

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 № 8 з
дисципліни «Алгоритми та
структури даних-1. Основи
алгоритмізації»

«Дослідження алгоритмів
пошуку та сортування»

Варіант 34

Виконав студент ІП-13 Шиманська Ганна Артурівна
(шифр, прізвище, ім'я, по батькові)

Перевірила Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 8

Дослідження алгоритмів пошуку та сортування

Мета – дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Варіант 34

№ варіанта	Розмірність	Тип даних	Обчислення значень елементів одновимірного масиву
34	5 x 7	Дійсний	Із середнього арифметичного додатних значень елементів стовпців двовимірного масиву. Відсортувати обміном за спаданням.

- **Постановка задачі**

Необхідно розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису змінної індексованого типу (двовимірний масив).
2. Ініціювання даної змінної.
3. Створення нової змінної індексованого типу (одновимірний масив) та її ініціювання значеннями середнього арифметичного додатних значень елементів стовпців двовимірного масиву.
4. Відсортовування одновимірного масиву за спаданням.

Побудова математичної моделі

Складемо таблицю змінних

Змінна	Тип	Ім'я	Призначення
Двовимірний масив	double	matrix	Проміжні дані

Одновимірний масив	double	array	Проміжні дані
Сума додатніх елементів стовпця	double	columnSum	Проміжні дані
Кількість знайдених елементів	int	counter	Проміжні дані
Лічильник i	int	i	Проміжні дані
Лічильник j	int	j	Проміжні дані
Змінна для запису i-того елемента на ітерації	double	tmp	Проміжні дані

Складемо таблицю функцій

Назва	Синтаксис	Призначення
Округлення до певної кількості знаків після коми в залежності від другого параметра	Round(a, b)	Округлює до b знаків після коми числа a
Генерація випадкового цілочисельного значення у певному діапазоні	Next(a,b)	Генерує ціле число з проміжку [a, b)
Генерація випадкової дробової частини числа	NextDouble()	Генерує правильний десятковий дріб з проміжку (0, 1)

Повернення довжини потрібного виміру багатовимірного масиву	GetLength(a)	Повертає довжину a- го виміру масиву
--	--------------	---

Отже, ми будемо заповнювати двовимірний масив за допомогою підпрограми **FillMatrix**, яка за допомогою двох арифметичних циклів заповнить **matrix** з вимірами 5×7 випадково згенерованими дійсними числами.

Далі використовуючи підпрограму **FindPosArithmetic**, робота якої базується на двох for- та одному if- циклах, заповнимо одновимірний масив **array** середнім арифметичним кожного стовпця матриці.

За допомогою підпрограми **SortArray**, у якій за допомогою двох for- та одного if- циклу реалізовано сортування бульбашкою, відсортуємо масив **array** за спаданням.

Використовуючи підпрограму **DisplayArray**, яка реалізується за допомогою арифметичного for- циклу, виведемо кожен елемент відсортованого масиву **array**.

Розв'язання

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію оголошення двовимірного та одновимірного масивів.

Крок 3. Деталізуємо підпрограму заповнення матриці випадково згенерованими значеннями.

Крок 4. Деталізуємо підпрограму заповнення одновимірного масиву середнім арифметичним додатніх елементів кожного стовпця.

Крок 5. Деталізуємо підпрограму сортування масиву за спаданням.

Крок 6. Деталізуємо підпрограму виведення заповненого одновимірного масиву.

- **Псевдокод алгоритму**

початок

double[][] matrix = new double[5][7];

double[] array = new double[7];

Виклик підпрограми FillMatrix(matrix)

Виклик підпрограми FindPosArithmetic(array, matrix)

Виклик підпрограми SortArray(array)

Виклик підпрограми DisplayArray(array)

кінець

підпрограма FillMatrix(double[][] matrix)

для i від 0 до 5 повторити

для j від 0 до 7 повторити

matrix[i][j] = Math.Round(rand.Next(-100, 100) +
rand.NextDouble(), 2)

все повторити

все потворити

все підпрограма

підпрограма FindPosArithmetic(double[] array, double[][] matrix)

double columnSum

int counter;

для j від 0 до 7 повторити

counter = 0

columnSum = 0

для i від 0 до 7 повторити

якщо matrix[i][j] > 0

columnSum += matrix[i, j]

counter++

все якщо

все повторити

якщо counter != 0

array[j] = columnSum / counter

все якщо

все підпрограма

підпрограма SortArray(double[] array)

double tmp;

для i від 0 до 6 повторити

для j від 0 до 6 повторити

якщо array[j] < array[j + 1]

tmp = array[j]

array[j] = array[j + 1]

array[j + 1] = tmp

все якщо

все повторити

все повторити

все підпрограма

підпрограма DisplayArray(double[] array)

для i від 0 до 7 повторити

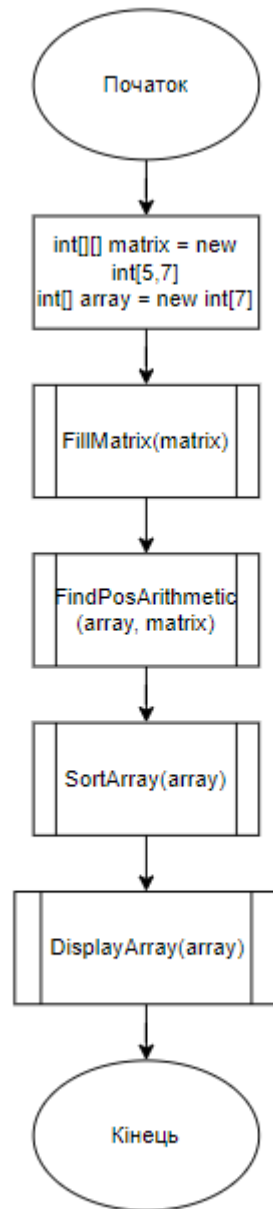
виведення array[i]

все повторити

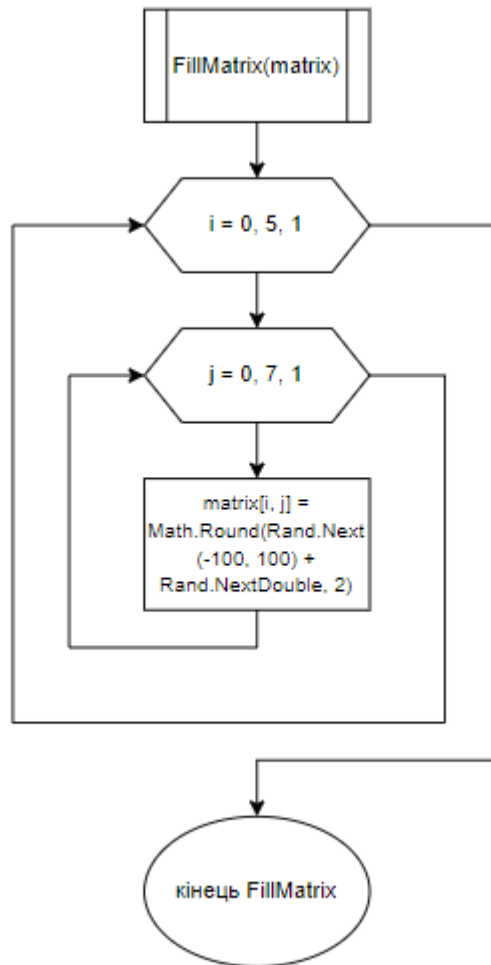
все підпрограма

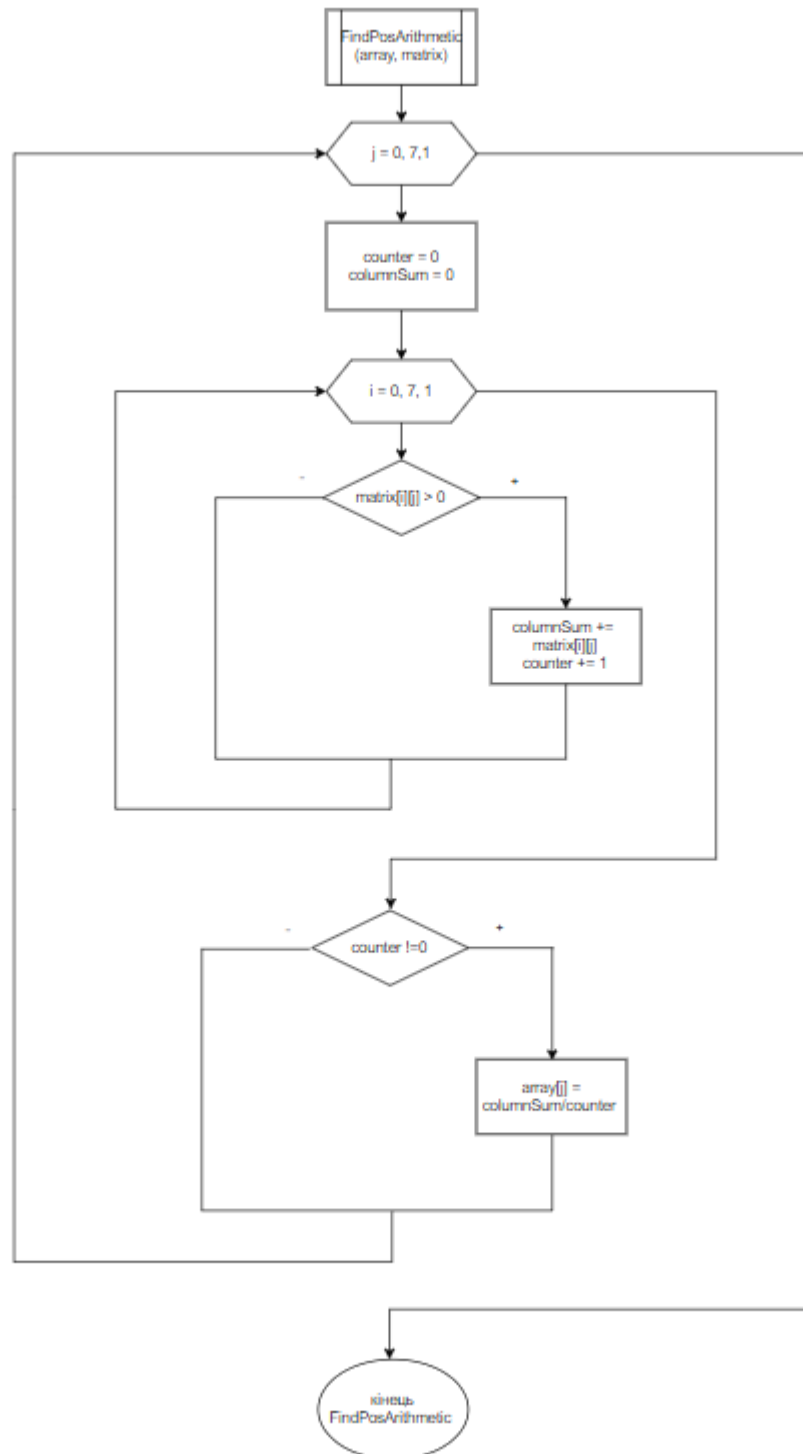
- **Блок-схема**

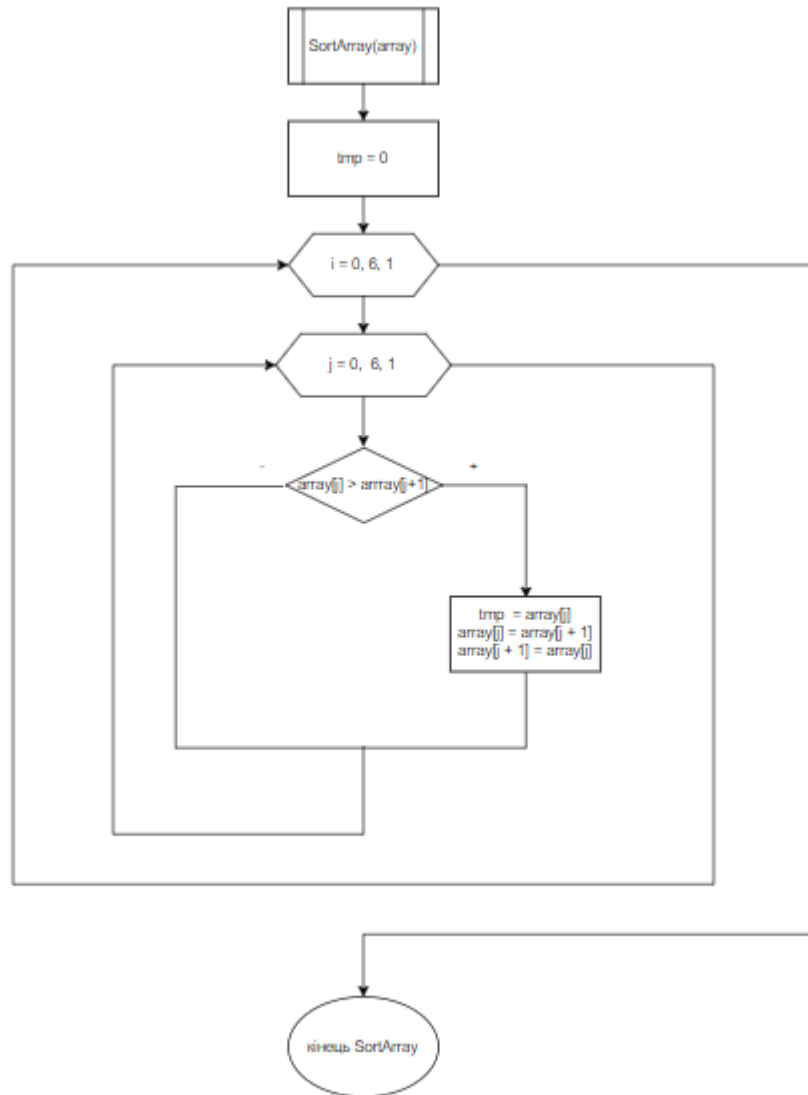
Основна програма

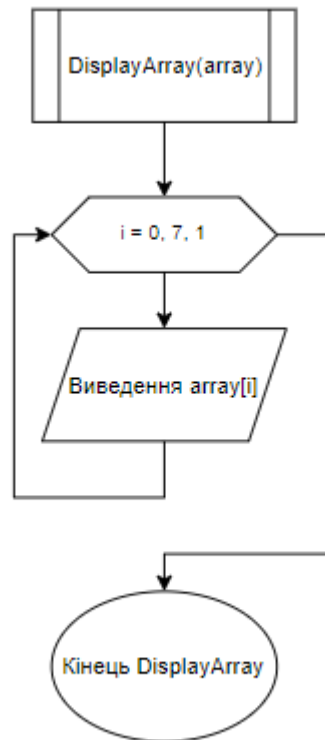


Підпрограми









- Код програми

Основна програма

```
1      using System;
2
3      namespace lab8
4      {
5          class Program
6          {
7              static void Main(string[] args)
8              {
9                  double[,] matrix = new double[5, 7];
10                 double[] array = new double[7];
11                 FillMatrix(matrix);
12                 Console.WriteLine("Середнє арифметичне додатніх значень елементів стовпців: ");
13                 FindPosArithmetic(array, matrix);
14                 SortArray(array);
15                 DisplayArray(array);
16             }
17         }
```

Підпрограми

```
18 static void FillMatrix(double[,] matrix)
19 {
20     Random rand = new Random();
21     for (int i = 0; i < matrix.GetLength(dimension:0); i++)
22     {
23         for (int j = 0; j < matrix.GetLength(dimension:1); j++)
24         {
25             matrix[i, j] = Math.Round(rand.Next(-100, 100) + rand.NextDouble(), 2);
26         }
27     }
28 }
29
```

```
30 static void FindPosArithmetic(double[] array, double[,] matrix)
31 {
32     double columnSum;
33     int counter;
34     for (int j = 0; j < matrix.GetLength(dimension:1); j++)
35     {
36         counter = 0;
37         columnSum = 0;
38         for (int i = 0; i < matrix.GetLength(dimension:0); i++)
39         {
40             if (matrix[i, j] > 0)
41             {
42                 columnSum += matrix[i, j];
43                 counter++;
44             }
45         }
46         if (counter != 0)
47             array[j] = columnSum / counter;
48     }
49 }
50
```

```
51 static void SortArray(double[] array)
52 {
53     double tmp;
54     for (int i = 0; i < array.Length - 1; ++i)
55     {
56         for (int j = 0; j < array.Length - 1; ++j)
57         {
58             if (array[j] < array[j + 1])
59             {
60                 tmp = array[j];
61                 array[j] = array[j + 1];
62                 array[j + 1] = tmp;
63             }
64         }
65     }
66 }
67
```

```

68 static void DisplayArray(double[] array)
69 {
70     for (int i = 0; i < array.Length; i++)
71     {
72         Console.Write("{0,7:f2}", array[i]);
73     }
74 }
75 }
76 }

```

```

-65,60  23,30 -27,83  20,30 -62,35  78,36 -16,05
 75,85  14,58 -69,35 -64,48  72,29 -66,10  28,70
   1,10 -61,11  65,07  52,52  69,12  90,15  87,29
-25,26  62,48 -27,81  83,17 -57,48   2,64 -81,35
   7,32 -73,54   5,90  58,09  -9,00   6,35 -62,64
-----

```

Середнє арифметичне додатніх значень елементів стовпців:
 70,71 58,00 53,52 44,37 35,48 33,45 28,09

```

-2,67  31,49 -53,47 -31,51  88,80  40,08   3,48
-75,67  -4,63 -18,91  44,38  55,54 -49,88  72,61
 -1,37  19,64 -14,88 -17,08 -68,16  71,76   0,98
 49,03 -76,59  74,42   7,28  43,61 -31,45 -16,90
 11,12 -53,88  33,73  99,31 -58,93  29,40 -85,80
-----

```

Середнє арифметичне додатніх значень елементів стовпців:
 62,65 54,08 50,32 47,08 30,07 25,69 25,56

```

 58,32 -98,91  95,31  45,74 -46,36 -23,82  18,55
 82,61  -9,26 -70,98 -78,73  39,06  80,35 -91,74
-84,45 -51,52  80,95  58,13  68,05 -18,49 -59,75
-94,82 -72,02 -40,44 -64,72  66,17 -27,67 -79,48
 24,94 -63,38 -46,54  33,72 -19,10 -45,36  51,78
-----

```

Середнє арифметичне додатніх значень елементів стовпців:
 88,13 80,35 57,76 55,29 45,86 35,16 0,00

• Висновки:

Виконуючи лабораторну роботу, я дослідила особливості алгоритмів пошуку та сортування, набула практичних навичок їх використання під час складання програмних специфікацій. Зокрема, я створила програму для знаходження середнього арифметичного додатніх елементів стовпців матриці та відсортувала

отриманий масив значень бульбашкою. Після сортування елементи розмістилися за спаданням. На третьому тестуванні помітно також, що у випадку, коли в стовпці матриці немає додатніх елементів, у масив передається значення “0” на місці відповідної суми.