

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 7

Виконав студент ІП-13, Гогіберідзе Торніке Лашаєвич

(шифр, прізвище, ім'я, по батькові)

Перевірила Вєчерковська Анастасія Сергіївна

(прізвище, ім'я, по батькові)

Київ 2021  
**Лабораторна робота 8**

## Дослідження алгоритмів пошуку та сортування

**Мета** – дослідити методи послідовного пошуку у впорядкованих і невпорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій. **Варіант**

7

### Завдання

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

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом (табл. 1).
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).

|   |       |         |  |
|---|-------|---------|--|
| 7 | 8 x 5 | Дійсний | Із добутку значень елементів рядків двовимірного масиву. Відсортувати обміном за зростанням. |
|---|-------|---------|--|

### Постановка задачі

Використовуємо 3 підпрограми для ініціалізації матриці (дволінкового масиву), ініціалізації масива і сортування цього масиву. Матриці заповнюємо випадковими дійсними числами. Масив заповнюємо добутками елементів кожного рядка матриці. Потім відсортувуємо масив обміном за зростанням.

Результатом розв'язку є обчислення матриці, обчислення і сортування масива.

### Побудова математичної моделі

Таблиця імен змінних

| Змінна | Тип | Ім'я | Призначення |
|--------|-----|------|-------------|
|--------|-----|------|-------------|

|                                     |           |                  |                         |
|-------------------------------------|-----------|------------------|-------------------------|
| Матриця                             | float     | matrix           | Проміжні дані           |
| Масив                               | float     | array            | Проміжні дані/Результат |
| Кількість рядків матриці            | int       | row              | Початкові дані          |
| Кількість стовпців матриці          | int       | col              | Початкові дані          |
| Процедура для ініціалізації матриці | Процедура | initializeMatrix | Початкові дані          |
| Лічильник i                         | int       | i                | Початкові дані          |
| Лічильник j                         | int       | j                | Початкові дані          |
| Процедура для ініціалізації масиву  | Процедура | initializeArray  | Початкові дані          |
| Добуток елементів рядків матриці    | float     | product          | Проміжні дані           |

|   |                     |  |                |
|---|---------------------|--|----------------|
| Процедура для сортування масиву обміном за зростанням | Процедура SortArray |  | Початкові дані |
| Змінна, яка тимчасово тримає значення елемента        | float temp          |  | Проміжні дані  |

Спочатку встановіть 8 і 5 як значення змінних row і col відповідно. Після оголошення матриці ми ініціалізуємо її за допомогою процедури initializeMatrix, яка приймає в якості параметрів матрицю, кількість рядків і кількість стовпців (матриця, рядок, стовпець). Ця процедура використовує функцію randomFloat (0.10) для заповнення матриці випадковими дійсними числами від 0 до 10.

Потім скористайтеся процедурою initializeArray для ініціалізації масиву масиву, який приймає в якості параметрів масив, матрицю, кількість рядків і кількість стовпців (масив, матриця, рядок, стовпець). Використовуючи два арифметичні цикли (зовнішній і вкладений) і лічильники sum і j, приймаючи значення від 0 до row і col відповідно, програма обчислює добуток кожного елемента рядка матриці і записує його у відповідний масив елементів. ми використовуємо процедуру SortArray, яка приймає масив і кількість рядків як параметри (масив, рядок). Цей процес реалізує сортування за зростанням. Використовуйте два арифметичні цикли

(Зовнішній і вкладений) із сумою лічильника та j, значення від 0 до першого рядка, кожне значення масиву обходиться, якщо воно більше за наступне (масив [j + 1] < масив [j]), вони оновлені Композиція (для аранжування використовується змінна температура).

Вихідний масив SortArray після процесу.

## *Розв'язання*

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

*Крок 1.* Визначимо основні дії.

*Крок 2.* Ініціалізація змінних row і col

*Крок 3.* Ініціалізація матриці

*Крок 4.* Ініціалізація масиву

*Крок 5.* Сортування та виведення масиву

## *Псевдокод*

### Основна програма:

*Крок 1*

#### **початок**

Ініціалізація змінних row і col

Ініціалізація матриці

Ініціалізація масиву

Сортування та виведення масиву

#### **кінець**

*Крок 2 початок*

row := 8

col := 5

Ініціалізація матриці

Ініціалізація масиву

Сортування та виведення масиву  
**кінець**

*Крок 3*

**початок**

row := 8

col := 5

initializeMatrix (matrix)

Ініціалізація масиву   Сортування та  
виведення масиву **кінець**

*Крок 4 початок*

row := 8

col := 5

initializeMatrix (matrix)

initializeArray (array)   Сортування та  
виведення масиву **кінець**

*Крок 5 початок*

row := 8

col := 5

initializeMatrix (matrix)

initializeArray (array)

SortArray(array)   **виведення**  
array **кінець**

Підпрограми:

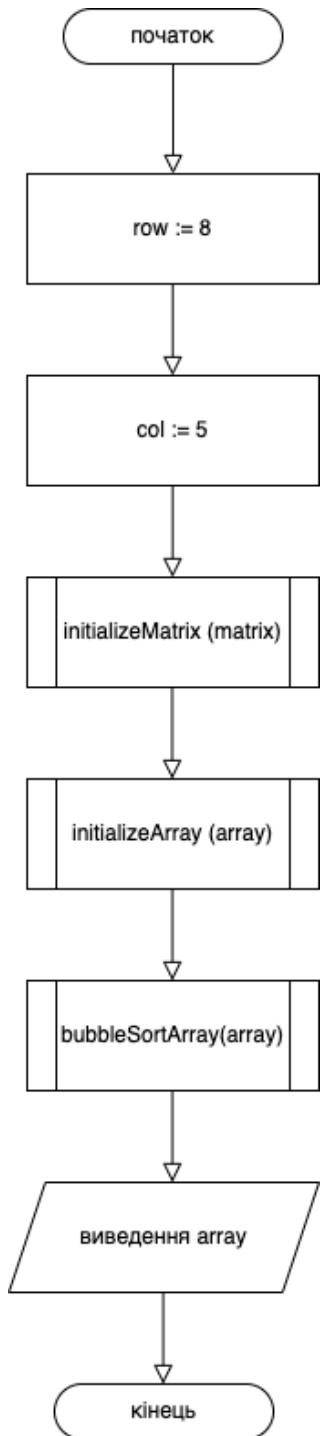
```
initializeMatrix (matrix, row, col)
    повторити для i від 1 до row
        повторити для j від 1 до col
            matrix[i][j] = randomFloat(0,10)
        все повторити
    все повторити
кінець
```

```
initializeArray (array, matrix, row, col)
    повторити для i від 1 до row
        product := 1
        повторити для j від 1 до col
            product := product * matrix[i][j]
        все повторити
        array[i] = product
    все повторити
кінець
```

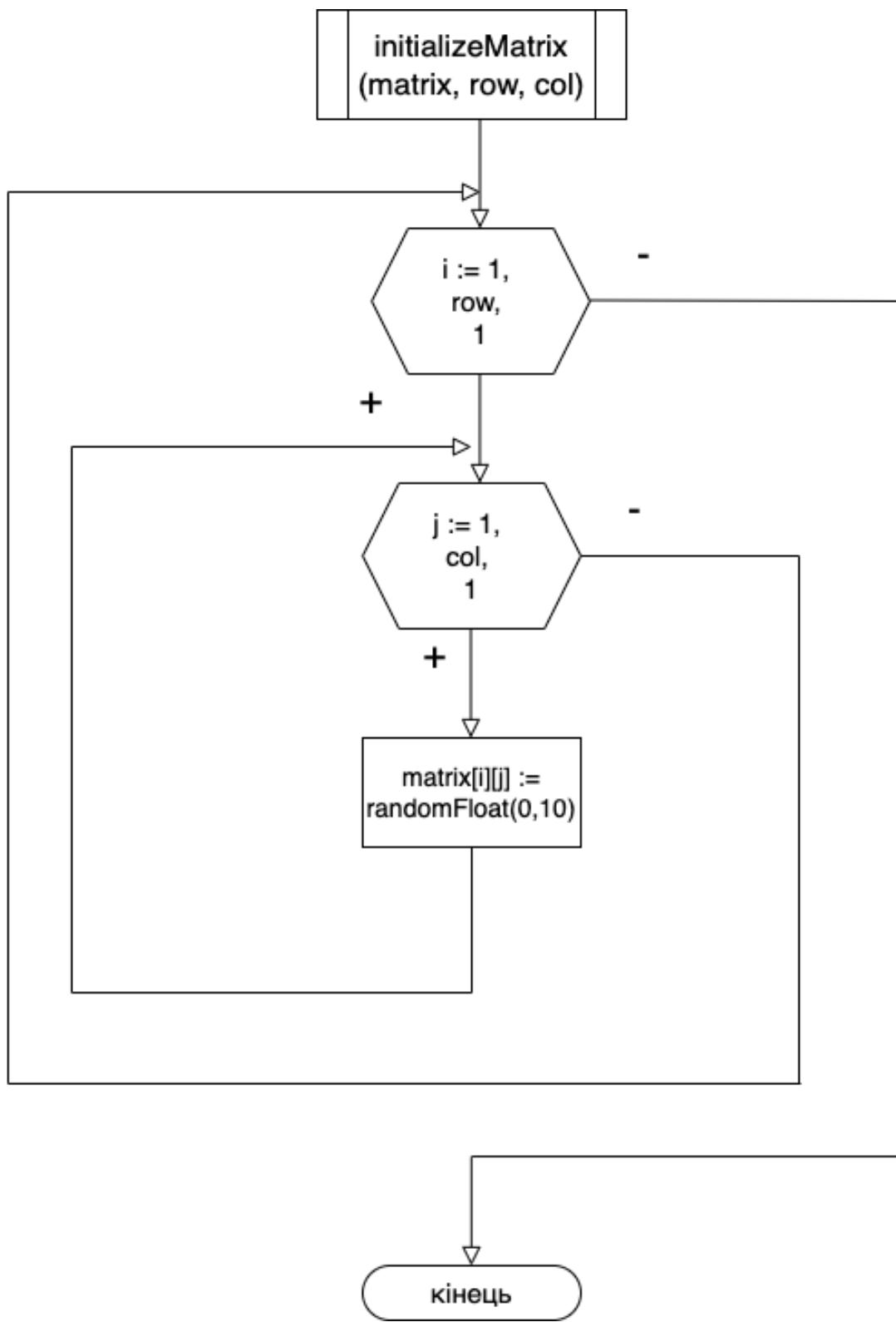
```
SortArray (array, row)    повторити для i від
    1 до (row - 1)          повторити для j від
    1 до (row - 1)          якщо array[j +
    1] < array[j]           то
        temp := array[j + 1]      array[j +
        1] := array[j]           array[j] := temp
    все якщо
    все повторити все
    повторити
кінець
```

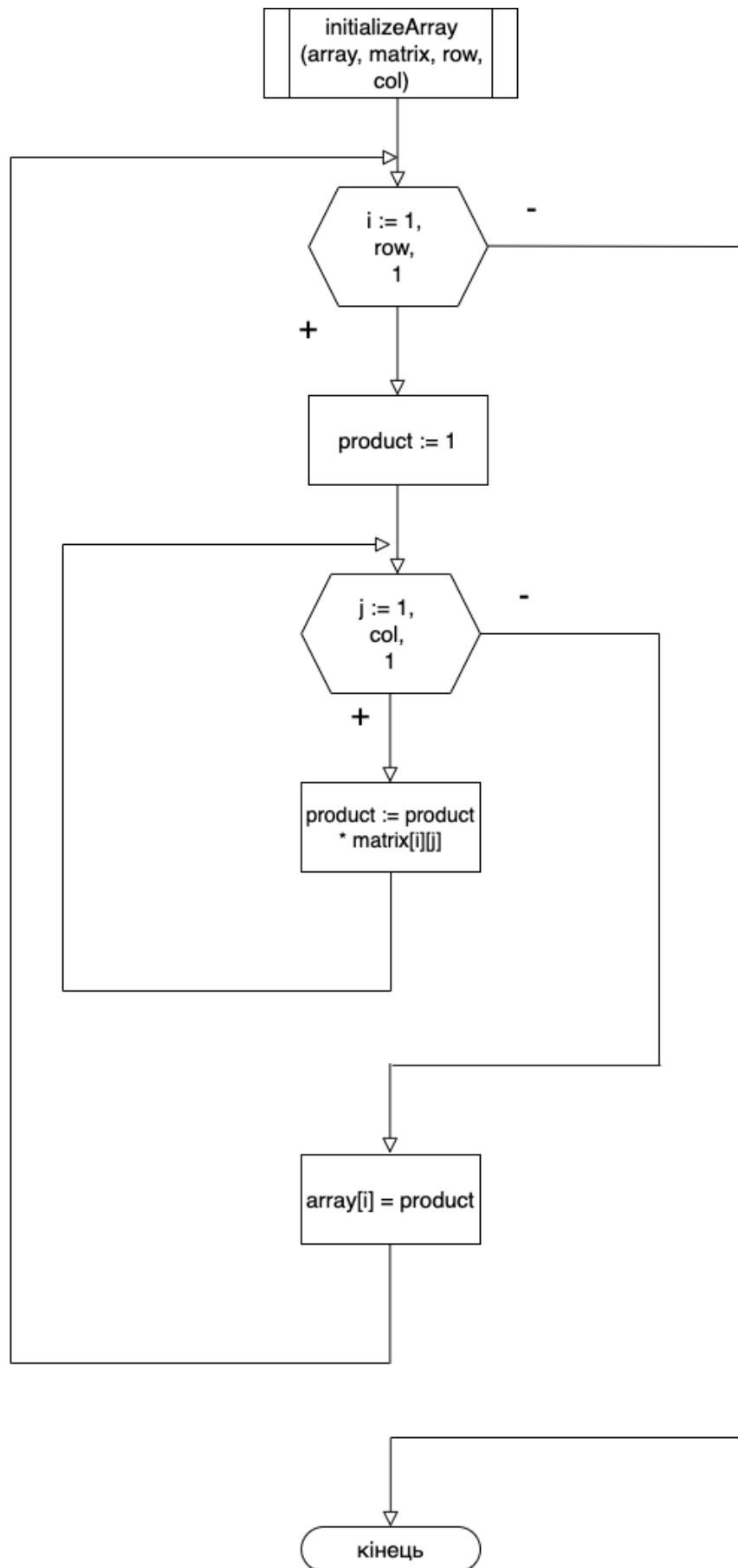
*Блок-схема:*

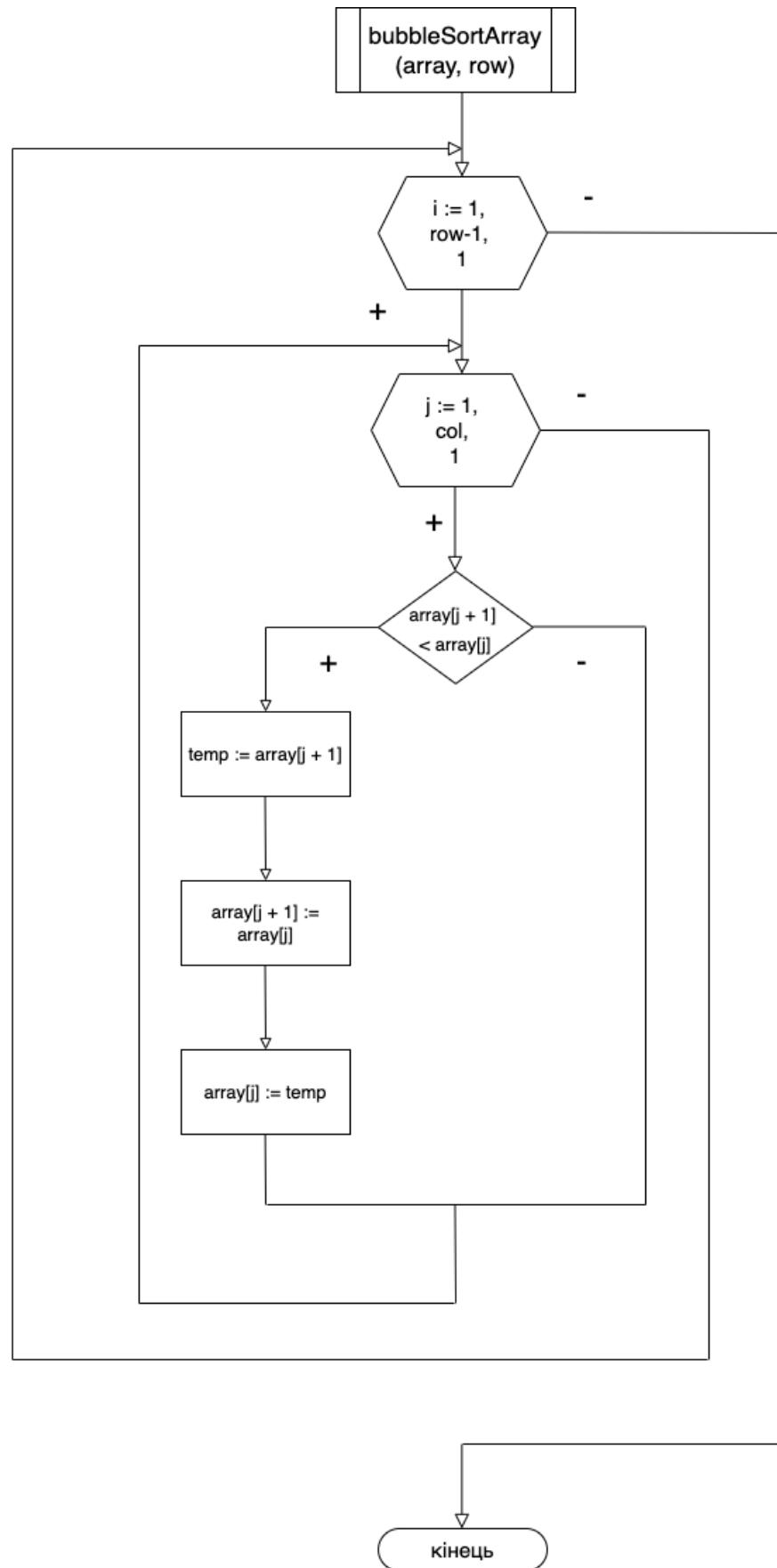
*Основна програма:*



Підпрограми:







```
1 #include <iostream>
2 #include <iomanip>
3 #include <time.h>
4
5 using namespace std;
6
7 void initializeMatrix(float**, int, int);
8 void initializeArray(float* array, float** matrix, int row, int col);
9 void SortArray (float* array, int length);
10
11 int main()
12 {
13     srand((float)time(NULL));
14
15     int row = 8, col = 5;
16
17     float** matrix = new float* [row];
18     for (int i = 0; i < row; i++) {
19         matrix[i] = new float[col];
20     }
21
22     initializeMatrix(matrix, row, col);
23
24     float* array = new float[row];
25     initializeArray(array, matrix, row, col);
26
27
28     cout << "Matrix" << endl;
29     for (int i = 0; i < row; i++) {
30         for (int j = 0; j < col; j++) {
31             printf("%8.3f", matrix[i][j]);
32         }
33         cout << endl;
34     }
35     cout << endl;
36
37
38     cout << "Unsorted Array" << endl;
39     for (int i = 0; i < row; i++) {
40         printf("%-11.3f", array[i]);
41     }
42     cout << endl;
43
44     SortArray(array, row);
45
46     cout << "\nSorted Array" << endl;
47     for (int i = 0; i < row; i++) {
48         printf("%-11.3f", array[i]);
49     }
50     cout << "\n\n";
```

```
52     delete[] array;
53
54     for (int i = 0; i < row; i++) {
55         delete[] matrix[i];
56     }
57     delete[] matrix;
58
59     return 0;
60 }
61
62 void initializeMatrix(float** matrix, int row, int col) {
63
64     for (int i = 0; i < row; i++) {
65         for (int j = 0; j < col; j++) {
66             matrix[i][j] = (float)(rand()) / (float)RAND_MAX * 10;
67         }
68     }
69 }
70
71 void initializeArray(float* array, float** matrix, int row, int col) {
72
73     float product;
74
75     for (int i = 0; i < row; i++) {
76         product = 1;
77
78         for (int j = 0; j < col; j++) {
79             product *= matrix[i][j];
80         }
81         array[i] = product;
82     }
83 }
84
85 void SortArray(float* array, int row) {
86
87     float temp;
88
89     for (int i = 0; i < row - 1; i++) {
90         for (int j = 0; j < row - 1; j++) {
91
92             if (array[j + 1] < array[j]) {
93
94                 temp = array[j + 1];
95                 array[j + 1] = array[j];
96                 array[j] = temp;
97             }
98         }
99     }
100 }
101 }
```

```
Microsoft Visual Studio Debug Console
Matrix
 9.960  7.071  9.276  9.661  9.675
 4.638  4.205  8.731  0.680  7.622
 5.314  1.017  9.580  5.849  9.232
 3.336  0.815  6.334  8.896  9.907
 7.635  6.620  4.811  4.830  5.901
 9.611  1.787  8.108  7.292  0.220
 0.242  8.008  3.454  9.156  6.104
 7.590  7.682  4.140  8.399  8.195

Unsorted Array
61068.383  882.384    2794.870   1518.119   6929.165   223.434    373.708   16618.150

Sorted Array
223.434    373.708   882.384    1518.119   2794.870   6929.165   16618.150   61068.383

C:\Users\user\source\repos\ASD_Lab8\Debug\ASD_Lab8.exe (process 9876) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

## Висновки

Під час роботи у восьмій лабораторії я вивчав методи послідовного пошуку в упорядкованих і невпорядкованих послідовностях та отримав практичні навички їх використання при написанні специфікацій програмного забезпечення. В результаті я отримав алгоритм, який ініціалізує матрицю з випадковими дійсними числами, а в цьому випадку ініціалізує масив і сортує масив.