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 лінійних алгоритмів»

Варіант 24

Виконав студент ІП-12 Орищенко Ярослав Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірив \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Лабораторна робота 8

Дослідження алгоритмів перетворення матриць та отримання агрегатних значень

Мета: дослідити підходи до пошуку та перетворення на матрицях та набути практичних навичок використання укладених керувальних дій повторення і їх з’єднання під час складання програмних специфікацій.

**Розв’язок:**

1. ***Постановка задачі***. Необхідно створити двовимірний масив розмірністю 7х7 й заповнити його випадковими цілими значеннями. Із значень побічної діагоналі утворюємо одновимірний масив розмірністю 7 і відсортовуємо його методом вставки від меншого елемента до більшого.
2. ***Математична модель*.** Складемо таблицю імен змінних.

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Розмір двовимірного масиву | Цілий | size | Проміжне дане |
| Двовимірний масив | Цілий | arr1 | Проміжне дане |
| Масив із значень побічнох діагоналі | Цілий | arr2 | Результат |

1. ***Псевдокод***

*Крок 1 Крок 2*

**початок** s size = 7 г генерація матриці с створення одновимірного масиву с сортування масиву **кінець**

**початок** s size = 7

printMatrix(arr1, size)с с створення одновимірного масиву с сортування масиву **кінець**

*Крок 3 Крок 4*

**початок** s size = 7

printMatrix(arr1, size)с с d diagonalArray(arr1, arr2, size)

printArray(arr2, size)с

сортування масиву  **кінець**

**початок** s size = 7

printMatrix(arr1, size)с с d diagonalArray(arr1, arr2, size)

printArray(arr2, size)с

insertionSort(arr2, size)

printArray(arr2, size)

**кінець**

***Псевдокод підпрограм***

*printMatrix diagonalArray*

**printMatrix**(\*\*arr, size) п **повторити** для i від 0 до size **п повторити** для j від 0 до size a arr[i][j] = rand()%9+1 в виведення arr[i][j] в **все повторити** в **все повторити кінець** printMatrix

**diagonalArray**(\*\*arr1, \*arr2, size) k j = size -1 п **п п повторити** для i від 0 до size a arr2[i] = arr1[i][j] t j j--в **в все повторити кінець** diagonalArray

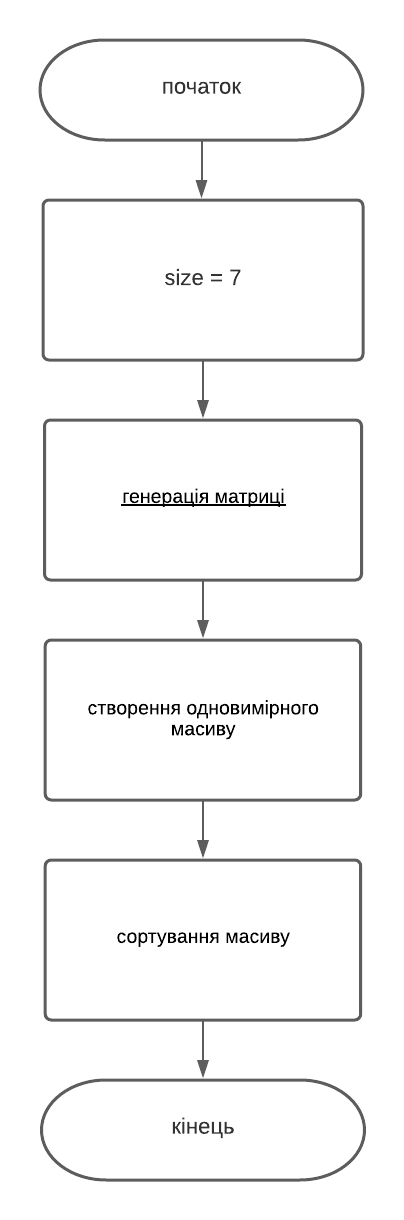
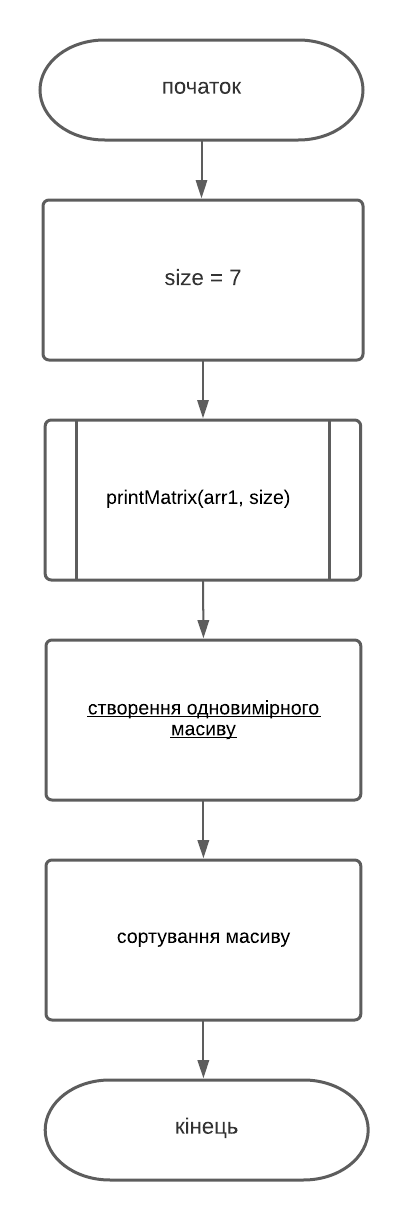
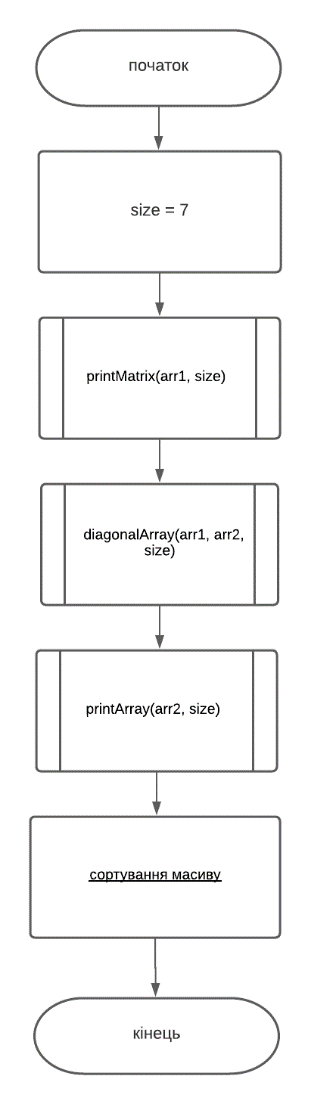
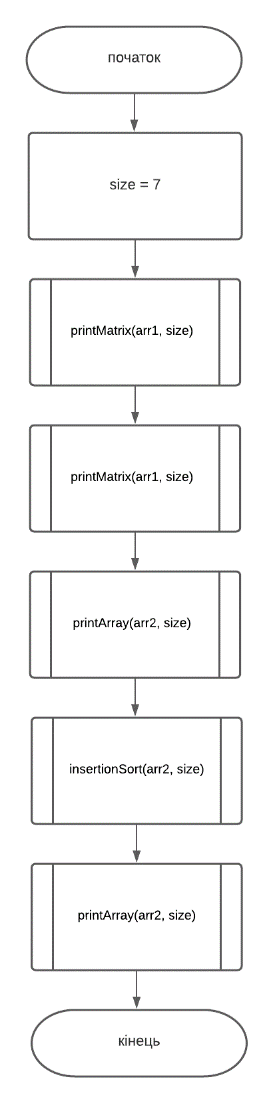
*printArray insertionSort*

**printArray**(\*arr, size) п **повторит**и для і від 0 до size в виведення arr[i] в **все повторити** **кінець** printArray

**insertionSort**(\*arr, size) п **повторити** для i від 1 до size t temp = arr[i] п **повторити** для j від i – 1 поки j >=0 & && arr[j] > temp з кроком -1 a arr[j+1] = arr[j] в **все повторити** a arr[j+1] = temp в **все повторити** **кінець** insertionSort

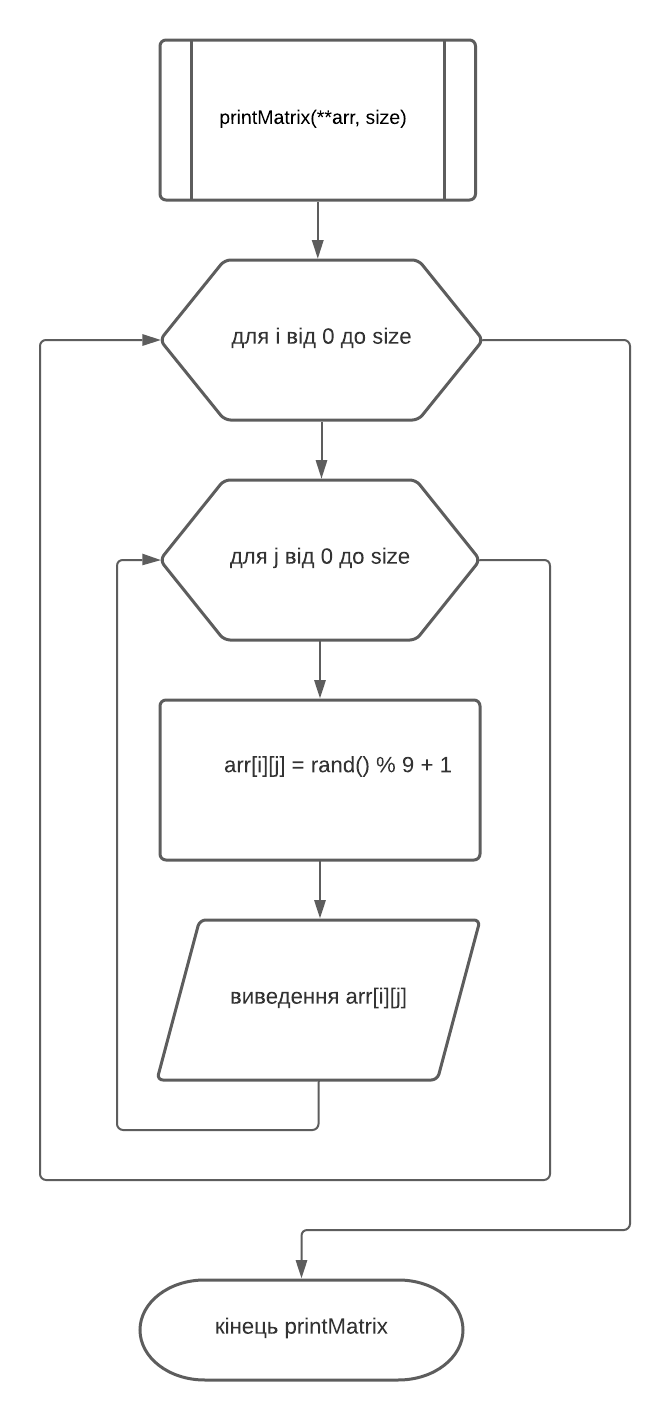
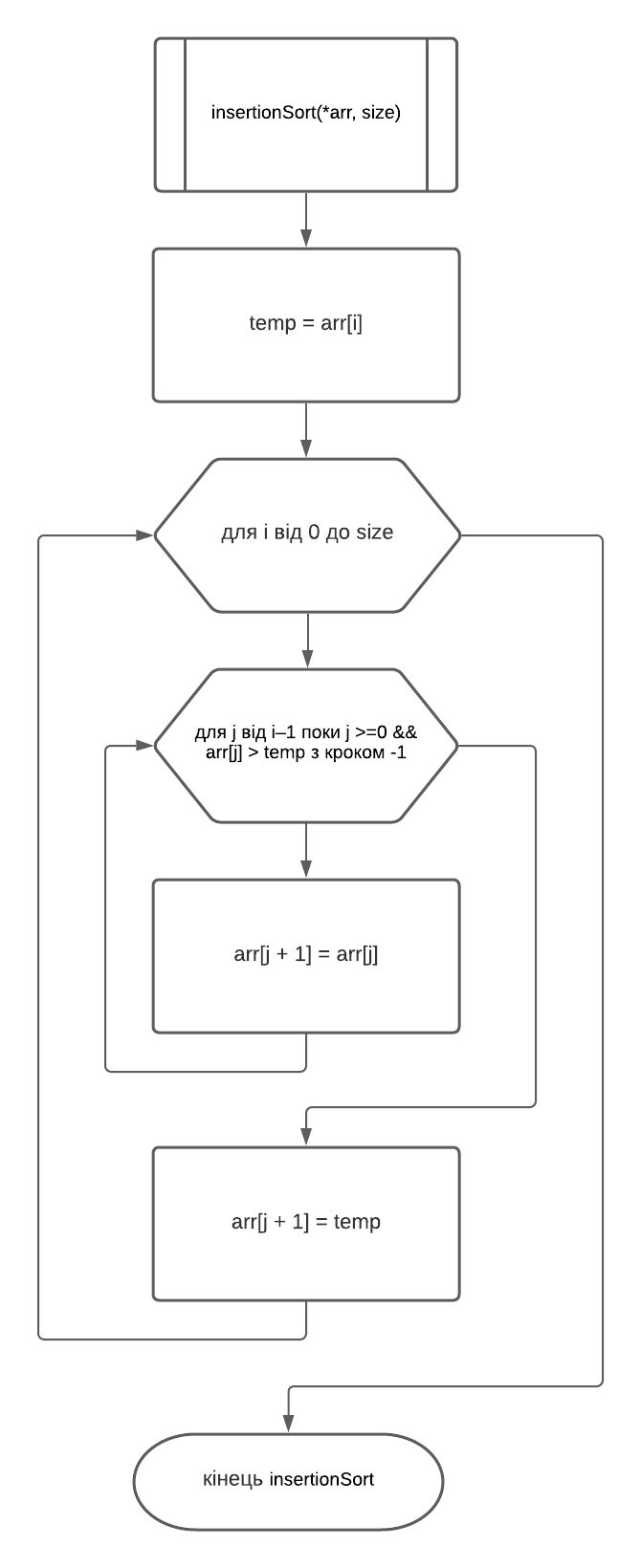
1. ***Блок-схема***

*Крок 1 Крок 2 Крок 3 Крок 4*

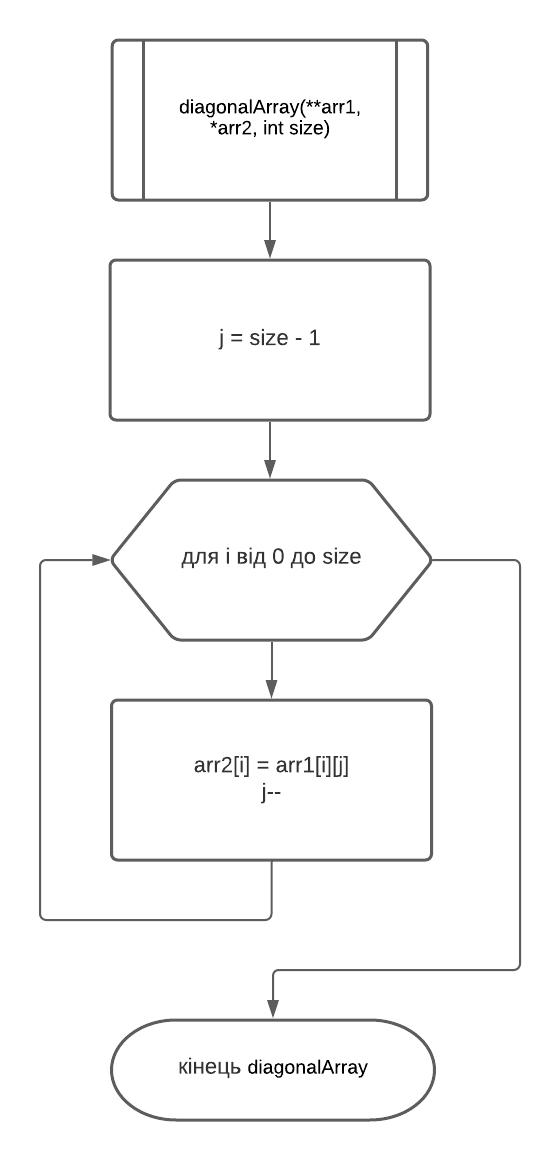
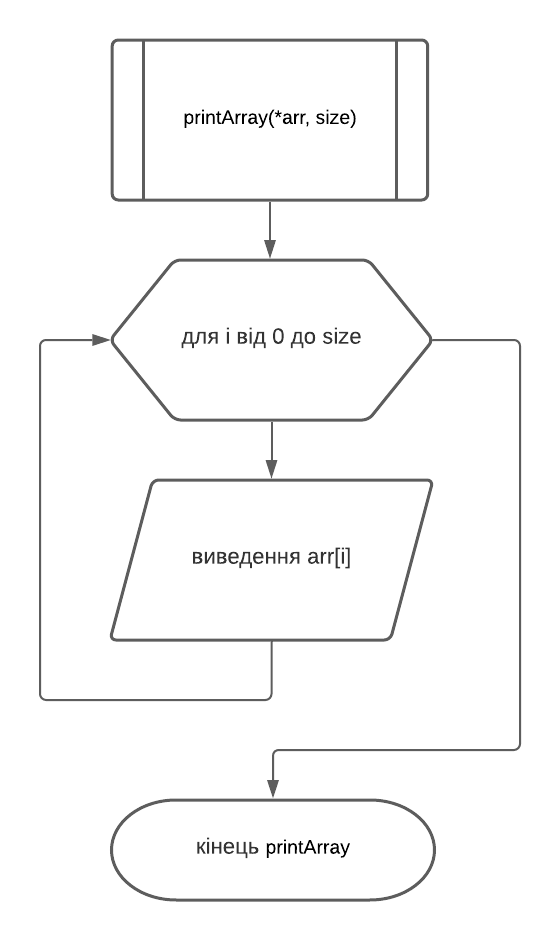
*   *

***Блок-схема підпрограм***

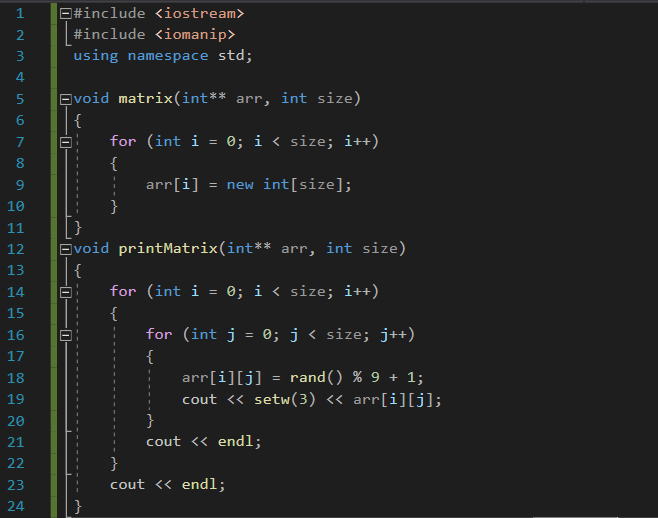
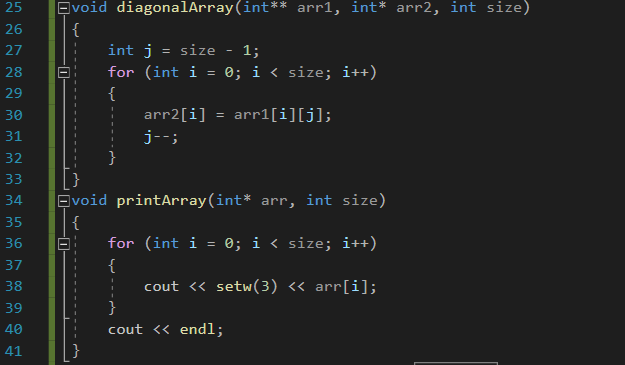
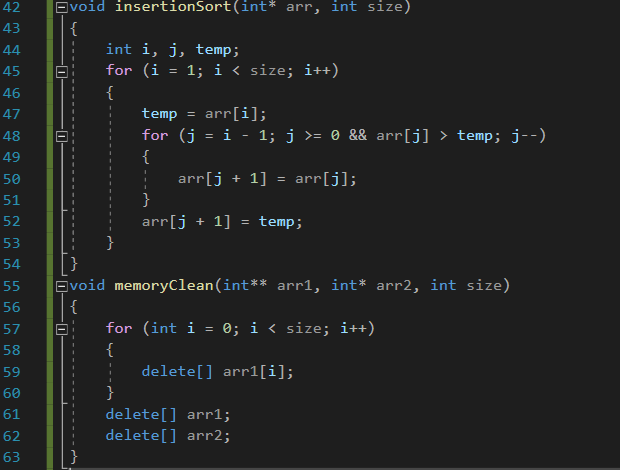
*printMatrix insertionSort*

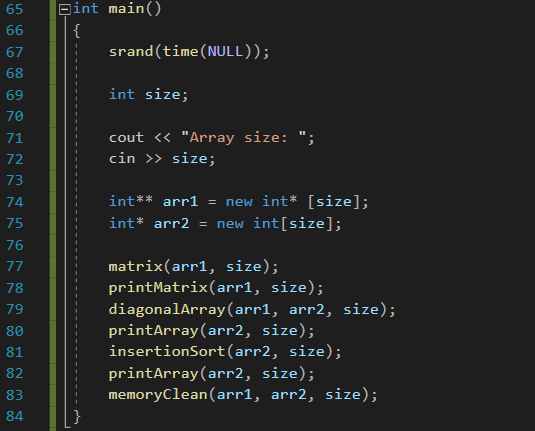
******  ******

*diagonalArray printArray*

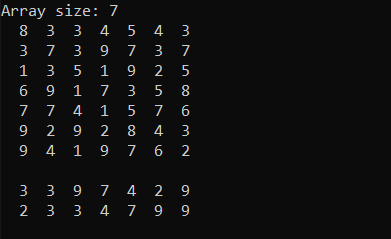
** **

1. ***Код програми***



1. ***Перевірка алгоритму***



1. ***Висновки***

Було досліджено підходи до пошуку та перетворення на матрицях та набуто практичних навичок використання укладених керувальних дій повторення і їх з’єднання під час складання програмних специфікацій.