Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічнийінститут 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 32

Виконав студент ІП-14, Шляхтун Денис Михайлович
(шифр, прізвище, ім'я, по батькові)

Перевірив доц. кафедри ІПІ Мартинова Оксана Петрівна
(прізвище, ім'я, по батькові)

Київ 2021

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.

Задача:

1. Описати та ініціювати змінну індексованого типу (двовимірний масив) з 5x8 цілих значень.
2. Описати та ініціювати змінну індексованого типу (одновимірний масив) із середнього арифметичного від'ємних значень елементів стовбців двовимірного масиву
3. Відсортувати одновимірний масив методом Шела за спаданням.

Постановка задачі. Результатом розв'язку є відсортований за спаданням одновимірний масив, що складається з середнього арифметичного від'ємних значень елементів стовбців двовимірного масиву. Сортування одновимірного масиву буде відбуватися за методом Шела. Ввідні дані, що вимагаються – двовимірний масив, що буде заповнюватися випадковими значеннями.

Побудова математичної моделі. Складемо таблицю імен змінних

Змінна	***Тип***	***Ім'я***	***Призначення***
Основні змінні			
Двовимірний масив	Цілий	arr1[5][8]	Початкове дане
Одновимірний масив	Цілий	arr2[8]	Результат
Змінні, що використовуються у підпрограмах			
Лічильники	Цілий	i, k	Проміжне значення
Проміжні змінні сортування	Цілий	d, j	Проміжне значення
Кількість елементів масиву	Цілий	n	Проміжне значення

Для виконання алгоритму були використані наступні функції зі стандартних бібліотек:

- srand(time(NULL)) та rand() – генерація випадкових чисел для масиву
- swap() – для зміни місцями значень двох елементів масиву

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію визначення першого масиву.

Крок 3. Деталізуємо дію визначення другого масиву.

Крок 4. Деталізуємо дію сортування другого масиву.

Псевдокод.

Крок 1.

початок

знаходження першого масиву

знаходження другого масиву

сортування другого масиву

кінець

Крок 2.

початок

arr1 = arrRand(arr1)

знаходження другого масиву

сортування другого масиву

кінець

Крок 3.

початок

arr1 = arrRand(arr1)

arr2 = averageNeg(arr1, arr2)

сортування другого масиву

кінець

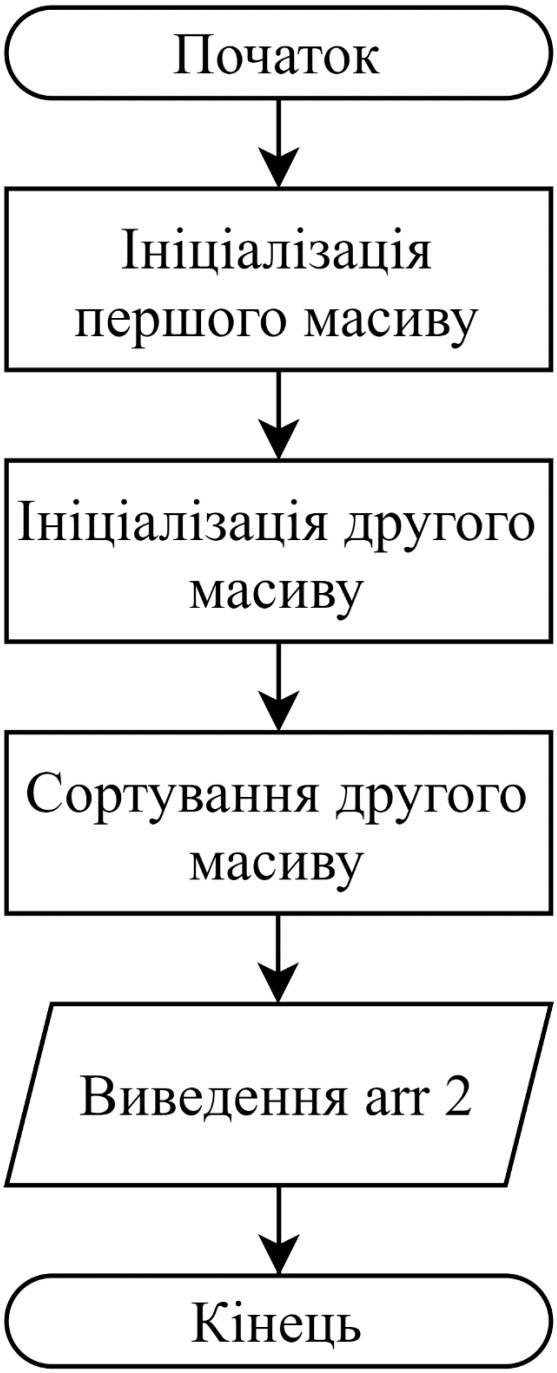
Крок 4.

```
початок
    arr1 = arrRand(arr1)
    arr2 = averageNeg(arr1, arr2)
    arr2 = sortShell(arr2, 8)
кінець

функція arrRand(arr1[][])
    повторити для i = 0; i < 5; i++
        повторити для k = 0; k < 8; l++
            arr1[i][k] = rand() % 199 - 99
        все повторити
    все повторити
    повернути arr1
кінець функції

функція averageNeg(arr1[][], ar2[])
    повторити для k = 0; k < 8; k++
        n = 0
        sum = 0
        повторити для i = 0; i < 5; i++
            якщо arr1[i][k] < 0
                sum += arr[i][k]
                n++
            все якщо
        все повторити
        якщо n != 0
            arr2[k] = sum / n
        інакше
            arr2[k] = 0
        все якщо
```

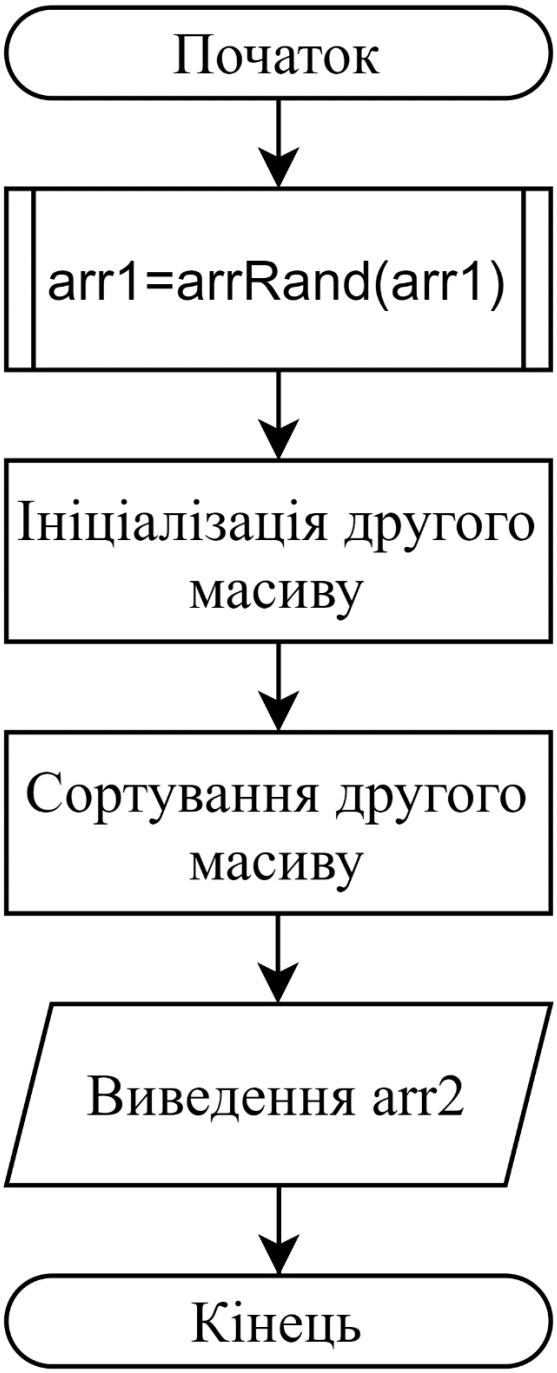
```
    все повторити
    повернути arr2
кінець функції

функція sortShell(arr2[], n)
    повторити для d = n/2, d >= 1, d /= 2
        повторити для i = d, i < n, i++
            повторити для j = i, j >= d && arr2[j-d] < arr2[j], j -= d
                swap(arr2[j], arr2[j-d])
            все повторити
        все повторити
    все повторити
    повернути arr2
кінець функції
```

Блок-схема алгоритму

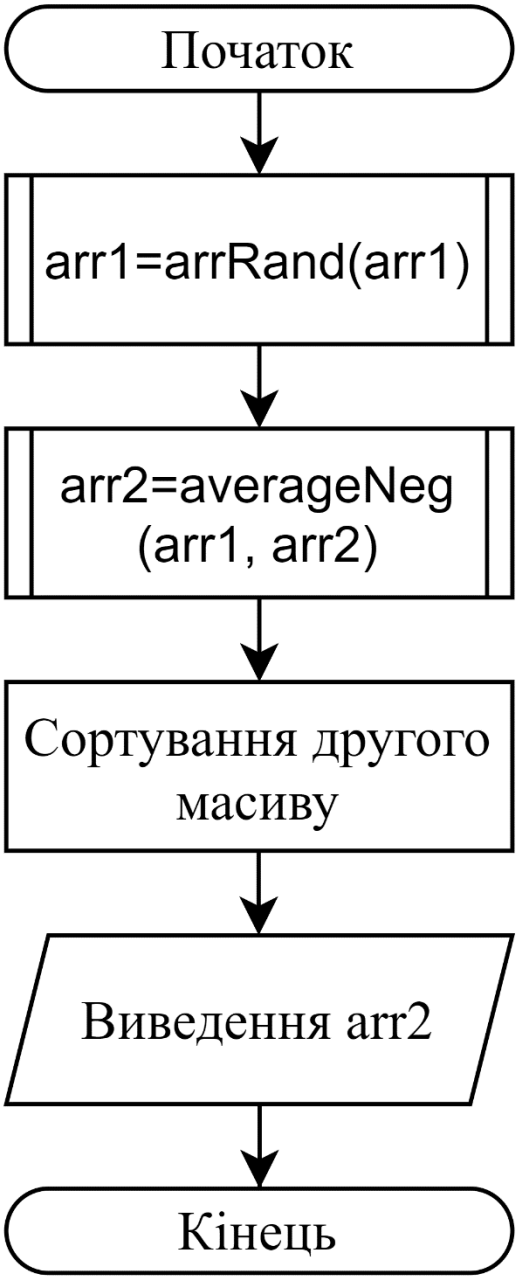
Крок 1


Початок
Ініціалізація першого масиву
Ініціалізація другого масиву
Сортування другого масиву
Виведення arr 2
Кінець

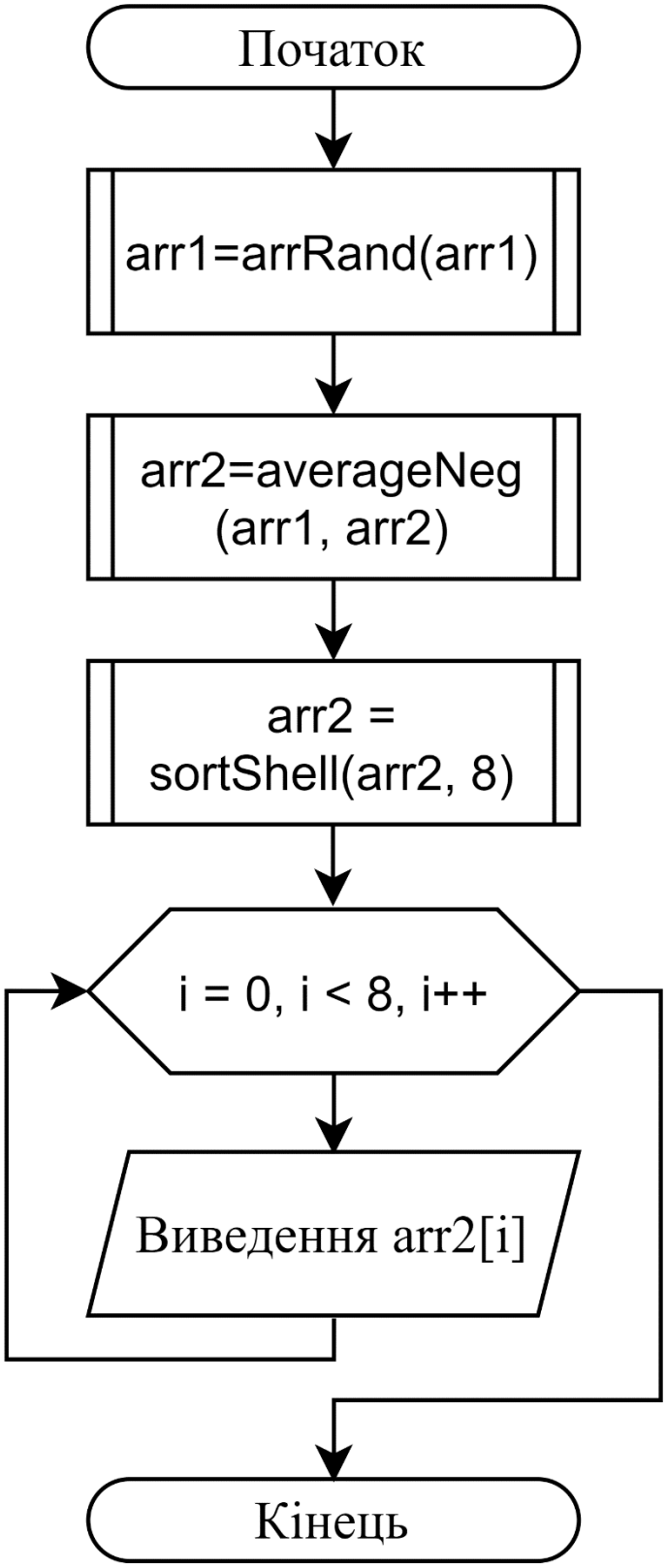
Крок 2

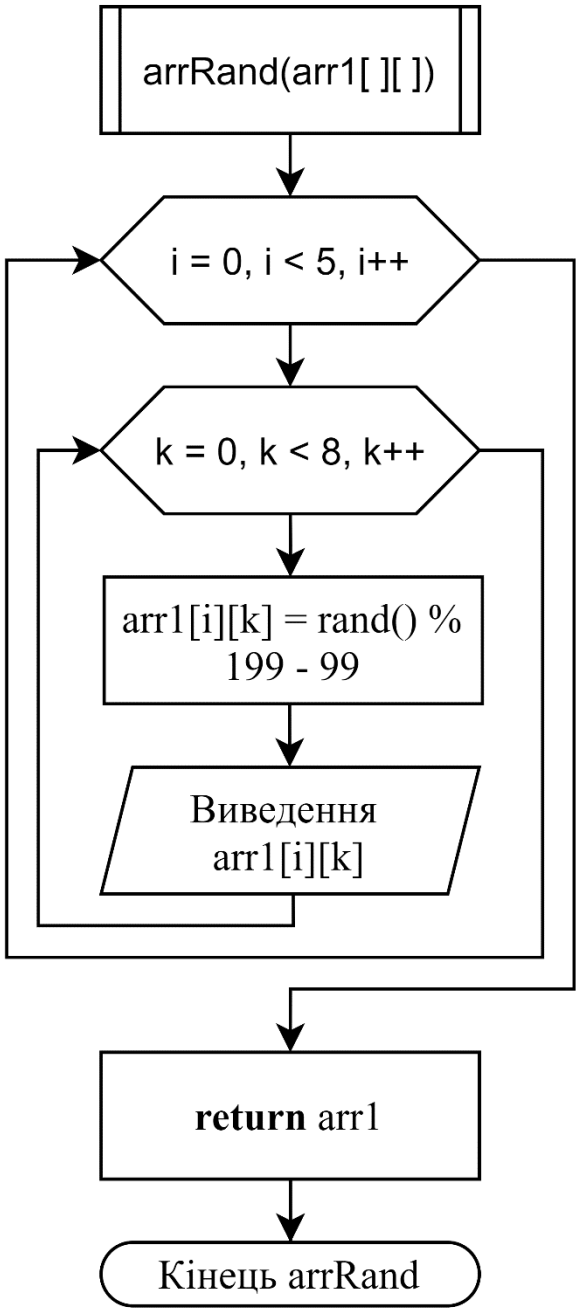

Початок
arr1=arrRand(arr1)
Ініціалізація другого масиву
Сортування другого масиву
Виведення arr2
Кінець

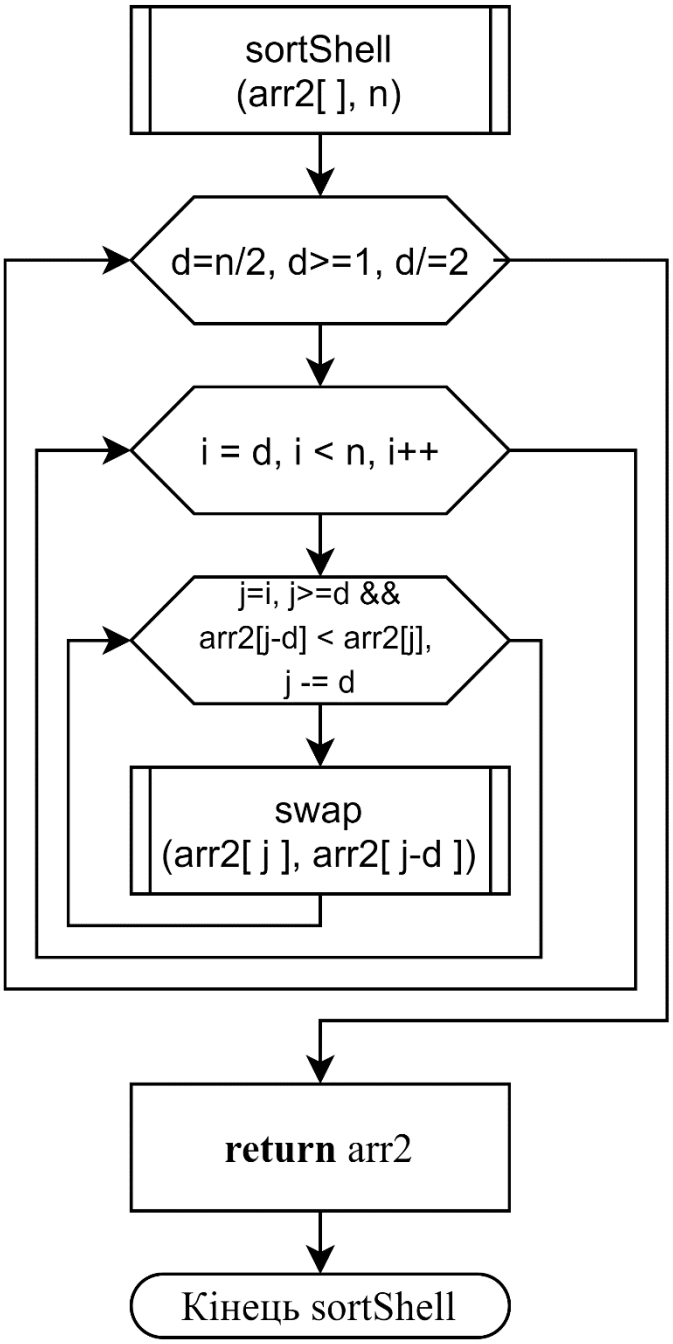
Крок 3

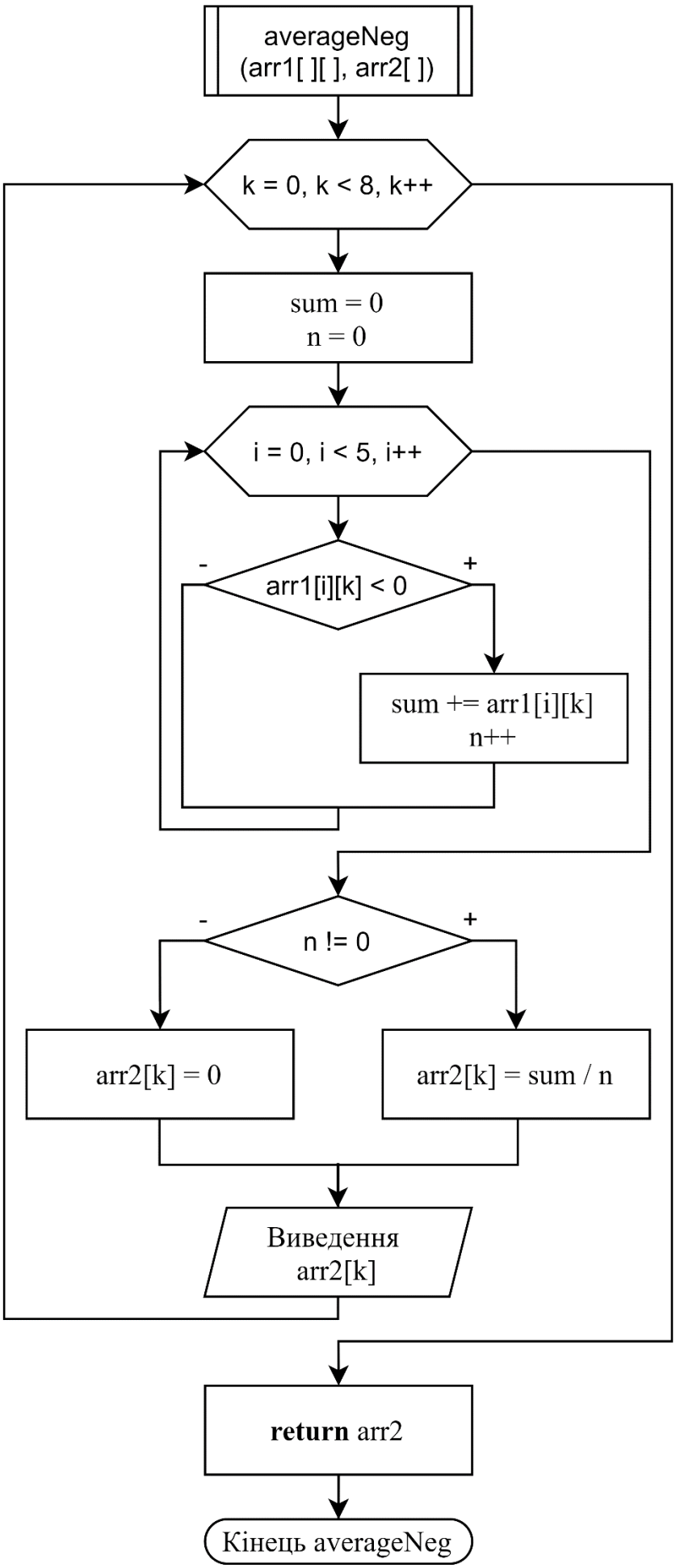

Початок
arr1=arrRand(arr1)
arr2=averageNeg (arr1, arr2)
Сортування другого масиву
Виведення arr2
Кінець

Крок 4


Початок
arr1=arrRand(arr1)
arr2=averageNeg (arr1, arr2)
arr2 = sortShell(arr2, 8)
i = 0, i < 8, i++
Виведення arr2[i]
Кінець


arrRand(arr1[ ][ ])
i = 0, i < 5, i++
k = 0, k < 8, k++
arr1[i][k] = rand() % 199 - 99
Виведення arr1[i][k]
return arr1
Кінець arrRand


sortShell (arr2[ ], n)
d=n/2, d>=1, d/=2
i = d, i < n, i++
j=i, j>=d && arr2[j-d] < arr2[j], j -= d
swap (arr2[j], arr2[j-d])
return arr2
Кінець sortShell


averageNeg
(arr1[ ][ ], arr2[ ])
k = 0, k < 8, k++
sum = 0
n = 0
i = 0, i < 5, i++
-
arr1[i][k] < 0
+
sum += arr1[i][k]
n++
-
n != 0
+
arr2[k] = 0
arr2[k] = sum / n
Виведення
arr2[k]
return arr2
Кінець averageNeg

Код програми (C++).

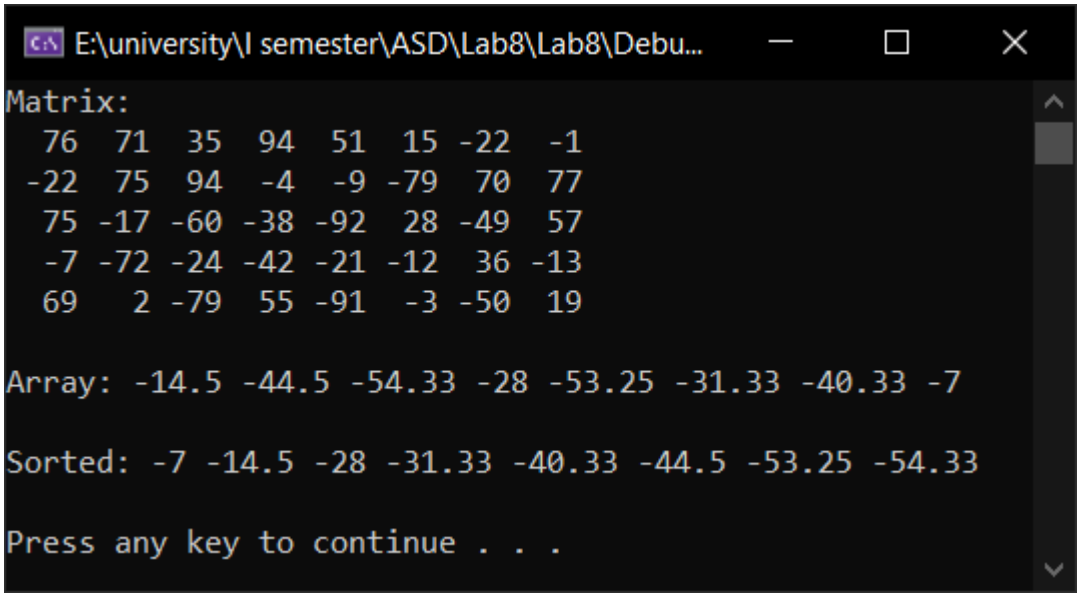
```cpp
#include <iostream>
#include <iomanip>
using namespace std;

void arrRand(int arr1[][8]);
void averageNeg(int arr1[][8], float arr2[]);
void sortShell(float arr2[], int);

int main()
{
    int arr1[5][8];
    arrRand(arr1);
    float arr2[8];
    averageNeg(arr1, arr2);
    sortShell(arr2, 8);
    cout << "Sorted: ";
    for (int i = 0; i < 8; i++)
    {
        cout << setprecision(4) << arr2[i] << " ";
    }
    cout << endl << endl;
    system("pause");
}
```

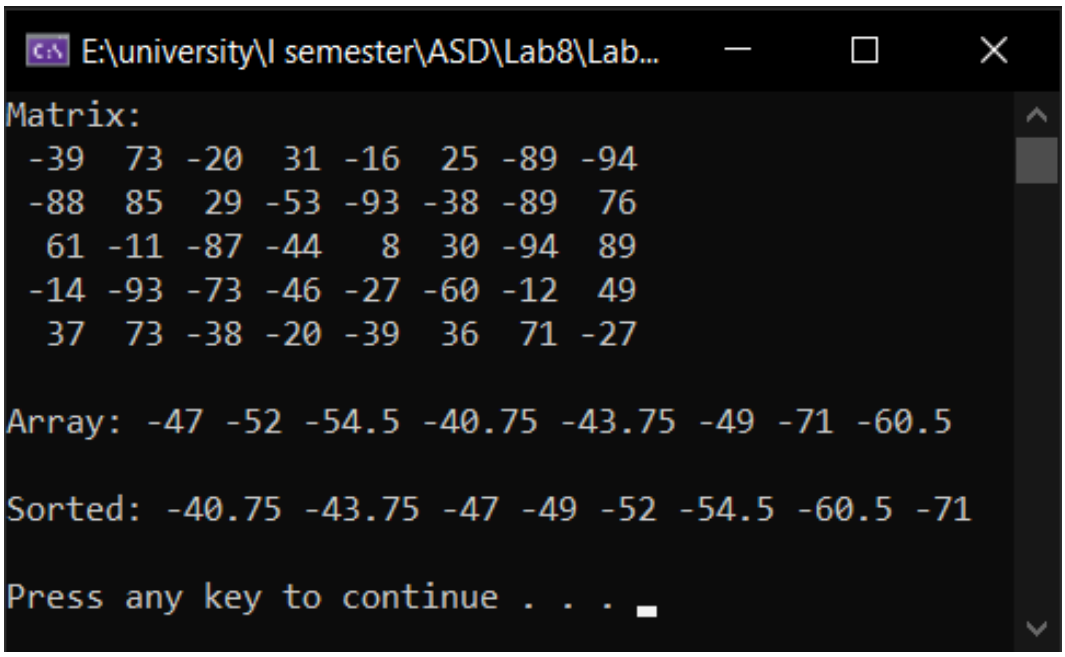
```cpp
void arrRand(int arr1[][8])
{
    srand(time(NULL));
    cout << "Matrix: " << endl;
    for (int i = 0; i < 5; i++)
    {
        for (int k = 0; k < 8; k++)
        {
            arr1[i][k] = rand() % 199 - 99;
            cout << setw(4) << arr1[i][k];
        }
        cout << endl;
    }
    cout << endl;
}
```

```
41  void averageNeg(int arr1[][8], float arr2[])
42  {
43      cout << "Array: ";
44      int n, sum;
45      for (int k = 0; k < 8; k++)
46      {
47          sum = 0;
48          n = 0;
49          for (int i = 0; i < 5; i++)
50          {
51              if (arr1[i][k] < 0)
52              {
53                  sum += arr1[i][k];
54                  n++;
55              }
56          }
57          if (n)
58              arr2[k] = float(sum) / n;
59          else
60              arr2[k] = 0;
61          cout << setprecision(4) << arr2[k] << " ";
62      }
63      cout << endl << endl;
64  }
```

```
66  void sortShell(float arr2[], int n)
67  {
68      for (int d = n / 2; d >= 1; d /= 2)
69          for (int i = d; i < n; i++)
70              for (int j = i; j >= d && arr2[j - d] < arr2[j]; j -= d)
71                  swap(arr2[j], arr2[j - d]);
72  }
```

Результат виконання програми.





Перевірка першого виконання.

Одновимірний масив:

1. (-22-7)/2 = -14,5
2. (-17-72)/2 = -44,5
3. (-60-24-79)/3 = -54,3
4. (-4-38-42)/3 = -28
5. (-9-92-21-91)/4 = -53,25
6. (-79-12-3)/3 = -31,3
7. (-22-49-50)/3 = -40,3
8. (-1-13)/2 = -7

Перевірка сортування: -7 > -14,5 > -28 > -31,3 > -40,3 > -44,5 > -53,25 > -54,3

Висновок.

При виконанні лабораторної роботи було використано матрицю – іменовану сукупність послідовностей значень одного типу, де кожен елемент має два порядкові номери. Для пошуку від’ємних значень двовимірного масиву було використано два вкладених оператори повторення. Також, в лабораторній роботі був використаний метод сортування Шелла, щоб відсортувати одновимірний масив за спаданням.