»

Варіант 35

Виконав студент ІП-15, Шабанов Метін Шаміль огли  
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська 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.

#### Індивідуальне завдання

#### Варіант 35

#### Умова задачі

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом.
2. Ініціювання змінної, що описана в п.1 даного завдання.
3. Обчислення змінної, що описана в п.1, згідно з варіантом.

Задано матрицю дійсних чисел  $A[n, n]$ , ініціалізувати матрицю обходом по стовбцях. У побічній діагоналі матриці визначити присутність заданого дійсного числа  $X$  і його місцезнаходження. Порівняти значення  $X$  із середньоарифметичним значенням елементів під головною діагоналлю.

#### Постановка задачі

Визначити присутність заданого дійсного числа  $X$  і його місцезнаходження та порівняти його значення із середньоарифметичним значенням елементів під головною діагоналлю матриці, яка була ініціалізована обходом по стовбцях.

#### Побудова математичної моделі

Змінна	Тип	Ім'я	Призначення
Генератор матриці	Підпрограма	matrixGenerator(matrix[,])	Функція
Визначення присутності числа у побічній діагоналі	Підпрограма	isInSideDiagonal(matrix[,], x)	Функція
Визначення середньоарифметичного значення під головною діагоналлю	Підпрограма	averageUnderMainDiagonal(matrix[,])	Функція
Порівняння двох значень	Підпрограма	comparator (a, b)	Функція
Кількість рядків і стовпців матриці	Ціле	size	Вхідні дані
Матриця	Індексований	matrix[size, size]	Проміжні дані
Довжина матриці	Метод	.GetLength()	Отримати

у відповідному вимірі			довжину матриці у відповідному вимірі
Лічильники циклів для ініціалізації та обробки масиву	Ціле додатне	i, j	Проміжні дані
Значення числа X	Дійсний	x	Вхідні дані
Напрямок ініціалізації по стовпцях	Ціле	dir	Проміжні дані
Поточне значення елемента масиву	Дійсне	currentNum	Проміжні дані
Сума елементів під головною діагоналлю	Дійсне	sum	Проміжні дані
Кількість елементів під головною діагоналлю	Дійсне	numberOfElements	Проміжні дані
Середньоарифметичне значення елементів під головною діагоналлю	Дійсне	average	Проміжні дані

Для ініціалізації обходу по стовпцях використаємо змінну `dir`, завдяки якій зможемо контролювати напрям “змійки”. Визначати присутність значення змінної `X` будемо за допомогою умовного оператора та циклу.

### Розв’язання

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

Крок 1. Визначимо основні дії;

Крок 2. Ініціалізуємо `size`;

Крок 3. Деталізуємо створення матриці розмірності `[size, size]` та ініціалізацію елементів матриці обходом по стовпцях у підпрограмі `matrixGenerator`;

Крок 4. Деталізуємо визначення присутності та місцезнаходження елемента `X` у побічній діагоналі за допомогою функції `isInSideDiagonal`;

Крок 5. Деталізуємо знаходження і порівняння середньоарифметичного значення під головною діагоналлю з X.

### Псевдокод

#### Основна програма

matrix [size, size]

matrixGenerator(matrix)

isInSideDiagonal(matrix, x)

averageUnderMainDiagonal(matrix)

comparator(x, average)

#### Підпрограма matrixGenerator (matrix[,])

dir = -1

currentNum = 1

##### повторити

для j від 0 до matrix.GetLength(1)

якщо dir < 0

то

##### повторити

для i від 0 до matrix.GetLength(0)

matrix[i, j] = currentNum

currentNum++

##### все повторити

dir = -dir

інакше

##### повторити

для i від (matrix.GetLength(0) – 1) до 0 включно

```
matrix[i, j] = currentNum  
currentNum++
```

**все повторити**

```
dir = -dir
```

**все якщо**

**все повторити**

**повернути** matrix

**Підпрограма** isInSideDiagonal (matrix[,], x)

```
i = 0
```

**повторити**

**для** j від (matrix.GetLength(1) – 1) до 0 включно

**якщо** x == matrix[i, j]

**то**

**повернути** "Число X є на позиції {i+1}, {j+1}"

**все якщо**

```
i++
```

**все повторити**

**повернути** "Такого числа у матриці немає"

**Підпрограма** averageUnderMainDiagonal (matrix[,])

```
sum = 0
```

```
numberOfElements = 0
```

**повторити**

**для** i від 0 до matrix.GetLength(0)

**для** j від 0 до matrix.GetLength(1)

**якщо** i > j

**то**

```
sum = sum + matrix[i, j]
```

```
numberOfElements++
```

**все якщо**

**все повторити**

**все повторити**

average = sum / numberOfElements

**повернути** average

**Підпрограма** comparator (a, b)

**якщо**  $a > b$

**то**

**повернути** "Число X більше, ніж середньоарифметичне."

**інакше якщо**  $a < b$

**повернути** "Середньоарифметичне більше, ніж число X."

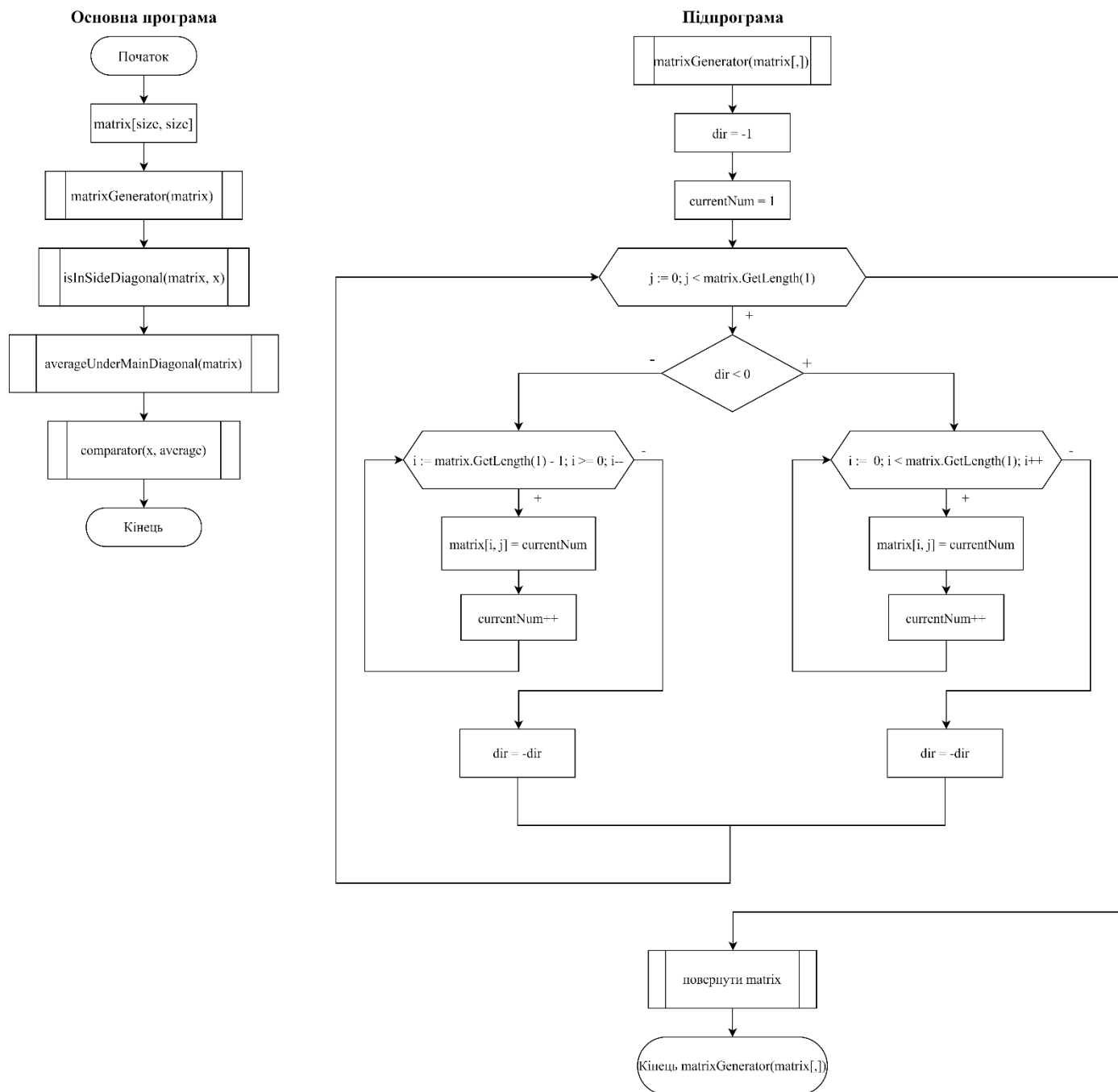
**інакше**

**повернути** "Рівні значення."

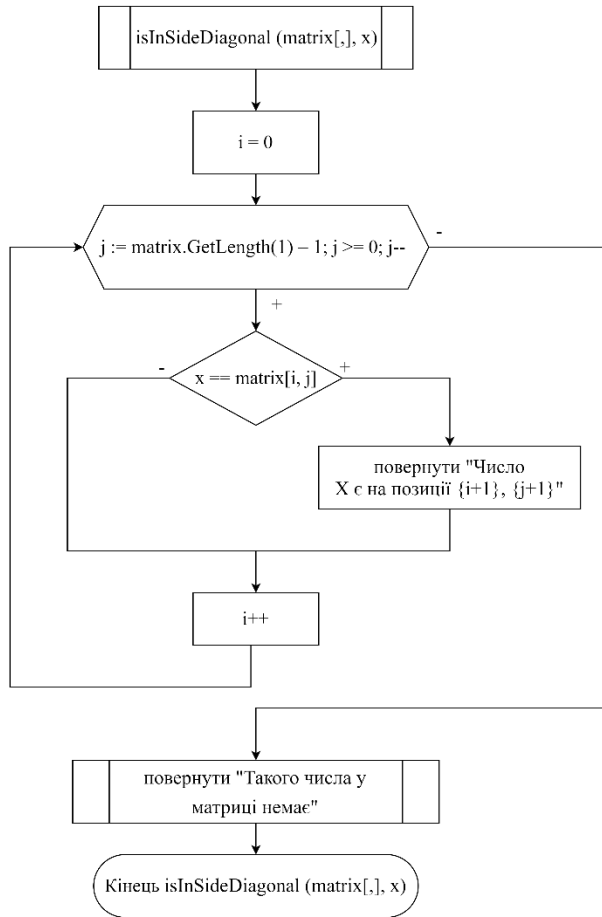
**все якщо**

**Кінець**

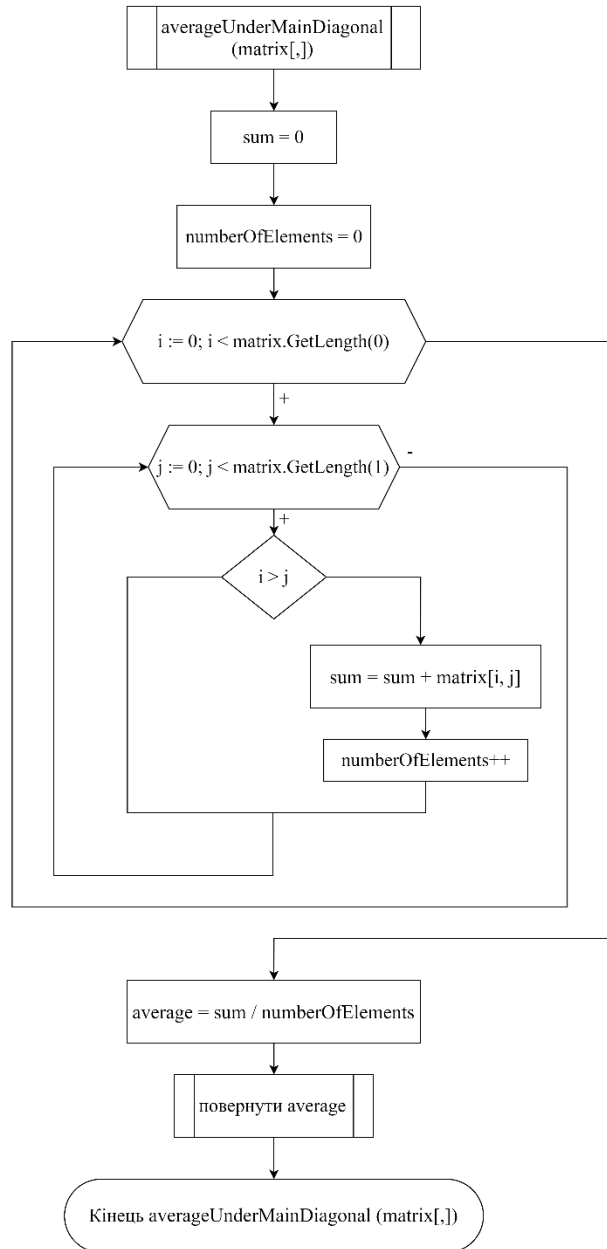
## Блок схема



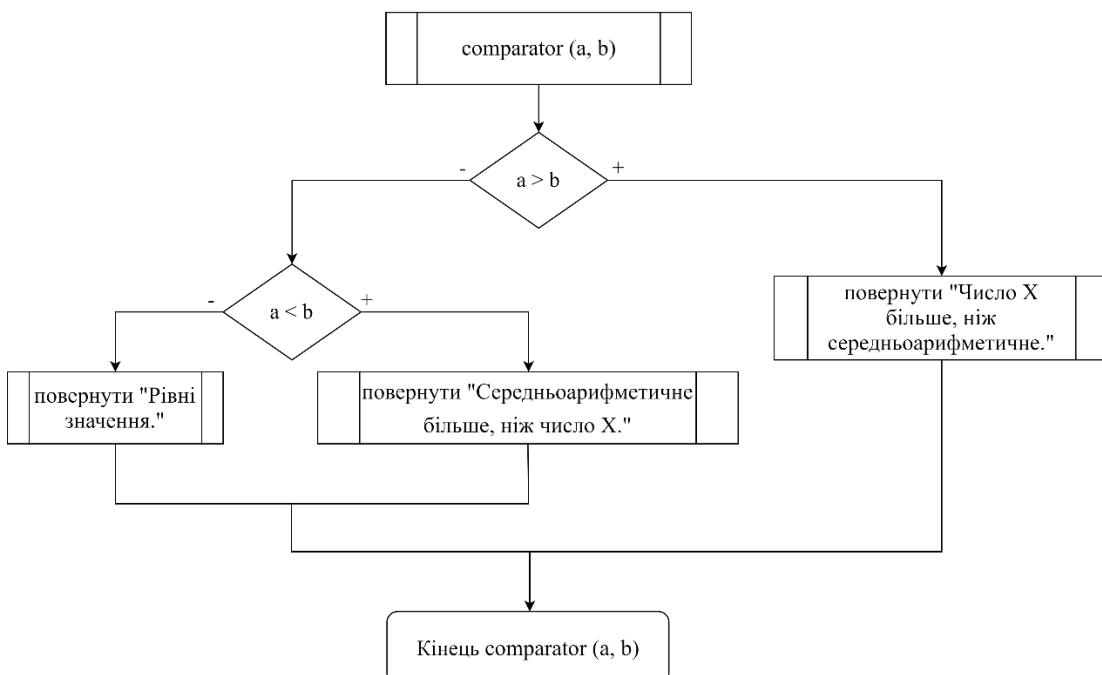
Підпрограма



Підпрограма



Підпрограма





## Код програми (C#)

```

1  using System;
2
3  namespace ConsoleApp1
4  {
5      Ссылка: 0
6      class Program
7      {
8          Ссылка: 0
9          static void Main(string[] args)
10         {
11             Console.Write("Enter the size of the matrix: ");
12             int size = int.Parse(Console.ReadLine());
13             Console.WriteLine();
14
15             double[,] matrix = new double[size, size];
16             matrixGenerator(matrix);
17             printMatrix(matrix);
18
19             Console.Write("Enter the number you'd like to check: ");
20             int x = int.Parse(Console.ReadLine());
21             Console.WriteLine(isInSideDiagonal(matrix, x));
22
23             double average = averageUnderMainDiagonal(matrix);
24             Console.WriteLine(comparator(x, average));
25         }
26
27         Ссылка: 1
28         static double[,] matrixGenerator (double[,] matrix)
29         {
30             int dir = -1;
31             int currentNum = 1;
32             for (int j = 0; j < matrix.GetLength(1); j++)
33             {
34                 if (dir < 0)
35                 {
36                     for (int i = 0; i < matrix.GetLength(0); i++)
37                     {
38                         matrix[i, j] = currentNum;
39                         currentNum++;
40                     }
41                     dir = -dir;
42                 }
43                 else
44                 {
45                     for (int i = matrix.GetLength(0) - 1; i >= 0; i--)
46                     {
47                         matrix[i, j] = currentNum;
48                         currentNum++;
49                     }
50                     dir = -dir;
51                 }
52             }
53             return matrix;
54         }
55
56         Ссылка: 1
57         static void printMatrix(double[,] matrix)
58         {
59             for (int i = 0; i < matrix.GetLength(0); i++)
60             {
61                 for (int j = 0; j < matrix.GetLength(1); j++)
62                 {
63                     Console.Write("{0, 5}", matrix[i, j]);
64                 }
65                 Console.WriteLine("\n");
66             }
67         }
68
69         Ссылка: 1
70         static string isInSideDiagonal(double[,] matrix, double x)
71         {
72             int i = 0;
73             for (int j = matrix.GetLength(1) - 1; j >= 0; j--)
74             {
75                 if (x == matrix[i, j])
76                 {
77                     return $"The number X is in side diagonal in {i + 1} line {j + 1} column.";
78                 }
79                 i++;
80             }
81             return "There is no such number in the matrix.";
82         }
83
84         Ссылка: 1
85         static double averageUnderMainDiagonal(double[,] matrix)
86         {
87             double sum = 0;
88             for (int i = 0; i < matrix.GetLength(0); i++)
89             {
90                 for (int j = 0; j < i; j++)
91                 {
92                     sum += matrix[i, j];
93                 }
94             }
95             return sum / (matrix.GetLength(0) * (matrix.GetLength(0) - 1) / 2);
96         }
97     }
98 }

```

```

83     int numberOfElements = 0;
84     for (int i = 0; i < matrix.GetLength(0); i++)
85     {
86         for (int j = 0; j < matrix.GetLength(1); j++)
87         {
88             if (i > j)
89             {
90                 sum = sum + matrix[i, j];
91                 numberOfElements++;
92             }
93         }
94     }
95     double average = sum / numberOfElements;
96
97     return average;
98 }
99
100 ссылка:1
101 static string comparator (double a, double b)
102 {
103     if (a > b)
104     {
105         return "The first value is bigger than the second.";
106     }
107
108     else if (a < b)
109     {
110         return "The second value is bigger than the first.";
111     }
112
113     else
114     {
115         return "The values are equal";
116     }
117 }
118 }

```

## Випробовування

1)

```

Enter the size of the matrix: 5

 1  10  11  20  21
 2   9  12  19  22
 3   8  13  18  23
 4   7  14  17  24
 5   6  15  16  25

Enter the number you'd like to check: 10
There is no such number in the matrix.
The first value is bigger than the second.

C:\Users\mitya\source\repos\ConsoleApp1\ConsoleApp1\bin\Debug\net5.0\ConsoleApp1.exe (процесс 12396) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...

```

2)

```

Консоль отладки Microsoft Visual Studio
Enter the size of the matrix: 7

 1  14  15  28  29  42  43
 2  13  16  27  30  41  44
 3  12  17  26  31  40  45
 4  11  18  25  32  39  46
 5  10  19  24  33  38  47
 6   9  20  23  34  37  48
 7   8  21  22  35  36  49

Enter the number you'd like to check: 31
The number X is in side diagonal in 3 line 5 column.
The first value is bigger than the second.

C:\Users\mitya\source\repos\ConsoleApp1\ConsoleApp1\bin\Debug\net5.0\ConsoleApp1.exe (процесс 10860) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...

```

## Висновки

Ми дослідили алгоритми обходу масивів, набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій. Як результат, ми отримали алгоритм визначення присутності заданого дійсного числа  $X$  і його місцезнаходження та порівняння його значення із середньоарифметичним значенням елементів під головною діагоналлю матриці, яку було ініціалізовано обходом по стовпцях, розділивши задачу на п'ять кроків: визначення основних дій, ініціалізація `size`, деталізація створення матриці розмірності `[size, size]` та ініціалізація елементів матриці обходом по стовпцях у підпрограмі `matrixGenerator`, деталізація визначення присутності та місцезнаходження елемента  $X$  у побічній діагоналі за допомогою функції `isInSideDiagonal`, деталізація знаходження і порівняння середньоарифметичного значення під головною діагоналлю з  $X$ . В процесі випробовування ми отримали різні результати, в залежності від розмірності матриці і заданого числа  $X$ .