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 лінійних алгоритмів»

Варіант 24

Виконав студент ІП-12 Орищенко Ярослав Олександрович

(шифр, прізвище, ім'я, по батькові)

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

Лабораторна робота 9

Дослідження складних циклічних алгоритмів

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

**Розв’язок:**

1. ***Постановка задачі***. Для розв’язку задачі необхідно створити двовимірний квадратний динамічний масив, після чого за допомогою обходу по стовпчикам знайти перший елемент, менший за нуль. Далі порівняти його з усіма елементами, які знаходяться нижче побічної діагоналі і підрахувати кількість елементів, які менші за нього.
2. ***Математична модель*.** Складемо таблицю імен змінних.

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Розмір масиву | Цілий | size | Проміжне дане |
| Матриця елементів | Дійсний | arr | Проміжне дане |
| Перший від’ємний елемент матриці | Дійсний | number | Результат |
| Кількість елементів під побічною діагоналлю менших за number | Цілий | amount | Результат |

1. ***Псевдокод***

**початок**  в введення size f fillArray(arr, size) n number = firstNegative(arr, size) a amount = lessThenFirst(arr, size, number) **кінець**

***Псевдокод підпрограм***

fillArray firstNegative

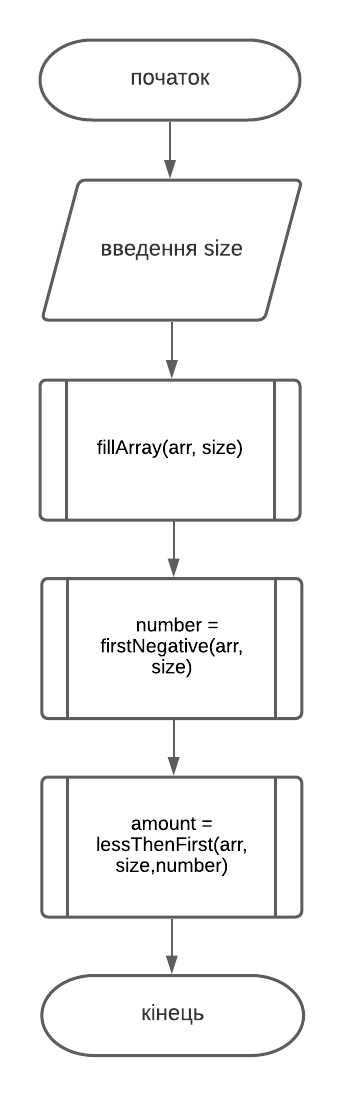
**fillArray**(arr, const SIZE) п **п п повторити** для i від 0 до SIZE **п повторити** для j від 0 до size a a arr[i] = rand() / RAND\_MAX \* 2 \* 50 - 50в в виведення arr[i] **в все повторити**  **в в в все** **повторити** **кінець fillArray**

**firstNegative**(arr, size) i і = 0 j j = 0 **я якщо** arr[0][0] < 0 в виведення arr[0][0], i+1, j+1 r return arr[0][0] **і інакше** **п поки** arr[i][j] >= 0 && j < size **п поки** arr[i][j] >= 0 && i < size і i++ **в все поки** **я якщо** arr[i][j] < 0 **в виведення** arr[i][j], i+1, j+1 **і інакше** і i = 0 j j++ **в все якщо в все поки** **в все якщо** r return arr[i][j] **к кінець firstNegative**

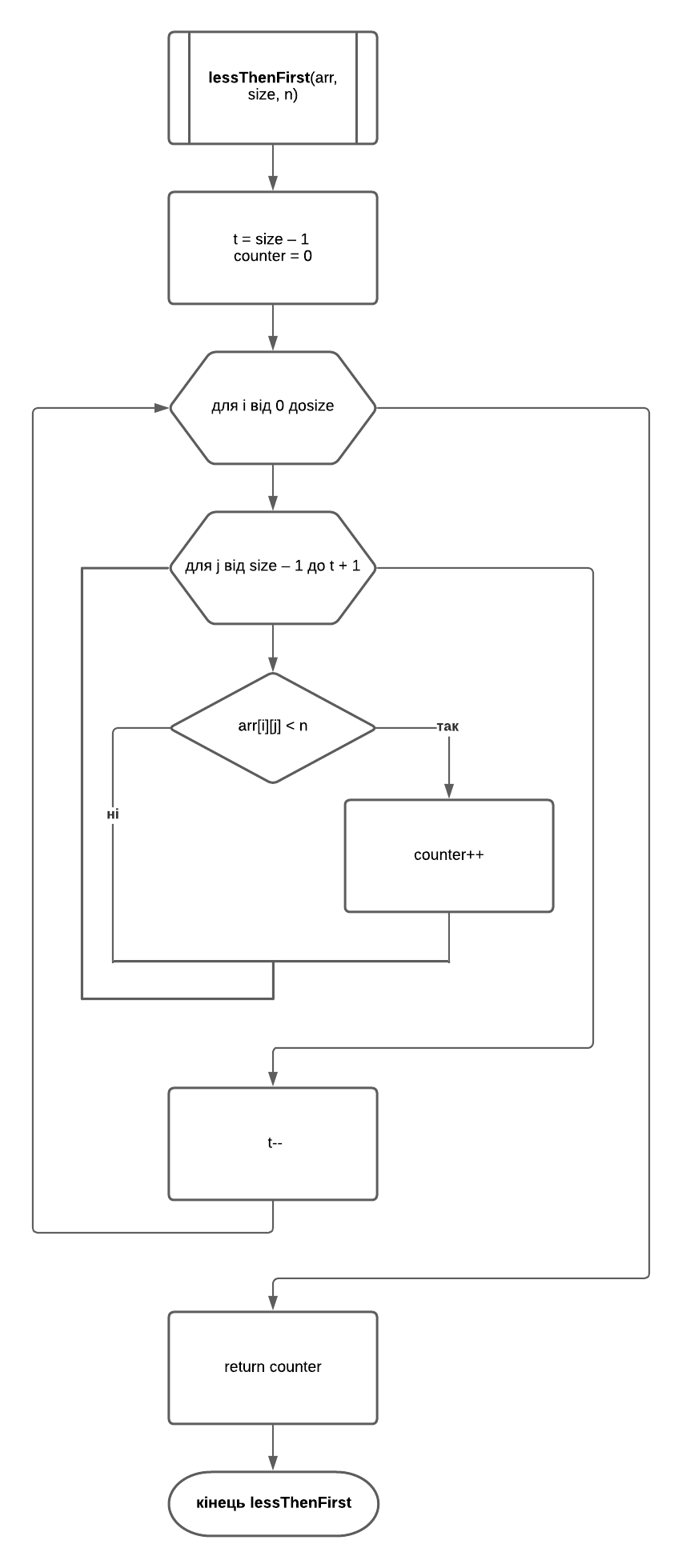
lessThenFirst

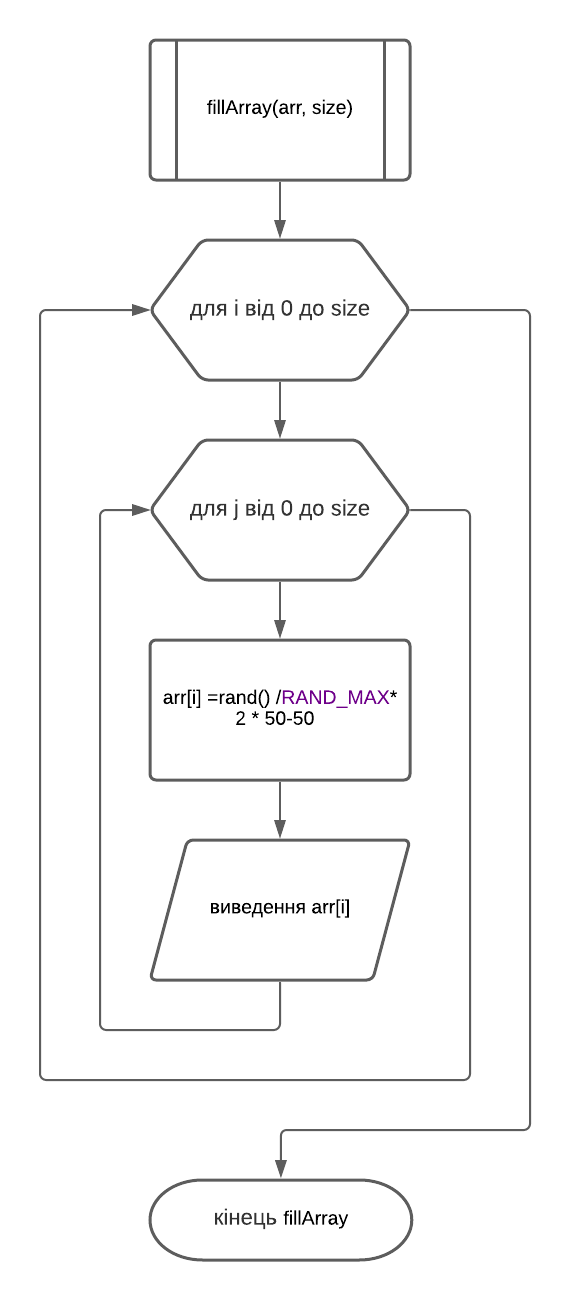
**lessThenFirst**(arr, size, n) t t = size – 1 c counter = 0 **п повторити** для і від 0 до size **п повторити** для j від size – 1 до t + 1 **я якщо** arr[i][j] < n c сounter++ **в все якщо вс все повторити** t t-- **в в все повторити** r return counter **кінець lessThenFirst**

1. ***Блок-схема***

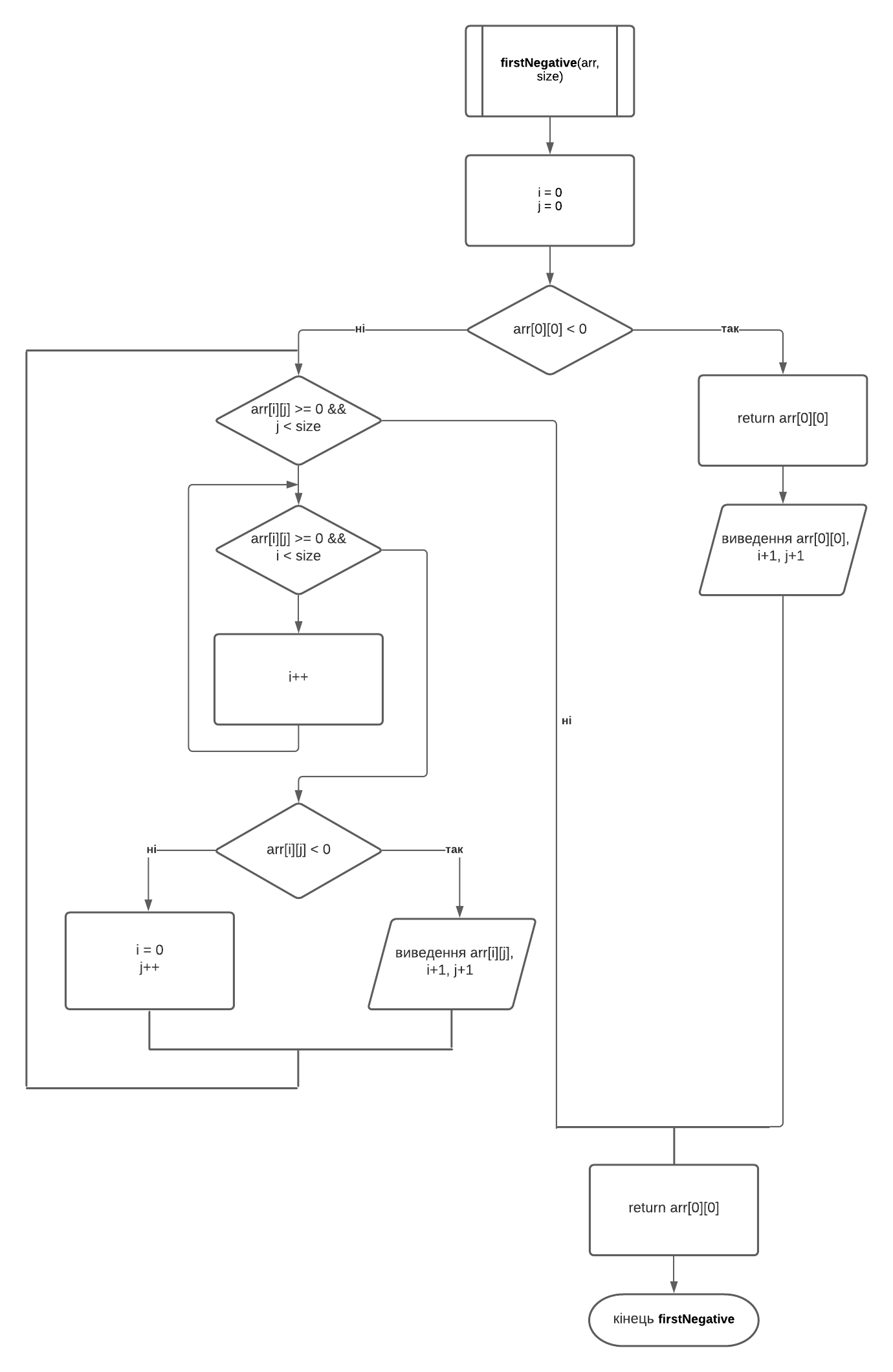


***Блок-схема підпрограм***

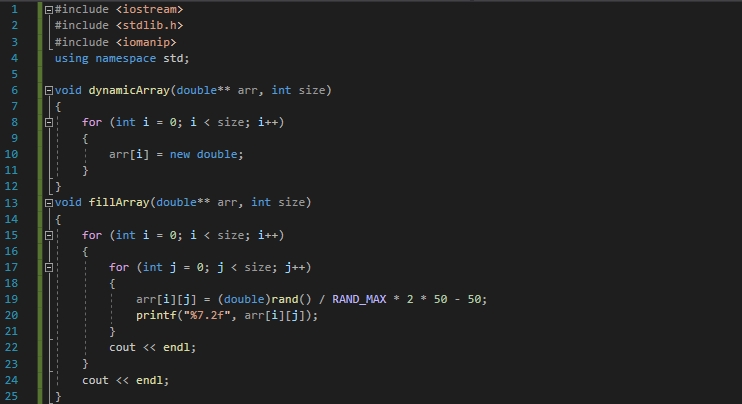
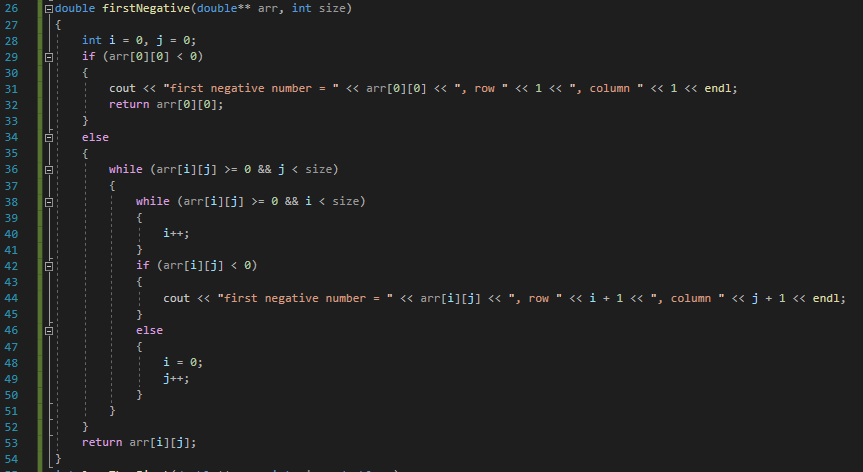
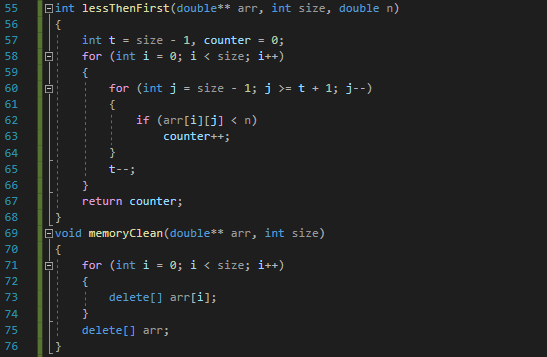
fillArray lessThenFirst ******

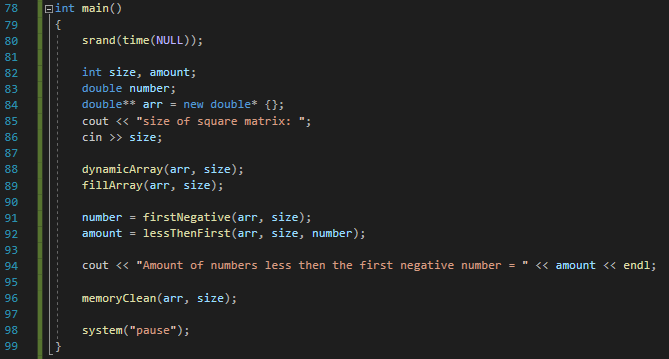


firstNegative

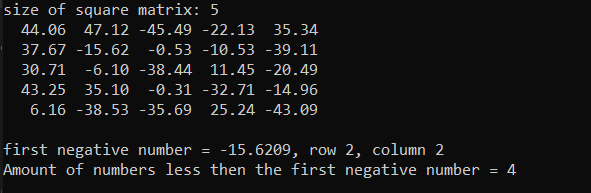
******

1. ***Код програми***



1. ***Перевірка алгоритму***



1. ***Висновки.*** Було досліджено алгоритми обходу масивів, набуто практичних навичок використання цих алгоритмів під час складання програмних специфікацій.