**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-14 Нікулін Павло Юрійович

(шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова Оксана Петрівна

(прізвище, ім'я, по батькові)

**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.

**Завдання**

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

1. Опису змінної індексованого типу (двовимірний масив) 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

**Знайти**: Створити дійсний двомірний масив *5 х 7*. Обчислення значень елементів одновимірного масиву із середнього арифметичного додатних значень елементів стовпців двовимірного масиву. Відсортувати обміном за спаданням.

*Розв’язання*

1. **Постановка задачі**: заповнимо матрицю *matrix* випадковими дійсними числами, створимо масив *average* та запонимо його середніми арифметичними додатніх значень стовпців *matrix*. Відсортуємо матрицю методом обміну («бульбашки»). Виведемо матрицю.
2. Побудова **математичної моделі**. Складемо таблицю імен змінних.

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Кількість рядків | Ціле | N | Початкове дане |
| Кількість стовпціві | Ціле | M | Початкове дане |
| Матриця для сортування | Дійсний | MATRIX | Початкове/Результат |
| Масив середніх арифметичних | Дійсний | AVERAGE | Проміжне дане |
| Задання матриці | Універсальний | INPUT | Функція |
| Середні арифметичні стовпців | Універсальний | AVERAGE\_COL | Функція |
| Сортування матриці | Універсальний | SORT | Функція |
| Виведення матриці | Універсальний | OUTPUT | Функція |
| Формальні масиви | Дійсний | ARR/AVE | Проміжне дане |
| Тимчасове значення | Дійсний | TEMP | Проміжне дане |
| Лічильник | Ціле | COUNT | Проміжне дане |
| Формальні змінні циклів | Ціле | K, I, J | Проміжне дане |

*Крок 1.* Визначимо основні дії.

*Крок 2.* Заповнити матрицю.

*Крок 3.* Знайти середні арифметичні.

*Крок 4.* Відсортувати матрицю.

*Крок 5.* Вивести матрицю.

.

*Псевдокод*

*крок 1*

**початок**

Заповнити матрицю

Знайти середні арифметичні

Відсортувати матрицю

Вивести матрицю

**кінець**

*крок 2*

**початок**

**поки** i < n **повторити**

matrix[i][j] = (rand() % 200 - 100) / 10

Вивести matrix[i][j]

i++

**все повторити**

Знайти середні арифметичні

Відсортувати матрицю

Вивести матрицю

**кінець**

*крок 3*

**початок**

**поки** i < n **повторити**

matrix[i][j] = (rand() % 200 - 100) / 10

Вивести matrix[i][j]

i++

**все повторити**

**поки** j < m **повторити**

average[j] = 0

**поки** i < n **повторити**

**якщо** arr[i][j] > 0 **то**

ave[j] += arr[i][j]

count++

**все якщо**

i++

**все повторити**

j++

**все повторити**

Відсортувати матрицю

Вивести матрицю

**кінець**

*крок 4*

**початок**

**поки** i < n **повторити**

matrix[i][j] = (rand() % 200 - 100) / 10

Вивести matrix[i][j]

i++

**все повторити**

**поки** j < m **повторити**

average[j] = 0

**поки** i < n **повторити**

**якщо** matrix [i][j] > 0 **то**

average [j] += matrix [i][j]

count++

**все якщо**

i++

**все повторити**

average [j] /= count

j++

**все повторити**

**поки** k = 1; k < m **повторити**

**поки** j < m - k **повторити**

**якщо** average [j + 1] > average [j] **то**

**поки** i < n **повторити**

temp = matrix [i][j + 1]

matrix [i][j + 1] = matrix [i][j]

matrix [i][j] = temp

i++

**все повторити**

temp = average [j + 1]

average [j + 1] = average [j]

average [j] = temp

**все якщо**

j++

**все повторити**

k++

**все повторити**

Вивести матрицю

**кінець**

*крок 5*

**початок**

**поки** i < n **повторити**

matrix[i][j] = (rand() % 200 - 100) / 10

Вивести matrix[i][j]

i++

**все повторити**

**поки** j < m **повторити**

average[j] = 0

**поки** i < n **повторити**

**якщо** matrix [i][j] > 0 **то**

average [j] += matrix [i][j]

count++

**все якщо**

i++

**все повторити**

average [j] /= count

j++

**все повторити**

**поки** k = 1; k < m **повторити**

**поки** j < m - k **повторити**

**якщо** average [j + 1] > average [j] **то**

**поки** i < n **повторити**

temp = matrix [i][j + 1]

matrix [i][j + 1] = matrix [i][j]

matrix [i][j] = temp

i++

**все повторити**

temp = average [j + 1]

average [j + 1] = average [j]

average [j] = temp

**все якщо**

j++

**все повторити**

k++

**все повторити**

**поки повторити**

**поки повторити**

Вивести matrix[i][j]

j++

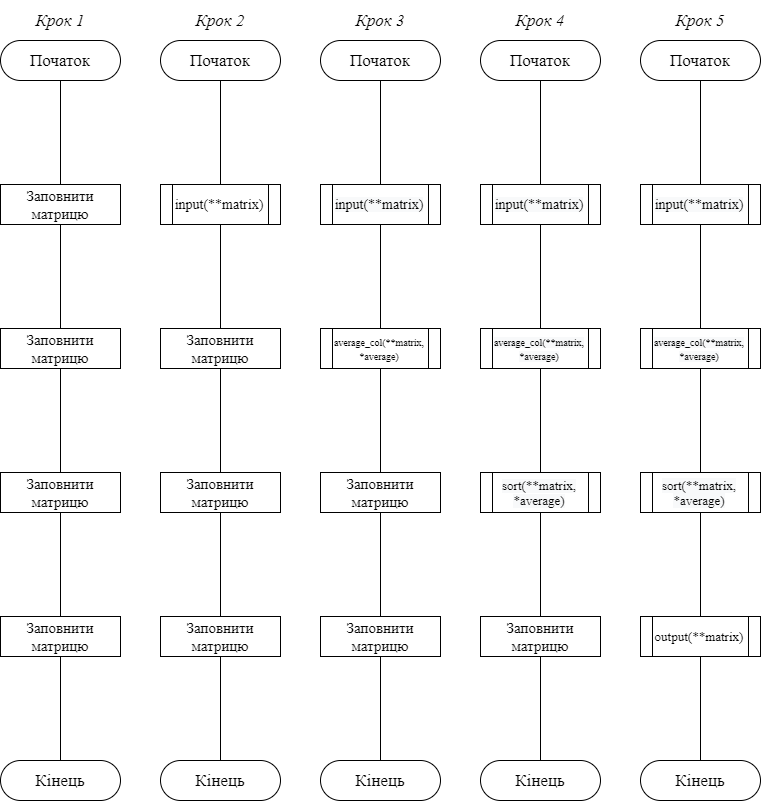
**все повторити**

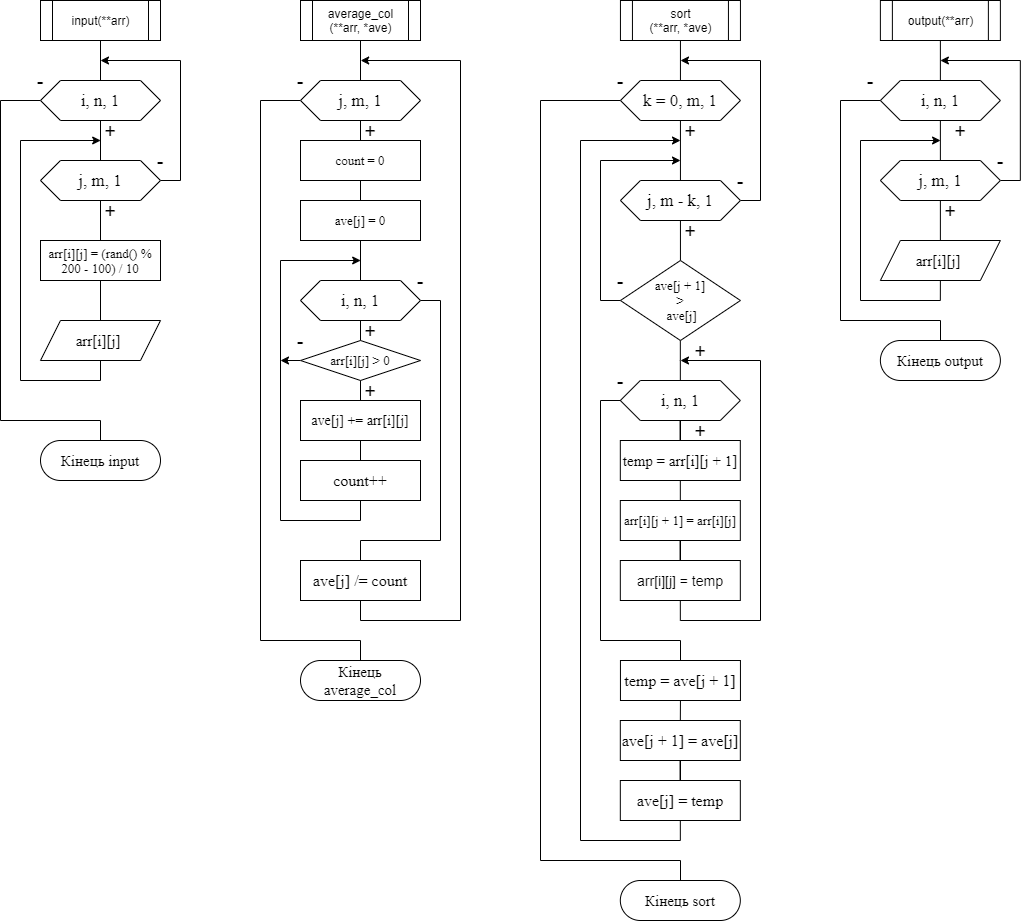
i++

**все повторити**

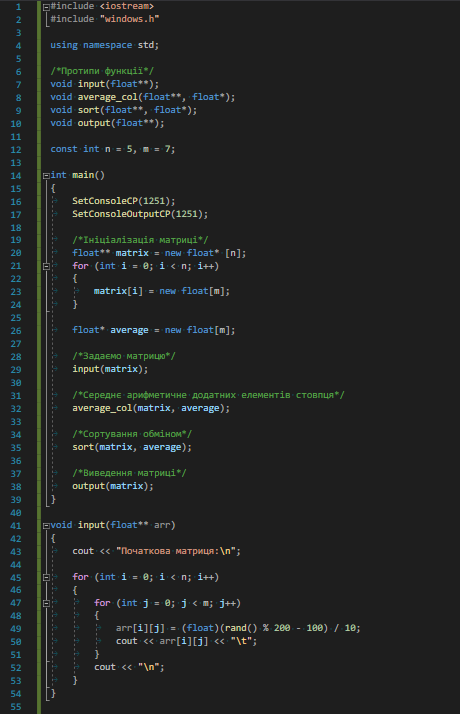
**кінець**

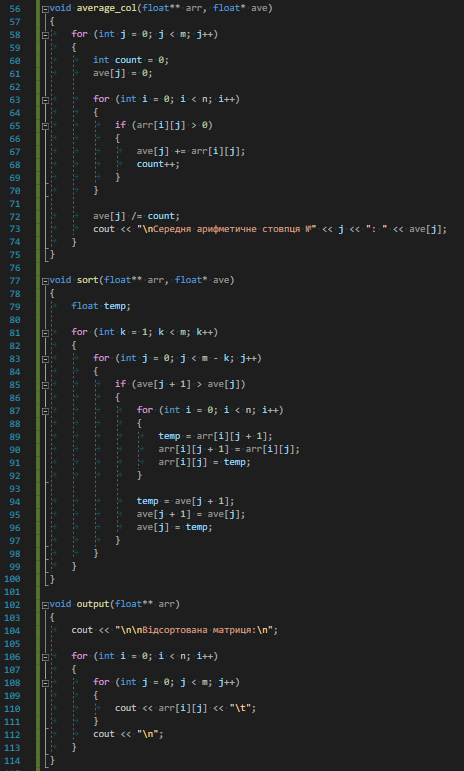
*Блок-схема*



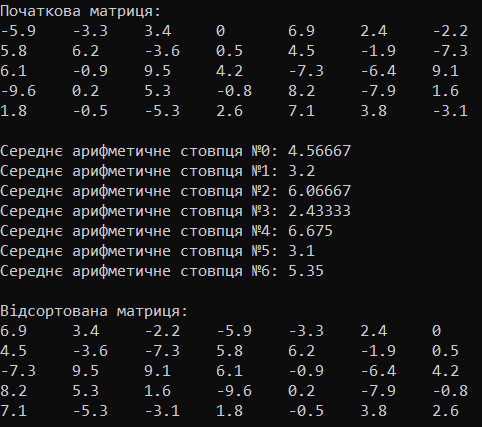
****

**Код програми**





*Тестування програми*



**Висновок**

Під час виконання лабораторної роботи було досліджено алгоритми пошуку та сортування, набуто практичних навичок використання цих алгоритмів під час складання програмних специфікацій. У роботі використано 5 функцій: *main, input, average\_col, sort* та *output*. У функції *input* ми задаємо випадкові дійсні значення матриці, у *average\_col* ми шукаємо середнє арифметичне додатніх значень для кожного стовпця. У функції *sort* ми сортуємо матрицю алгоритмом обміну («бульбашки»), а саме розташовуємо стовпці у порядку спадання їх середніх арифметичних. У функції *output* ми виводимо відсортовану матрицю на екран. Було використано 6 звичаних і 3 вкладених цикли *for*, де у ролі інкремент виступали *i, j* та *k*. Код написаний на мові програмування C++. Результат роботи є вірним, виконано згідно з алгоритмом.