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 22

Виконав студент ІП-13, Музичук Віталій Андрійович

(шифр, прізвище, ім'я, по батькові)

Перевірила Вєчерковська Анастасія Сергіївна

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 6  
Дослідження алгоритмів пошуку та сортування**

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

**Варіант 22**

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

1. Опису змінної індексованого типу (двовимірного масиву) згідно з варіантом.

2. Ініціювання змінної, що описана в п.1 даного завдання.

3. Створення нової змінної індексованого типу (одновимірний масив) та її ініціювання значеннями, що обчислюються згідно з варіантом (табл. 1).



**Постановка задачі**

Спочатку ми створюємо двовимірний масив із 5 рядками та 7 стовпцями. Ініціалізація цієї змінної відбувається випадково числами з проміжку [-9, 9] за допомогою підпрограми initMatrix(). Далі за допомогою підпрограми outputMatrix() виводимо матрицю в консоль для наглядності. Після цього створюємо окрему змінну array[] та заповнюємо її середнім арифметичним кожного стовпця за допомогою підпрограми averageColumn() та арифметичного циклу. Сортуємо цей масив методом «бульбашки» по спаданню й виводимо в консоль.

**Побудова математичної моделі**

Складемо таблицю змінних

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Двовимірний масив | Дійсний | *matrix* | Проміжне значення |
| Одновимірний масив | Дійсний | *array* | Проміжне значення |
| Підпрограма для ініціалізації матриці | void | *initMatrix* | Ініціалізація матриці |
| Підпрограма для виведення матриці | void | *outputMatrix* | Виведення матриці в консоль |
| Підпрограма для визначення середнього арифметичного стовпця | Дійсний | *averageColumn* | Визначення середнього арифметичного стовпця |
| Підпрограма для сортування масиву | void | *sort* | Сортування масиву |

**initMatrix –** підпрограма яка заповнює матрицю випадковими числами з проміжку [-9; 9].

**outputMatrix** – підпрограма, яка поелементно виводить в консоль дані з матриці за допомогою 2 ітераційних циклів.

**averageColumn -** підпрограма яка повертає середнє арифметичне значення додатних елементів матриці.

**sort –** підпрограма призначена для сортування одновимірного масиву методом обміну («бульбашки»).

**Rand**(a, b) – функція яка генерує випадкове число на проміжку [a, b]

**Розв’язання**

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

*Крок 1*. Визначаємо основні дії  
*Крок 2.* Створення й ініціалізація матриці  
*Крок 3.* Виведення матриці  
*Крок 4.* Ініціалізація й заповнення одновимірного масиву  
*Крок 5.* Сортування масиву  
*Крок 6.* Виведення масиву

**Псевдокод**

*крок 1***початок**

створення й ініціалізація матриці  
виведення матриці  
ініціалізація й заповнення одновимірного масиву  
сортування масиву  
виведення масиву

**кінець**

*крок 2*

**початок**

strings = 5  
columns = 7matrix [strings, columns]  
initMatrix(matrix)  
виведення матриці  
ініціалізація й заповнення одновимірного масиву  
сортування масиву  
виведення масиву

**кінець**

*крок 3*

**початок**

strings = 5  
columns = 7matrix [strings, columns]  
initMatrix(matrix)  
outputMatrix(matrix)  
ініціалізація й заповнення одновимірного масиву  
сортування масиву  
виведення масиву

**кінець**

*крок 4*

**початок**

strings = 5  
columns = 7matrix [strings, columns]  
initMatrix(matrix)  
outputMatrix(matrix)  
array[columns]  
**для** і **від** 1 **до** columns

array[i] = averageColumn(matrix, i);

**все повторити**  
сортування масиву  
виведення масиву

**кінець**

*крок 5*

**початок**

strings = 5  
columns = 7matrix [strings, columns]  
initMatrix(matrix)  
outputMatrix(matrix)  
array[columns]  
**для** і **від** 1 **до** columns

array[i] = averageColumn(matrix, i);

**все повторити**  
sort(array)  
виведення масиву

**кінець**

*крок 6*

**початок**

strings = 5  
columns = 7matrix [strings, columns]  
initMatrix(matrix)  
outputMatrix(matrix)  
array[columns]  
**для** і **від** 1 **до** columns

array[i] = averageColumn(matrix, i);

**все повторити**  
sort(array)  
**для** і **від** 1 **до** columns

**виведення** array[i]

**все повторити**

**кінець**

**Псевдокод підпрограм:**

**initMatrix(**matrix**)**

**для** і **від** 1 **до** strings **повторити**

**для** j **від** 1 **до** columns **повторити**

matrix[i, j] = rand(-9, 9)

**все повторити**

**все повторити**

**outputMatrix(**matrix)

**для** і **від** 1 **до** strings **повторити**

**для** j **від** 1 **до** columns **повторити**

**виведення** matrix[i, j]

**все повторити**

**виведення ‘\n’**

**все повторити**

**averageColumn(**matrix, numOfColumn**)**

k = 0  
 sum = 0  
 **для** і **від** 1 **до** strings

**якщо** matrix[i, numOfColumn] >= 0

sum = sum + matrix[i, numOfColumn]

k = k + 1

**все якщо**

**все повторити  
якщо** k == 0

**повернути 0**

average = sum / k  
**повернути** average

**sort(**array**)**

**для i від 1 до** columns - 1 **повторити**

**для j від 1 до** columns - 1 **повторити**

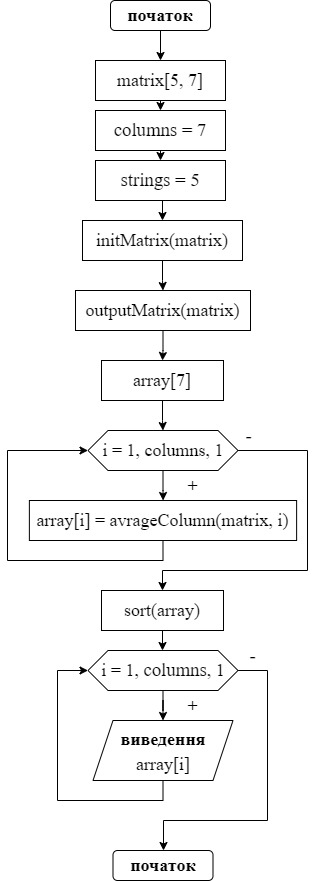
**якщо** array[j] < array[j+1]

tmp = array[j + 1]  
 array[j + 1] = array[j]  
 array[j] = tmp

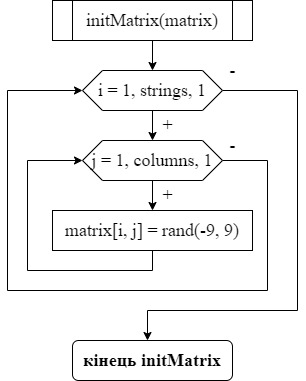
**все якщо**

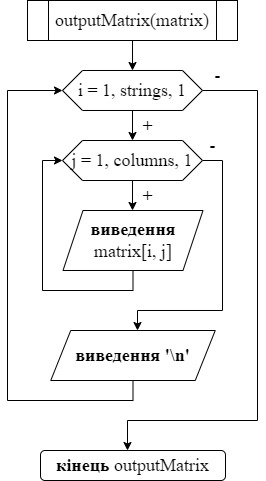
**все повторити**

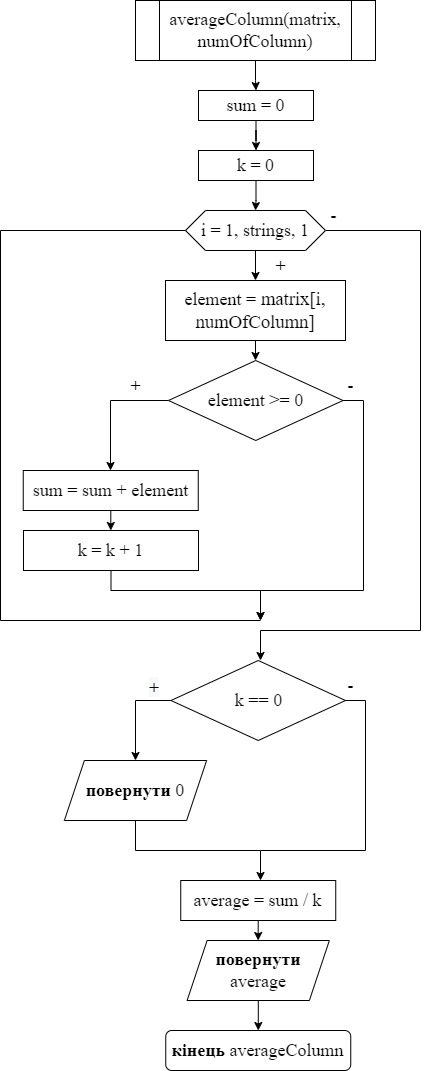
**Блок-схема**

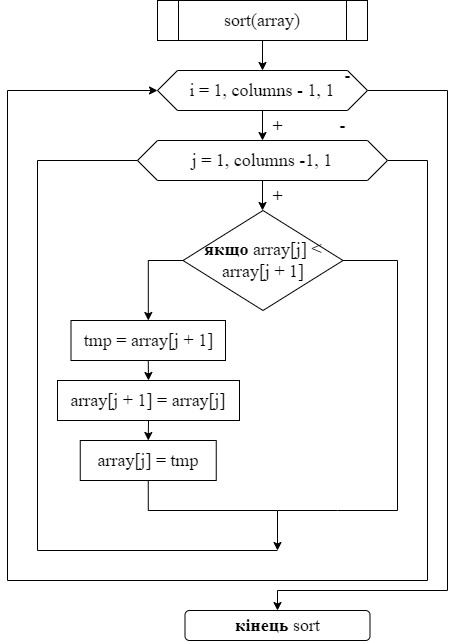


*Блок схеми підпрограм:*

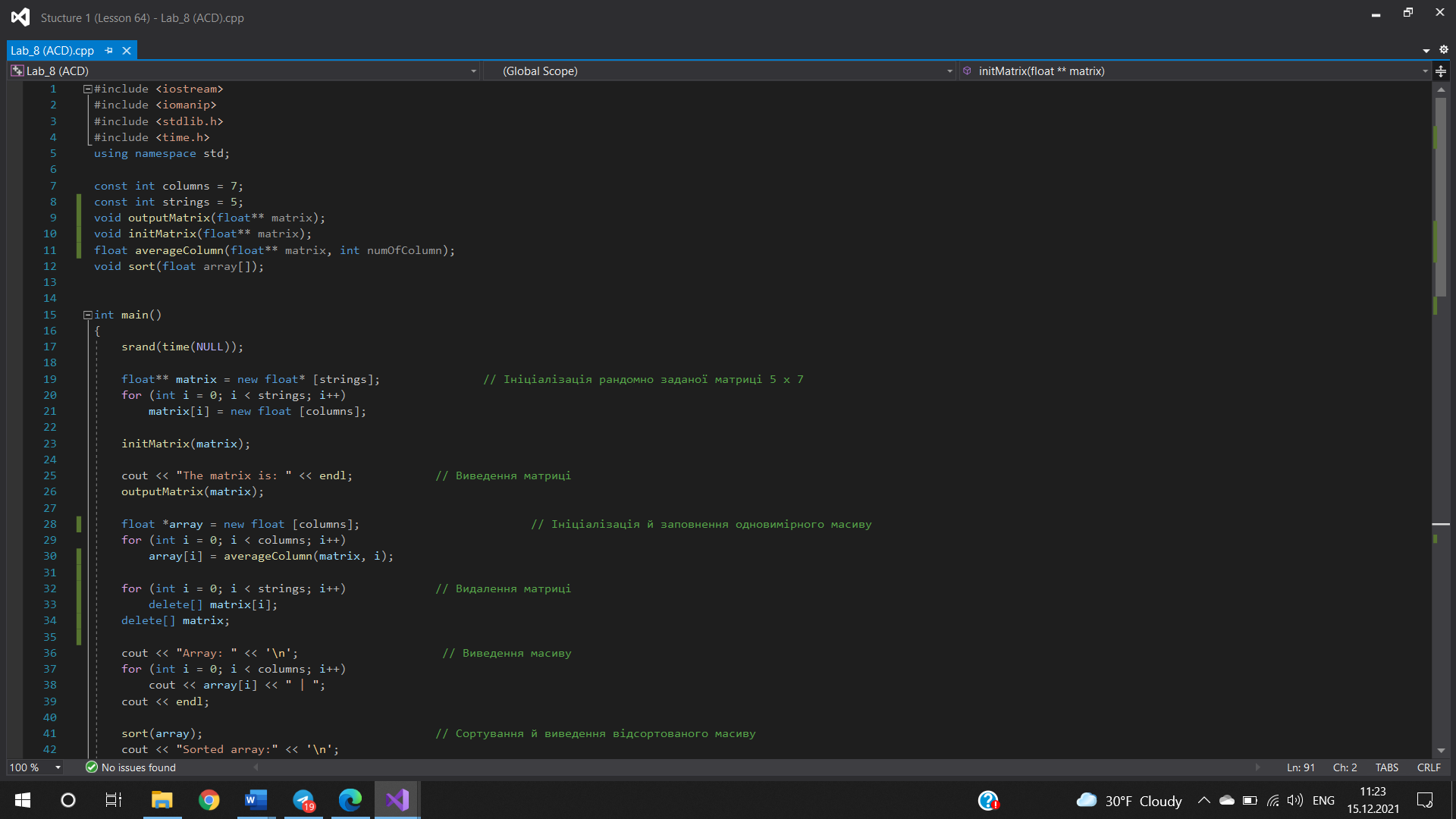
**

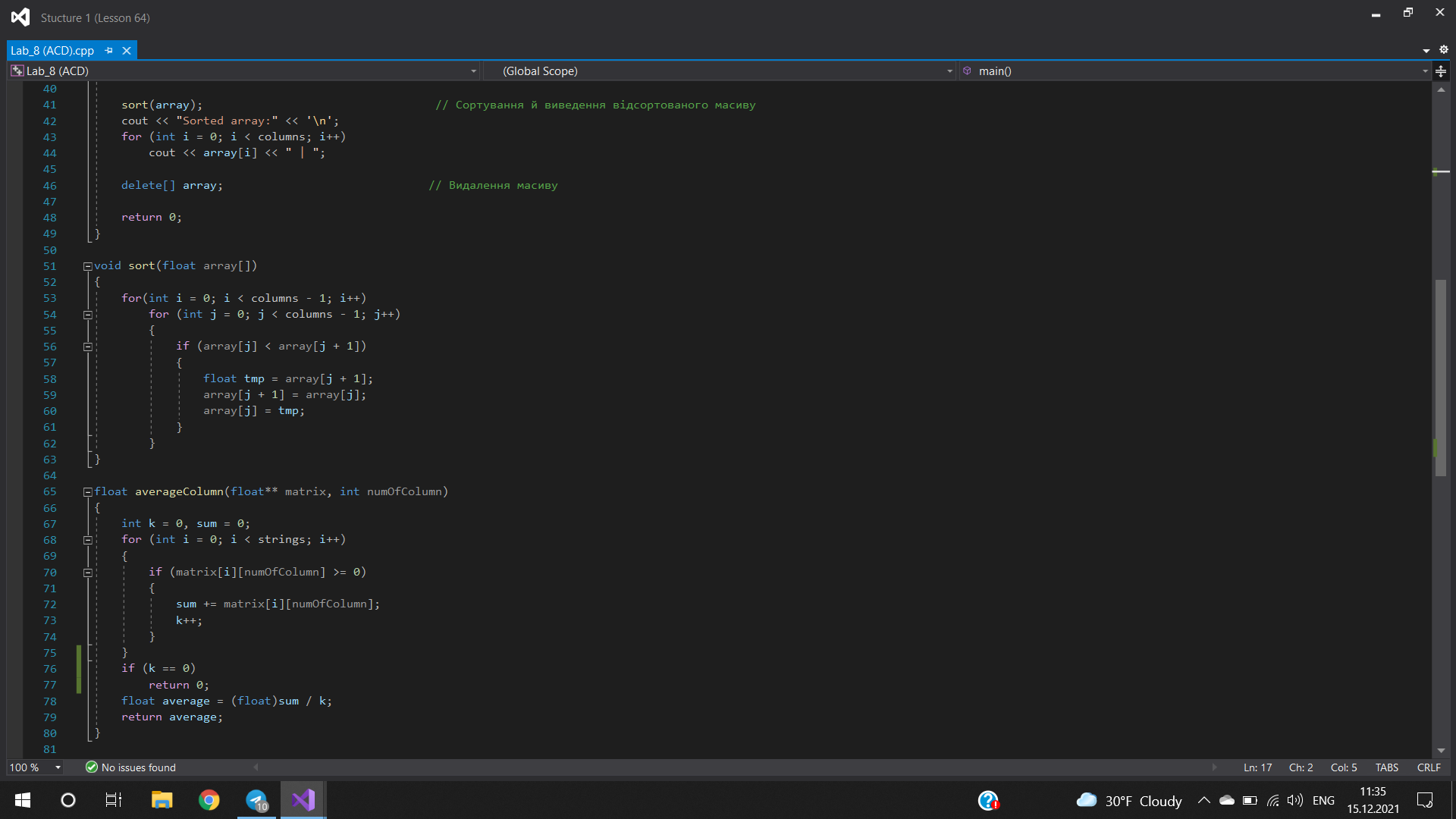


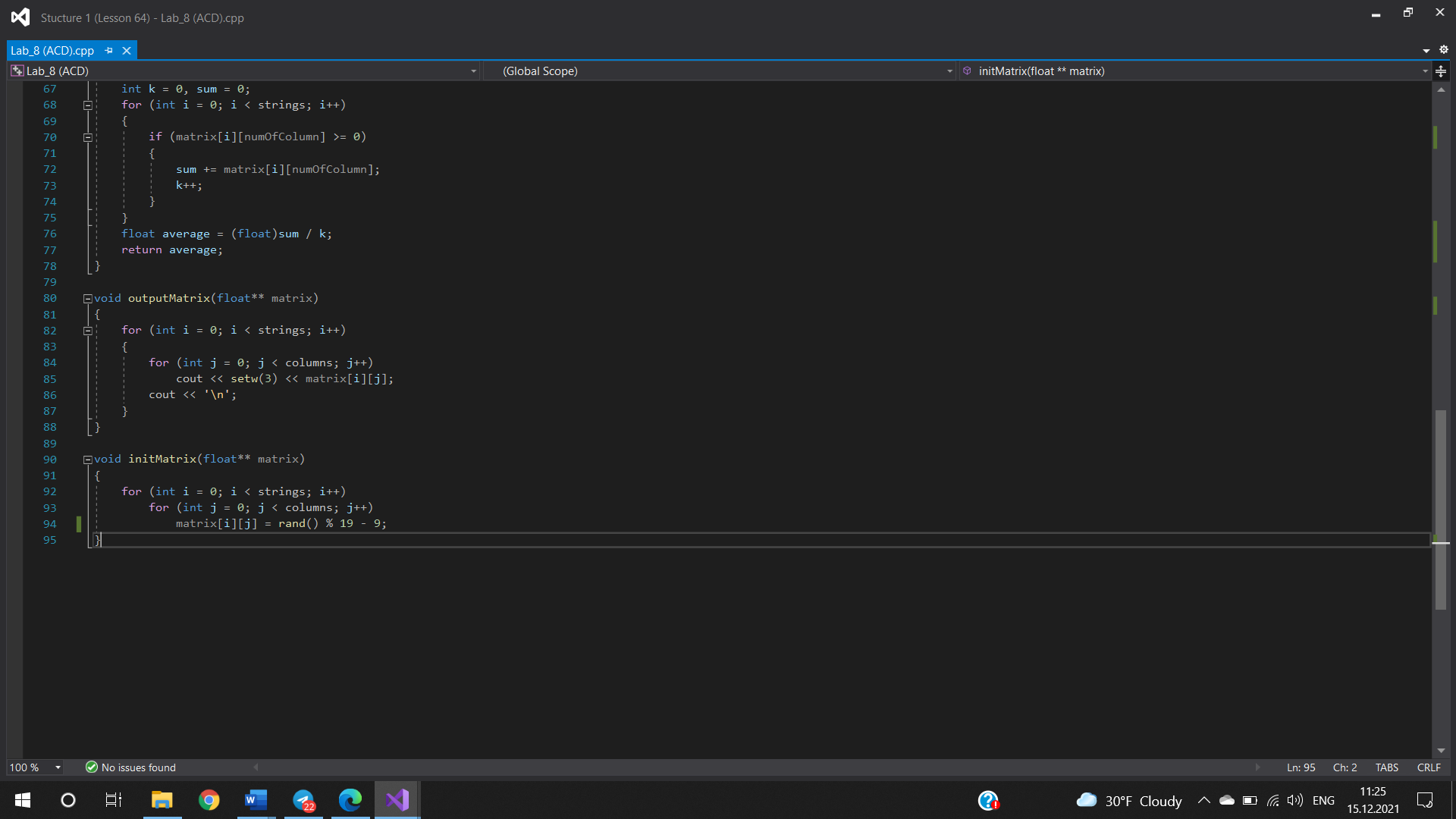




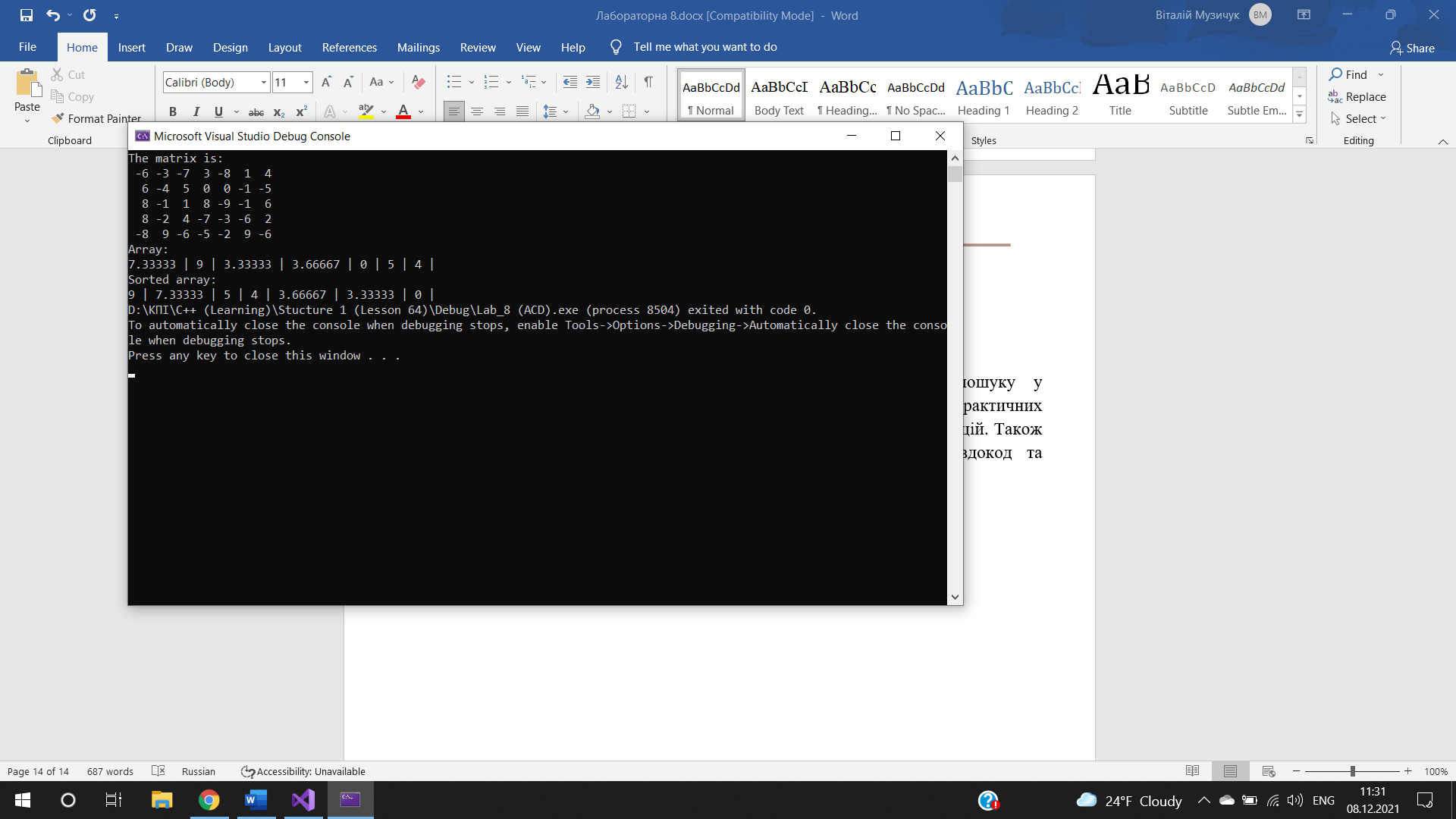
Код програми:

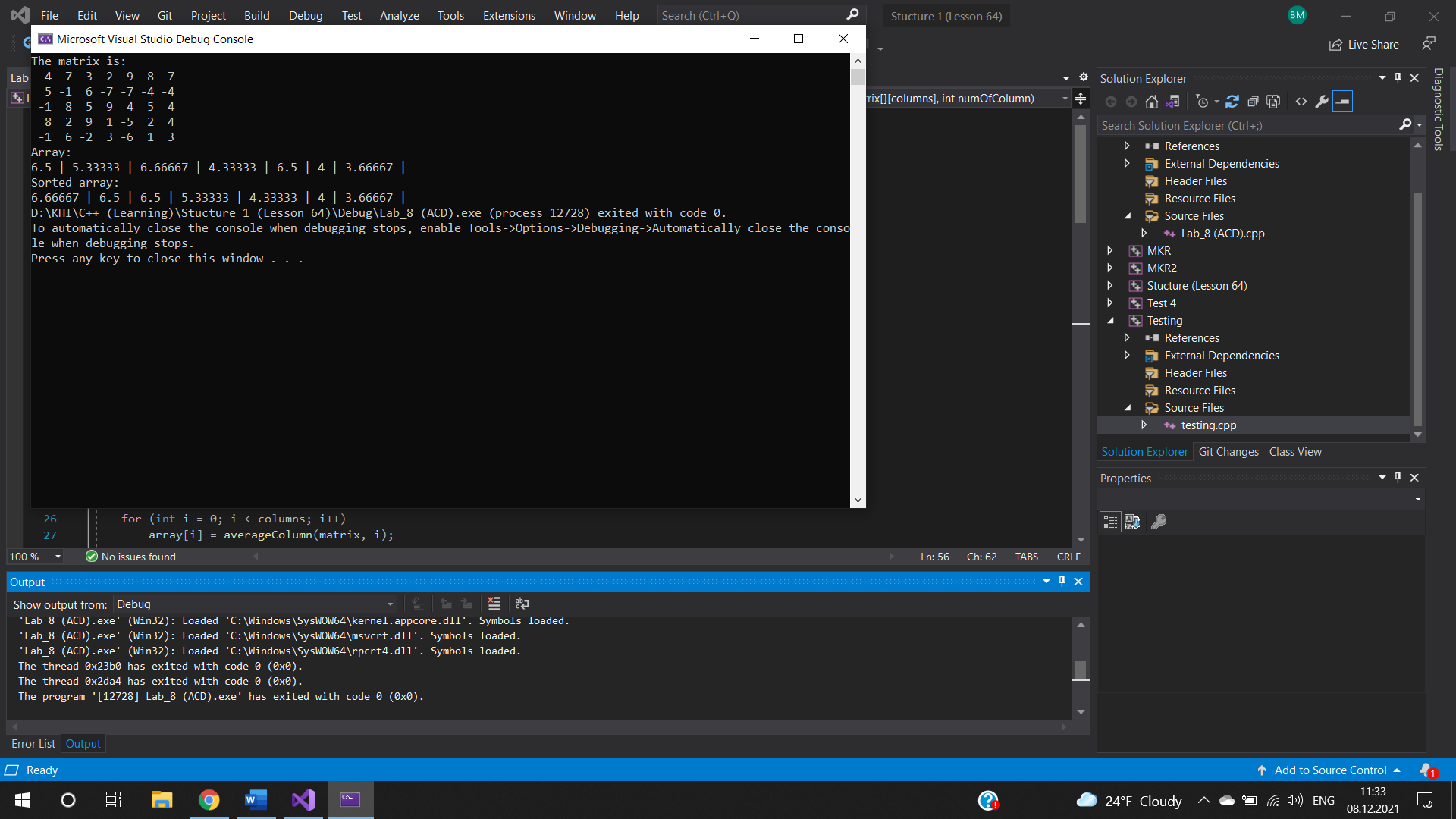






**Випробування алгоритму:**





**Висновки:**

На цій практичній ми дослідити алгоритми пошуку та сортування, набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій. Також зробили постановку задачі склали матмодель написали псевдокод та намлювали блок-схему. Отримали очікуваний результат.