

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 32

Виконав студент ІІ-12 Федій Олександр Валерійович

(шифр, прізвище, ім'я, по батькові)

Перевірив _____

(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 8

Дослідження алгоритмів пошуку та сортування

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

Варіант 32

Задача 6.32.

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 Шелла 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 Шелла. Ввідні дані, що вимагаються – двовимірний масив, що буде заповнюватися випадковими значеннями.

Математична побудова. Складемо таблицю змінних.

Змінна	Тип	Ім'я	Призначення
Двовимірний масив	Цілий	arr1[5][8]	Початкове дане
Одновимірний масив	Цілий	arr2[8]	Результат
Лічильники	Цілий	i, k	Проміжне дане
Проміжні змінні сортування	Цілий	d, j	Проміжне дане

Кількість елементів масиву	Цілий	n	Проміжне дане
----------------------------	-------	---	---------------

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

Розв’язання:

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

Псевдокод підпрограми

Функція arrRand(arr1[[]])

повторити для i = 0; i < 5; i++

повторити для k = 0; k < 8; k++

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 $n \neq 0$

$arr2[k] = sum / n$

інакше

$arr2[k] = 0$

все якщо

все повторити

повернути arr2

Кінець функції

Функція sortShell(arr2[], n)

повторити для $d = n/2, d \geq 1, d /= 2$

повторити для $i = d, i < n, i++$

повторити для $j = i, j \geq d \ \&\& \ arr2[j-d] < arr2[j], j -= d$

$swap(arr2[j], arr2[j-d])$

все повторити

все повторити

все повторити

повернути arr2

Кінець функції

Блок-схема

Крок 1



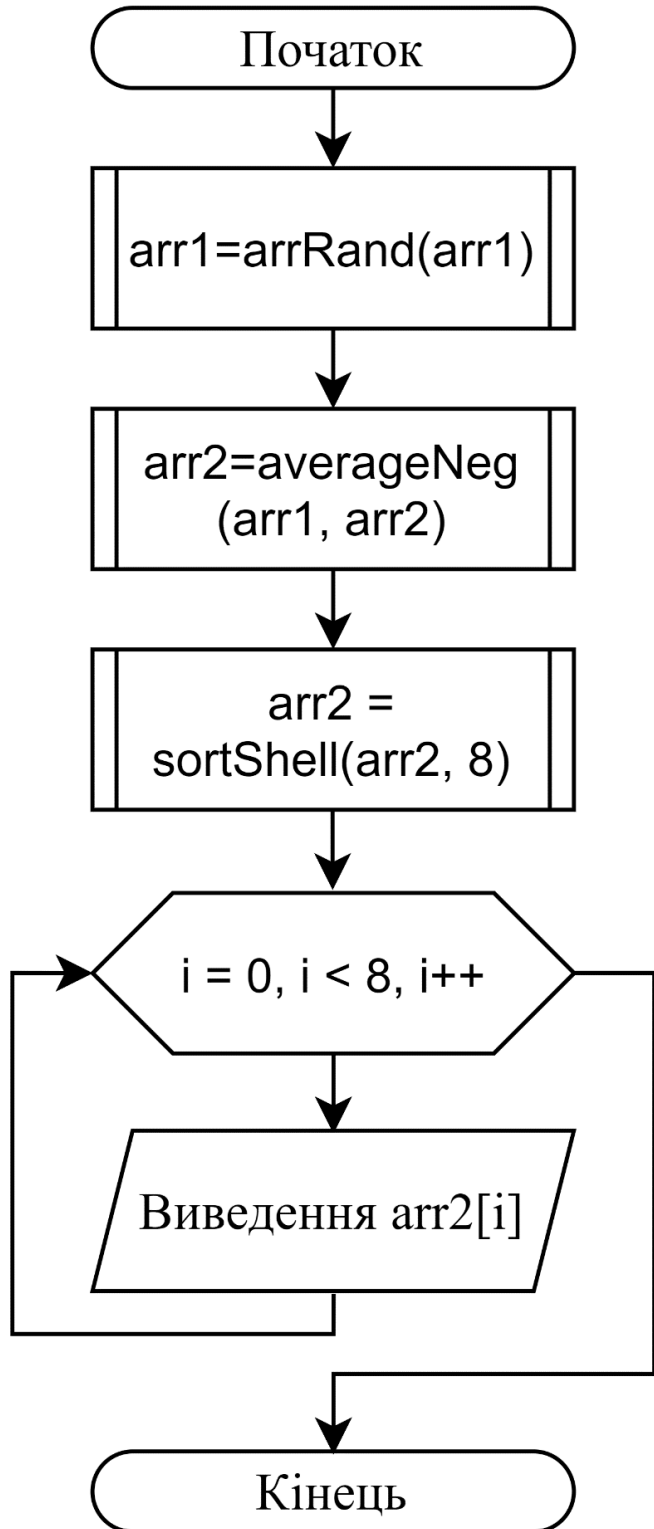
Крок 2

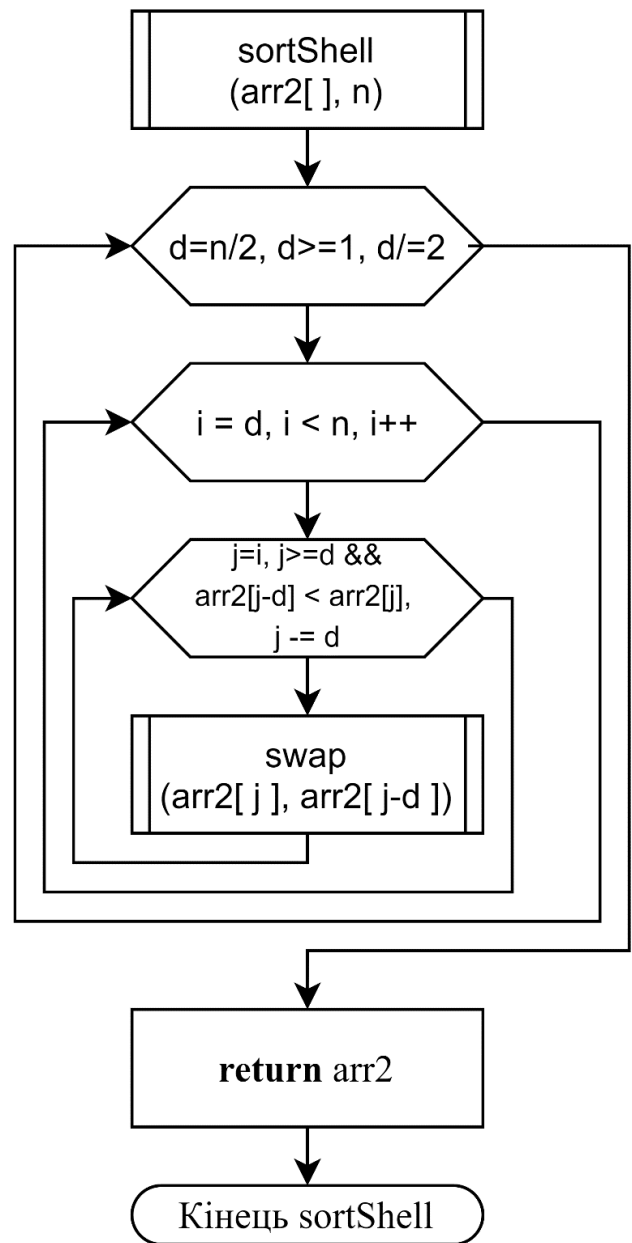
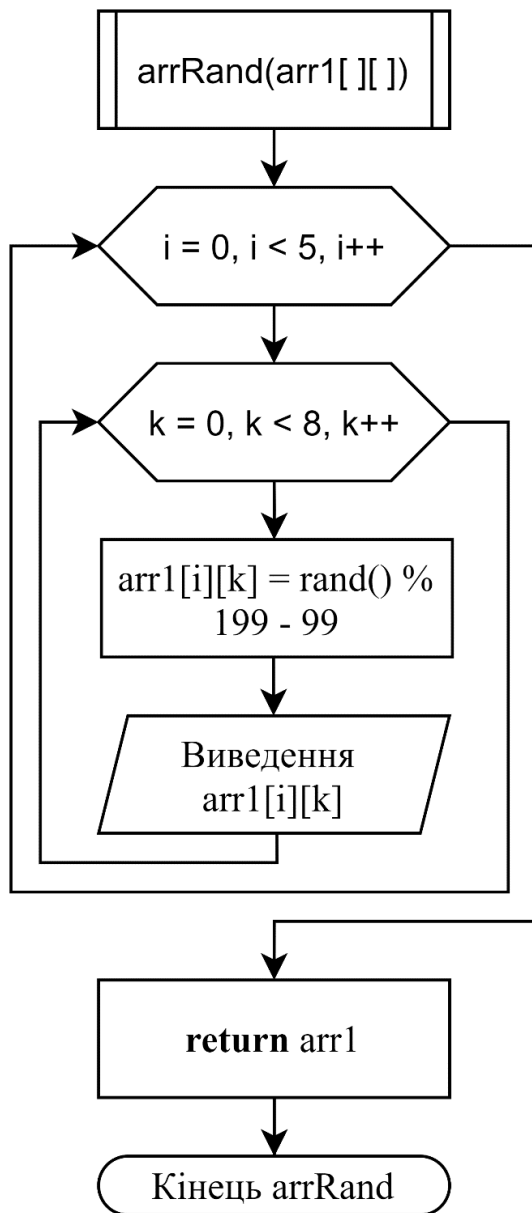


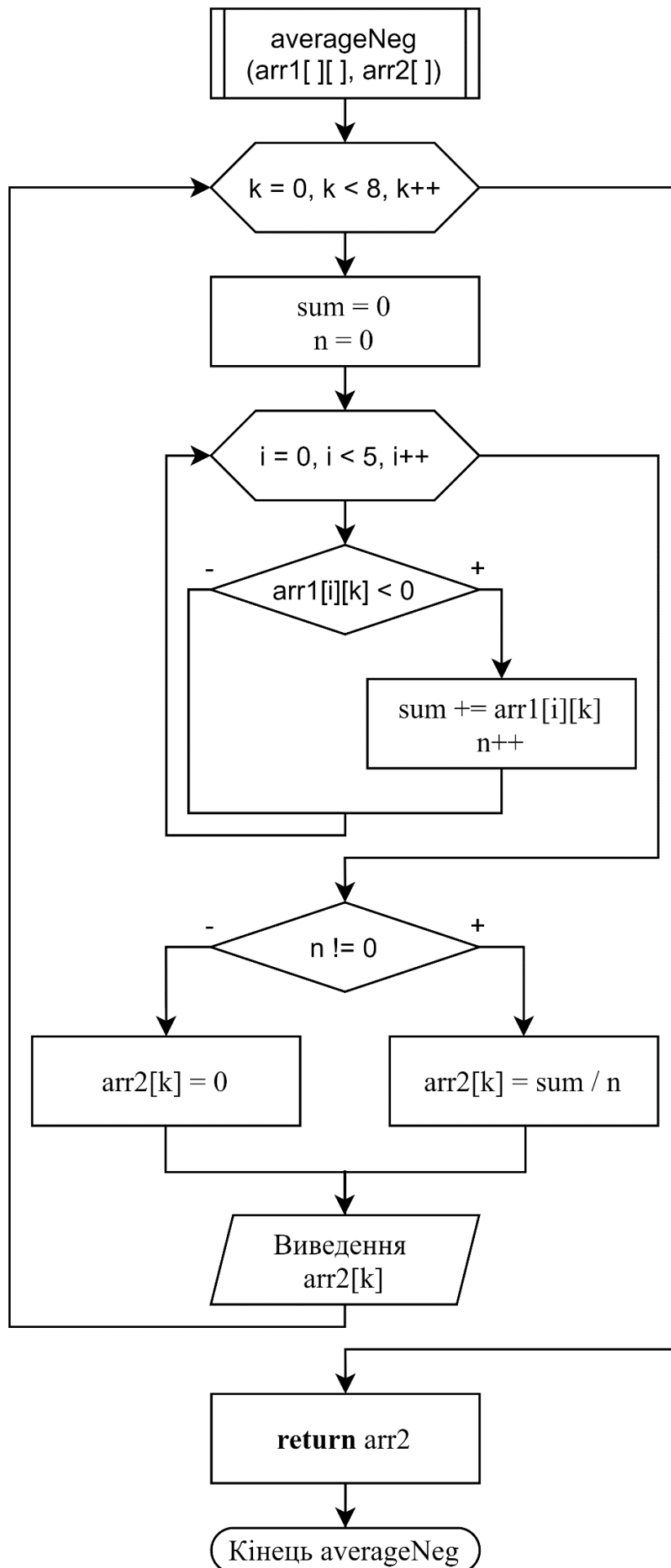
Крок 3



Крок 4







Код програми

```
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  void arrRand(int arr1[][8]);
6  void averageNeg(int arr1[][8], float arr2[]);
7  void sortShell(float arr2[], int);
8
9  int main()
10 {
11     int arr1[5][8];
12     arrRand(arr1);
13     float arr2[8];
14     averageNeg(arr1, arr2);
15     sortShell(arr2, 8);
16     cout << "Sorted: ";
17     for (int i = 0; i < 8; i++)
18     {
19         cout << setprecision(4) << arr2[i] << " ";
20     }
21     cout << endl << endl;
22     system("pause");
23 }
24
25 void arrRand(int arr1[][8])
26 {
27     srand(time(NULL));
28     cout << "Matrix: " << endl;
29     for (int i = 0; i < 5; i++)
30     {
31         for (int k = 0; k < 8; k++)
32         {
33             arr1[i][k] = rand() % 199 - 99;
34             cout << setw(4) << arr1[i][k];
35         }
36         cout << endl;
37     }
38     cout << endl;
39 }
```

```

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 }

```

```

Matrix:
 76  71  35  94  51  15 -22  -1
-22  75  94  -4  -9 -79  70  77
 75 -17 -60 -38 -92  28 -49  57
 -7 -72 -24 -42 -21 -12  36 -13
 69   2 -79  55 -91  -3 -50  19

Array: -14.5 -44.5 -54.33 -28 -53.25 -31.33 -40.33 -7

Sorted: -7 -14.5 -28 -31.33 -40.33 -44.5 -53.25 -54.33

Press any key to continue . . .

```

```

Matrix:
-39  73 -20  31 -16  25 -89 -94
-88  85  29 -53 -93 -38 -89  76
 61 -11 -87 -44   8  30 -94  89
-14 -93 -73 -46 -27 -60 -12  49
 37  73 -38 -20 -39  36  71 -27

Array: -47 -52 -54.5 -40.75 -43.75 -49 -71 -60.5

Sorted: -40.75 -43.75 -47 -49 -52 -54.5 -60.5 -71

Press any key to continue . . .

```

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

Під час виконання лабораторної роботи було досліджено методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набуто практичних навичок їх використання під час складання програмних специфікацій. 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.