

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 11

Виконав студент ІП-13, Дем'янчук Олександр Петрович
(шифр, прізвище, ім'я, по батькові)

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

Лабораторна робота 7

Дослідження алгоритмів пошуку та сортування

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

Індивідуальне завдання

Варіант 11

Завдання

№ варіанта	Розмірність	Тип даних	Обчислення значень елементів одновимірного масиву
11	7 x 5	Дійсний	Із суми від'ємних значень елементів рядків двовимірного масиву. Відсортувати методом вставки за зростанням.

1. Постановка задачі

Ініціалізуємо двовимірний масив дійсного типу розмірністю 7 x 5, що заповнюється випадковими числами. Ініціалізуємо одновимірний масив, що заповнюємо елементами, які рівні сумі від'ємних значень елементів рядків суми. Після цього сортуємо одновимірний масив методом вставки за зростанням

2. Математична модель

Побудуємо таблицю імен змінних:

Змінна	Тип	Ім'я	Призначення
Двовимірний масив	Дійсний	matrix	Проміжні дані
Одновимірний масив	Дійсний	array	Вихідні дані
Кількість рядків	Цілий	rows	Проміжні дані
Кількість стовпців	Цілий	columns	Проміжні дані
Генератор випадкових чисел	Дійсний	випадкове[-10.0; 10.0]	Проміжні дані

Таблиця функцій:

<i>Змінна</i>	<i>Ім'я</i>
Заповнення матриці	<code>fill_matrix</code>
Заповнення масиву	<code>fill_array</code>
Пошук суми від'ємних ел. рядка	<code>negatives_sum</code>
Виведення елементів матриці	<code>output_matrix</code>
Виведення елементів масиву	<code>output_array</code>
Сортування вставкою	<code>insertion_sort</code>

Розмірність матриці задається у змінних **rows = 7, columns = 5** відповідно.

Створення та заповнення двовимірної масиви відбувається через підпрограму **fill_matrix**, що заповнюється випадковими дійсними числами.

Виведення елементів матриці у підпрограмі **output_matrix** реалізоване для перегляду проміжних результатів та оцінки роботи алгоритму.

Підпрограмою **fill_array** ініціалізуємо та заповнюємо одновимірний масив. Елементи для заповнення масиву отримуємо за допомогою підпрограми **negaitves_sum**, що обчислює суму від'ємних елементів рядку матриці.

Після заповнення одновимірної масиви сортуємо його елементи методом вставки через підпрограму **insertion_sort**.

Виведення одновимірної масиви реалізовано у підпрограмі **output_array**.

Розв'язання:

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

Крок 1. Визначимо основні дії;

Крок 2. Деталізуємо дію визначення розмірності матриці через змінні **rows, columns**;

Крок 3. Деталізуємо підпрограму ініціалізації та заповнення двовимірної масиви **fill_matrix**;

Крок 4. Деталізуємо підпрограму ініціалізації та заповнення одновимірної масиви **fill_array**;

Крок 5. Деталізуємо підпрограму пошуку суми від'ємних елементів рядку матриці **negatives_sum**;

Крок 6. Деталізуємо підпрограму методу сортування вставкою **insertion_sort** та її виклик у підпрограмі **fill_array**;

Крок 7. Деталізуємо підпрограму виведення одновимірного масиву **output_array**

Псевдокод:

Крок 1

початок

rows := 7, columns := 5

виклик підпрограми **fill_matrix** для створення і заповнення двовимірного масиву

виклик підпрограми **output_matrix** для виведення елементів двовимірного масиву

виклик підпрограми **fill_array** для створення і заповнення одновимірного масиву

виклик підпрограми **negaitves_sum** всередині підпрограми **fill_array** для
обчислення елементів масиву

виклик підпрограми **insertion_sort** для сортування отриманого одновимірного
масиву

виклик підпрограми **output_array** для виведення одновимірного масиву

кінець

Крок 7

початок

rows := 7

columns := 5

fill_matrix(matrix[[]], rows, columns)

output_matrix(matrix[[]], rows, columns)

fill_array(array[], rows)

insertion_sort(array[], rows)

output_array(array[], rows)

кінець

підпрограма fill_matrix(matrix[[]], row, col)

повторити для і від 0 до row з кроком 1

повторити для j від 0 до col з кроком 1

matrix[row][col] = випадкове[-10.0; 10.0]

все повторити

все повторити

все підпрограма

підпрограма output_matrix(matrix[[]], row, col)

повторити для і від 0 до row з кроком 1

повторити для j від 0 до col з кроком 1

вивести matrix[row][col]

все повторити

все повторити

все підпрограма

підпрограма negatives_sum(matrix[i][j], i, col)

sum := 0

повторити для j від 0 до col з кроком 1

якщо matrix[i][j] < 0 то

sum := sum + matrix[i][j]

все якщо

все повторити

return sum

все підпрограма

підпрограма fill_array(array[i], row)

повторити для i від 0 до row з кроком 1

array[i] := negatives_sum(matrix, i, col)

все повторити

все підпрограма

підпрограма insertion_sort(array[], row)

повторити для i від 1 до row з кроком 1

tmp := array[i]

j = i - 1

поки tmp < array[j] && j >= 0

array[j+1] = array[j]

все поки

array[j+1] := tmp

все повторити

все підпрограма

підпрограма output_array(array, row)

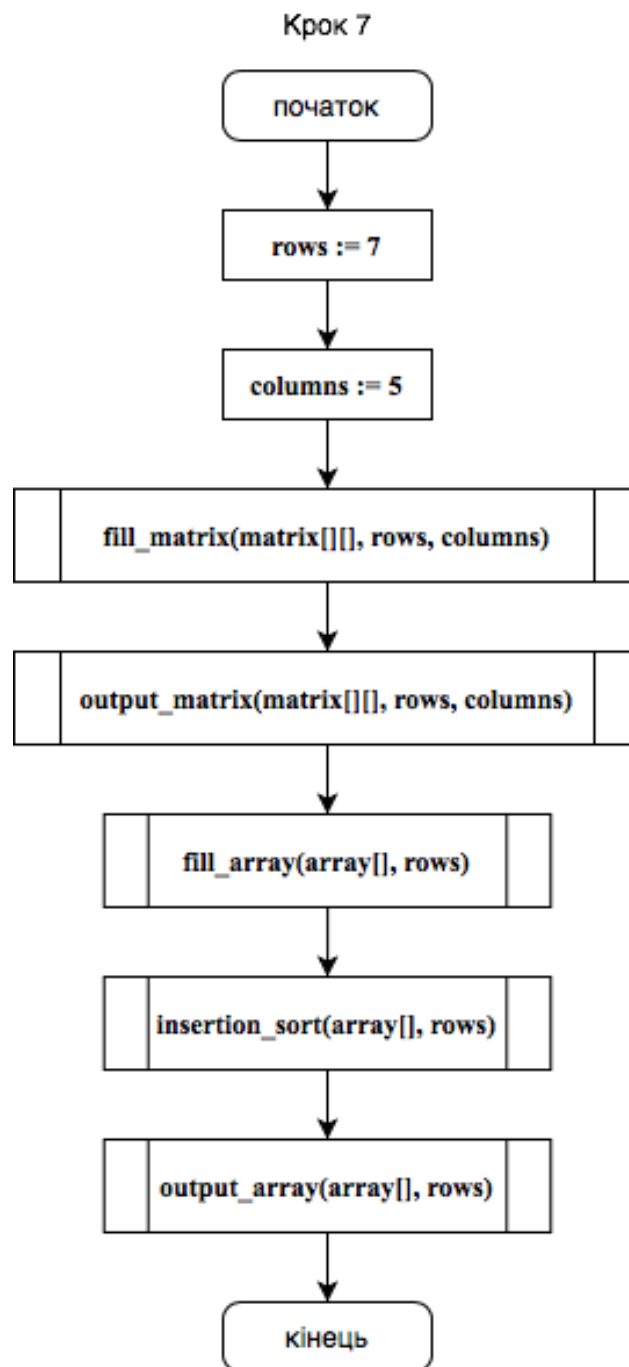
повторити для i від 0 до row з кроком 1

виведення array[i]

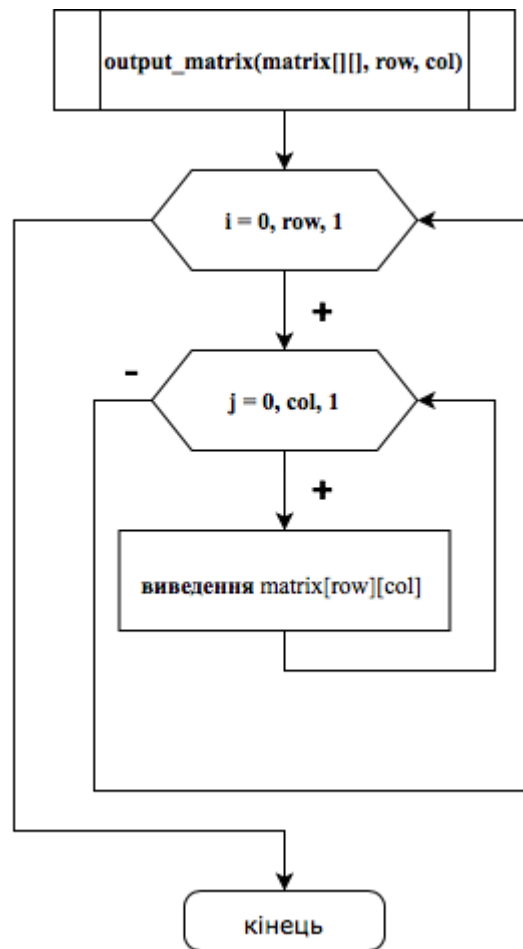
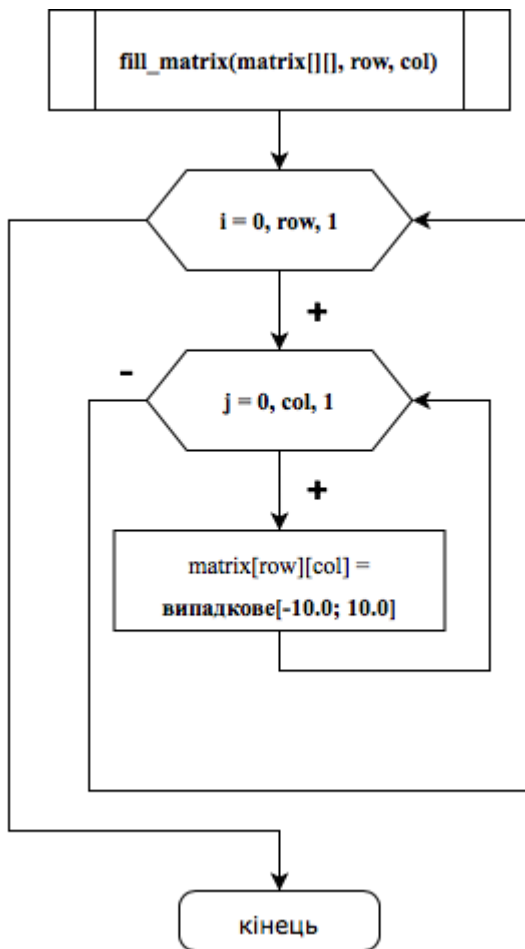
все повторити

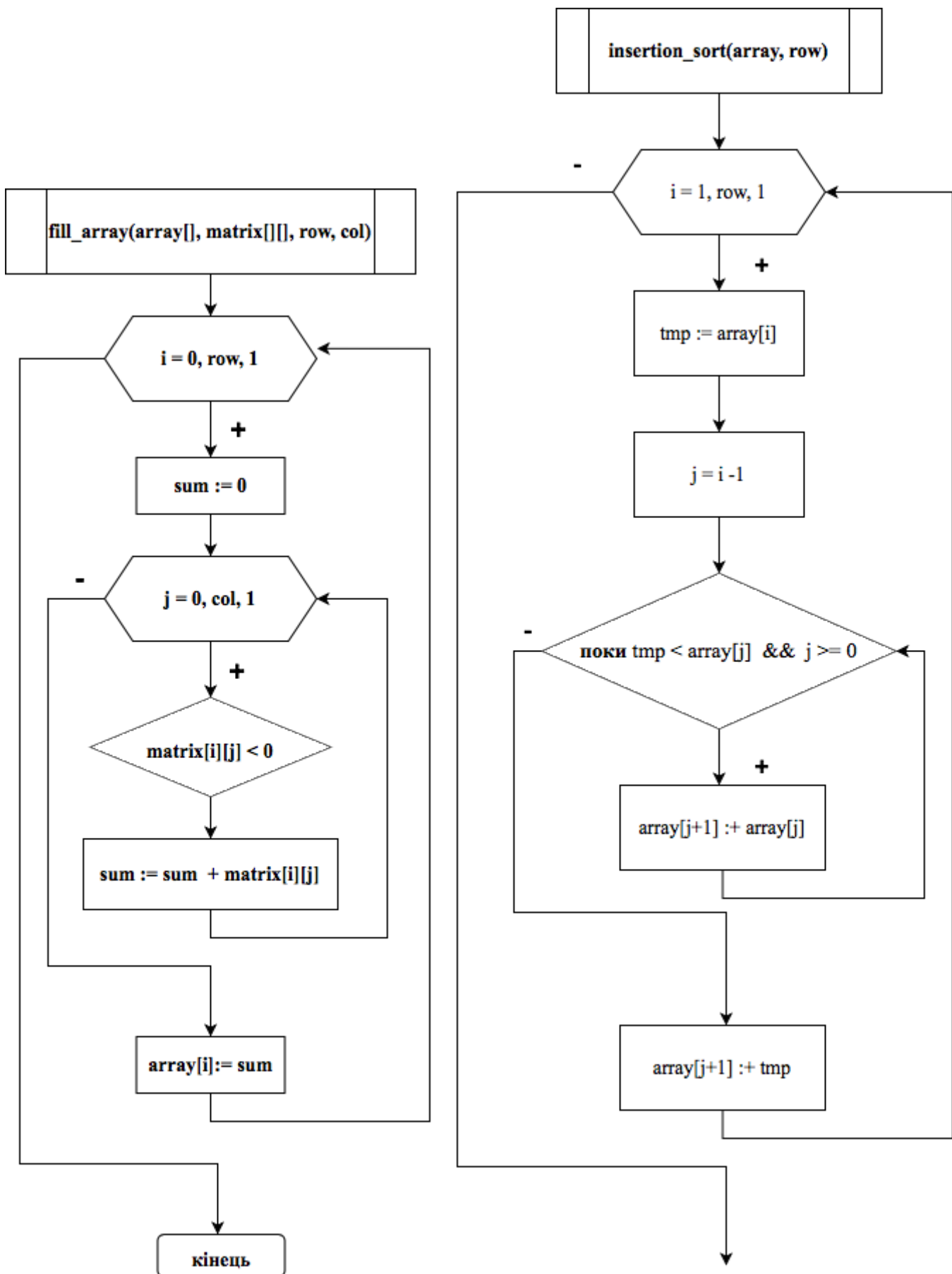
все підпрограма

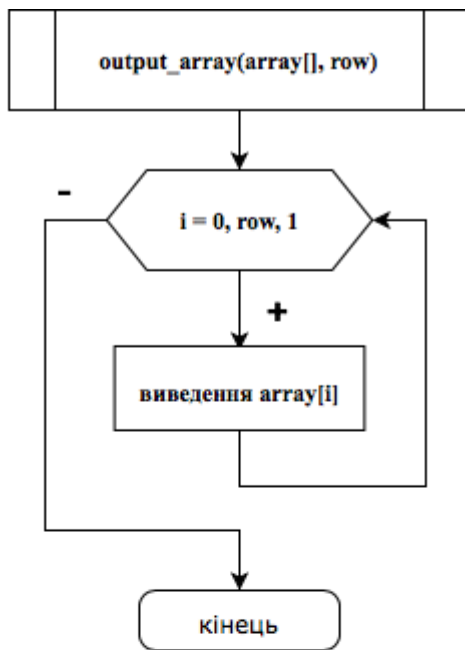
Блок-схема



Основи_програмування – 1. Алгоритми та структури даних







Програмний код

Висновок

На лабораторній роботі дослідив методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набув практичних навичок їх використання під час складання програмних специфікацій.