

*Додаток 1*

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 №1 з дисципліни  
«Алгоритми та структури даних-1.  
Основи алгоритмізації»

**Дослідження алгоритмів пошуку та сортування**

Варіант 18

Виконав студент ІП-13 Король Валентин Олександрович  
(шифр, прізвище, ім'я, по батькові)

Перевірив Всечерковська Анастасія Сергіївна  
(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.

### Індивідуальне завдання

#### Варіант 18.

#### Постановка задачі:

18	5 x 5	Дійсний	Із від'ємних значень елементів побічної діагоналі двовимірного масиву. Відсортувати методом вставки за спаданням.
----	-------	---------	---

#### Побудова математичної моделі:

Задача зводиться до необхідного сортування масиву, сформованого з елементів побічної діагоналі заданої матриці. Принцип сортування вставками за спаданням полягає в тому, що ми вибираємо один з елементів вхідних даних і вставляємо його на потрібну позицію у вже відсортованому списку доти, доки набір вхідних даних не буде вичерпано.. Так як матриця квадратна, індекси елементів побічної діагоналі двовимірного масиву будуть рівні, тому сформувати масив зі значень головної діагоналі можна сформувати за допомогою одного циклу.

Змінна	Тип	Ім'я	Призначення
Матриця	дійсний	Matr	Вхідні дані
Функція генерації матриці	дійсний	RAND_MAX	Проміжні дані
Одновимірний масив	дійсний	Vect	Проміжні дані

#### Розв'язання:

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Заповнимо матрицю згенерованими числами

Крок 3. Деталізуємо вивід одновимірного масиву

Крок 4. Відсортуємо масив

## **Псевдокод:**

### **Крок 1**

#### **початок**

Ввід n

Matr = float (RAND\_MAX)/-(float(rand()))

array = **createArr**(matrix, n)

Vect[i] = Matr[i][4 - i]

Sort(result, 5)

**кінець**

## Крок 2

### Початок

повторити для  $i$  від 0 до 5:

    повторити для  $j$  від 0 до 5:

        Вивід     $\text{Matr}[i][j]$                     **все**

повторити    **все**    повторити

$\text{Vect}[i] = \text{Matr}[i][4 - i]$

        Sort(result, 5)

кінець

## Крок 3

### Початок

повторити для  $i$  від 0 до 5:

    повторити для  $j$  від 0 до 5:

        Вивід     $\text{Matr}[i][j]$                     **все**

повторити    **все**    повторити

        повторити для  $i$  від 0 до 5:

$\text{Vect}[i] = \text{Matr}[i][4 - i]$

Sort(result, 5)

Кінець

## Крок 4

### Початок

повторити для  $i$  від 0 до 5:

**повторити для j від 0 до 5:**

Вивід    Matr[i][j]                    **все**

**повторити   все повторити**

**повторити для i від 0 до 5:**

        Vect[i] = Matr[i][4 - i]

**повторити для i від 0 до 5:**

**key = Vect[i]**

**Поки**            j >= 0 та Vect[j] < key

**повторити для j від i - 1**

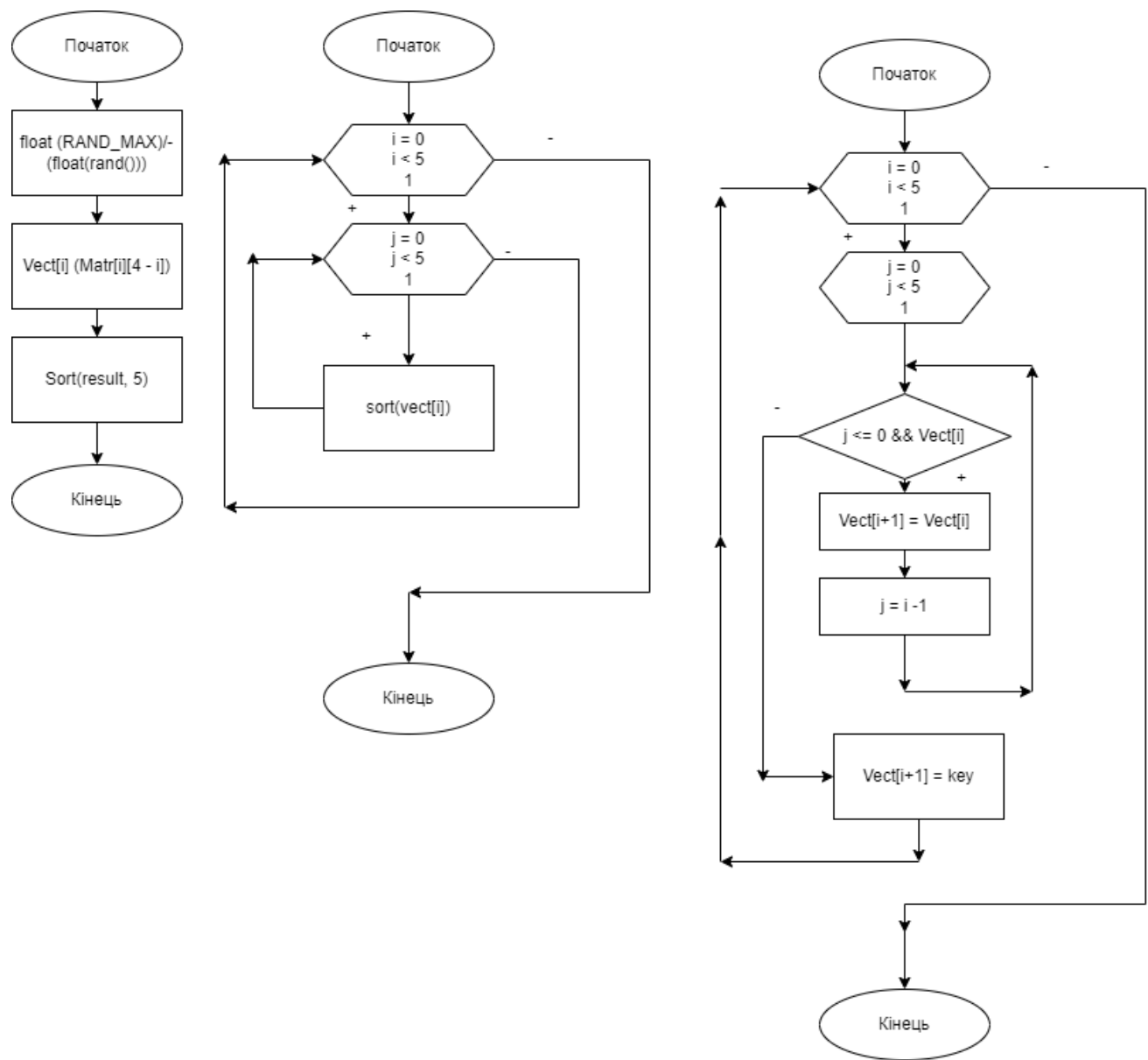
            Vect[j + 1] = Vect[j]

            Vect[j + 1] = key

**Все повторити**

**Кінець**

## Блок-схема:



## Код алгоритму:

Скріншоти коду на C++:

```
(Global Scope)
#include <iostream>
#include <iomanip>
#include <ctime>

using namespace std;

int main()
{
    double Matr[5][5];
    double Vect[5];
    srand((float)time(NULL));

    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            Matr[i][j] = float (RAND_MAX)/(float(rand()));
            cout << Matr[i][j] << "    ";
        }
        cout << "\n" << endl;
    }
    cout << "\n";

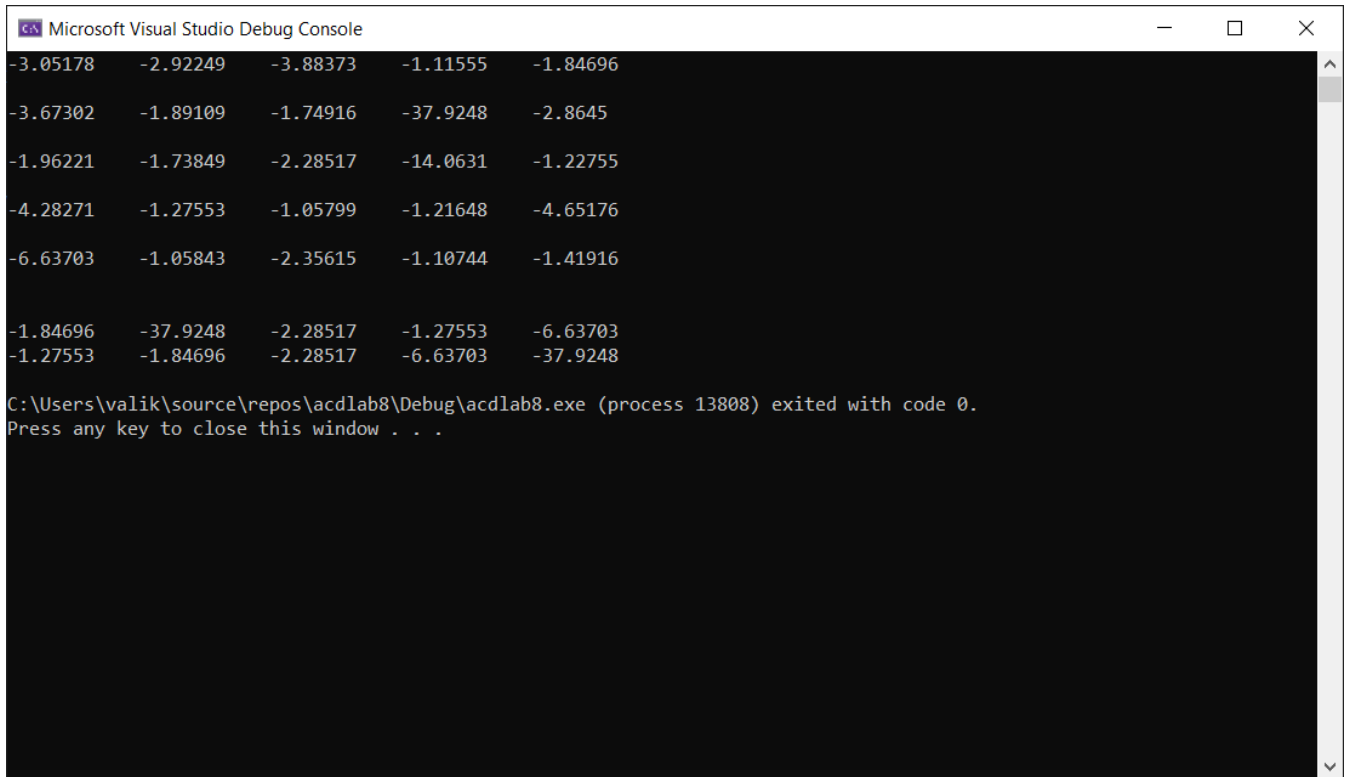
    for (int i = 0; i < 5; i++)
    {
        Vect[i] = Matr[i][4 - i];
        cout << Vect[i] << "    ";
    }
    cout << "\n";
    int j;
    float key; int temp;
    for (int i = 1; i < 5; i++)
    {
        key = Vect[i];
        for (j = i - 1; j >= 0 && Vect[j] < key; j--)
            Vect[j + 1] = Vect[j];
        Vect[j + 1] = key;
    }
}
```

```

        for (j = i - 1; j >= 0 && Vect[j] < key; j--)
            Vect[j + 1] = Vect[j];
        Vect[j + 1] = key;
    }
    for (int i = 0; i < 5; i++)
    {
        cout << Vect[i] << "    ";
    }
    cout << "\n";
}

```

Виконання програми у терміналі:



```

Microsoft Visual Studio Debug Console
-3.05178    -2.92249    -3.88373    -1.11555    -1.84696
-3.67302    -1.89109    -1.74916    -37.9248    -2.8645
-1.96221    -1.73849    -2.28517    -14.0631    -1.22755
-4.28271    -1.27553    -1.05799    -1.21648    -4.65176
-6.63703    -1.05843    -2.35615    -1.10744    -1.41916
-1.84696    -37.9248    -2.28517    -1.27553    -6.63703
-1.27553    -1.84696    -2.28517    -6.63703    -37.9248
C:\Users\valik\source\repos\acdlab8\Debug\acdlab8.exe (process 13808) exited with code 0.
Press any key to close this window . . .

```

Основи програмування – 1. Алгоритми та структури даних

## Висновок:

Я дослідив особливості роботи алгоритмів пошуку та сортування, набув практичних навичок їх використання під час складання програмних специфікацій. Протягом роботи набув навички складання програмних специфікацій пошуку, де було описано математичну модель алгоритму сортування методом вставки, побудовано блок-схему задачі, написано код програми на C++.