

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 20

Виконав студент

ІП-15, Ликова Катерина Олександрівна
(шифр, прізвище, ім'я, по батькові)

Перевірив

(прізвище, ім'я, по батькові)

Київ 2021

Мета – дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Постановка задачі

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).

Табл.1

№ варіанта	Розмірність	Тип даних	Обчислення значень елементів одновимірного масиву
20	5 x 4	Цілий	Із середнього арифметичного від'ємних значень елементів рядків двовимірного масиву. Відсортувати методом Шела за спаданням.

Математична модель

Змінна	Тип	Ім'я	Призначення
Кількість рядків матриці	Цілий	m	Початкові дані
Кількість стовпців матриці	Цілий	n	Початкові дані
Матриця	Цілий	A[][]	Проміжні дані
Максимальний коефіцієнт масиву В	Цілий	l	Проміжні дані
Масив, що містить середні арифметичні від'ємних значень елементів рядків матриці	Цілий	B[]	Результат
Функція, що надає матриці А значення елементів	Порожній	input	Проміжні дані
Функція, що виводить значення елементів матриці А	Порожній	output	Проміжні дані
Функція, що надає масиву В значення елементів	Порожній	arith_mean_row_neg	Проміжні дані

Функція, що виводить значення елементів масиву В	Порожній	arith_output	Проміжні дані
Функція, що сортує значення елементів масиву В методом Шела за спаданням	Порожній	sort_shell	Проміжні дані
Лічильник циклів	Цілий	i	Проміжні дані
Лічильник циклів	Цілий	j	Проміжні дані
Сума від'ємних значень елементів рядків матриці А	Цілий	s	Проміжні дані
Кількість від'ємних значень елементів рядків матриці А	Цілий	k	Проміжні дані
Змінна для перестановки двох значень елементів масиву В	Цілий	t	Проміжні дані
Змінна для розбиття значень елементів масиву В на 2	Цілий	d	Проміжні дані
Параметр функцій input, output, arith_mean_row_neg	Цілий	matr	Проміжні дані
Параметр функцій input, output, arith_mean_row_neg, arith_output, sort_shell	Цілий	p1	Проміжні дані
Параметр функцій input, output, arith_mean_row_neg	Цілий	p2	Проміжні дані
Параметр функцій arith_mean_row_neg, arith_output, sort_shell	Цілий	arr	Проміжні дані
Параметр функцій arith_mean_row_neg	Цілий	p3	Проміжні дані

Для вирішення даного завдання потрібно спочатку присвоїти значення змінним m і n . Після цього за допомогою функції input, що включає в себе функцію $\text{rand}() \% x + y$, яка приймає різні числові значення у діапазоні від x до $x+y-1$, присвоюємо значення

матриці $A[][]$ та виводимо їх функцією `output`. Далі треба присвоїти значення змінній `l` та присвоїти середні арифметичні від'ємних значень елементів рядків матриці $A[][]$ масиву $B[]$, застосувавши функцію `arith_mean_row_neg`, а також виводимо знайдені значення за допомогою функції `arith_output`. Наступним кроком стане сортування отриманого масиву методом Шела за спаданням, використовуючи функцію `sort_shell` та виведення кінцевого результату функцією `arith_output`.

Розв'язання

Крок 1. Визначимо основні дії.

Крок 2. Присвоюємо значення змінній `m,n`.

Крок 3. Присвоюємо значення змінній $A[][]$ та виводимо її.

Крок 4. Присвоюємо значення змінній `l`.

Крок 5. Деталізуємо дії знаходження змінної $B[]$ та виводимо її.

Крок 6. Сортуємо змінну $B[]$ та виводимо її.

Псевдокод

Основна програма

крок 1

початок

присвоєння значення змінній `m,n`

присвоєння значення змінній $A[][]$ та її вивід

присвоєння значення змінній `l`

деталізація дії знаходження змінної $B[]$ та її вивід

сортування змінної B та її вивід

кінець

крок 2

початок

`m=5, n=4`

присвоєння значення змінній $A[][]$ та її вивід

присвоєння значення змінній `l`

деталізація дії знаходження змінної $B[]$ та її вивід

сортування змінної B та її вивід

кінець

крок 3

початок

m=5, n=4

input(A, m, n)

output(A, m, n)

присвоєння значення змінній l

деталізація дії знаходження змінної B[] та її вивід

сортування змінної B та її вивід

кінець

крок 4

початок

m=5, n=4

input(A, m, n)

output(A, m, n)

l=-1

деталізація дії знаходження змінної B[] та її вивід

сортування змінної B та її вивід

кінець

крок 5

початок

m=5, n=4

input(A, m, n)

output(A, m, n)

l=-1

arith_mean_row_neg(A, m, n, B, l)

arith_output(B, l)

сортування змінної B та її вивід

кінець

крок 6

початок

m=5, n=4

input(A, m, n)

output(A, m, n)

l=-1

arith_mean_row_neg(A, m, n, B, l)

arith_output(B, l)

sort_shell(B, l)

arith_output(B, l)

кінець

Підпрограми

input(matr, p1, p2)

повторити для i від 0 до p1

повторити для i від 0 до p2

matr[i][j] = rand() % 199 – 99

все повторити

все повторити

кінець input(matr, p1, p2)

output(matr, p1, p2)

повторити для i від 0 до p1

повторити для i від 0 до p2

виведення matr[i][j]

все повторити

все повторити

кінець output(matr, p1, p2)

arith_mean_row_neg(matr, p1, p2, arr, p3)

повторити для i від 0 до $p1$

$s = 0$

$k = 0$

повторити для i від 0 до $p2$

якщо $\text{matr}[i][j] < 0$

то

$s = s + \text{matr}[i][j]$

$k = k + 1$

все якщо

все повторити

якщо $s < 0$

то

$p3 = p3 + 1$

$\text{arr}[p3] = s / k$

все якщо

все повторити

return $p3$

кінець $\text{arith_mean_row_neg}(\text{matr}, p1, p2, \text{arr}, p3)$

$\text{arith_output}(\text{arr}, p1)$

повторити для i від 0 до $p1$

виведення $\text{arr}[i]$

все повторити

кінець $\text{arith_output}(\text{arr}, p1)$

$\text{sort_shell}(\text{arr}, p1)$

повторити для d від 1 до $p1/2$, крок $d = d/2$

повторити для i від $d+1$ до $p1$

повторити для j від 0 до $i-d$ і $\text{arr}[j] < \text{arr}[j+d]$, крок $j = j-d$

$t = \text{arr}[j]$

$\text{arr}[j] = \text{arr}[j + d]$

$\text{arr}[j + d] = t$

все повторити

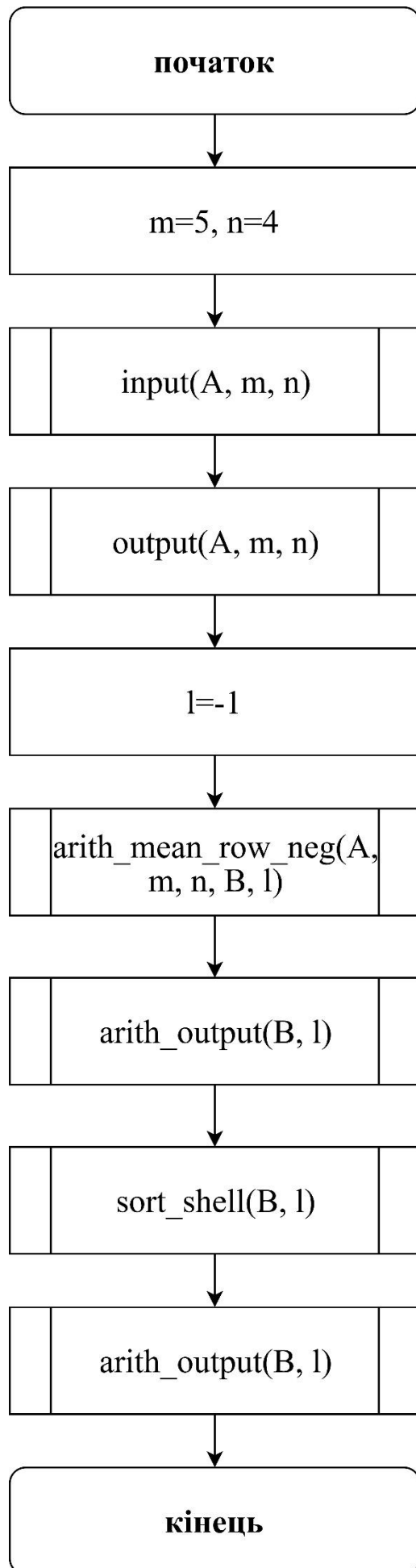
все повторити

все повторити

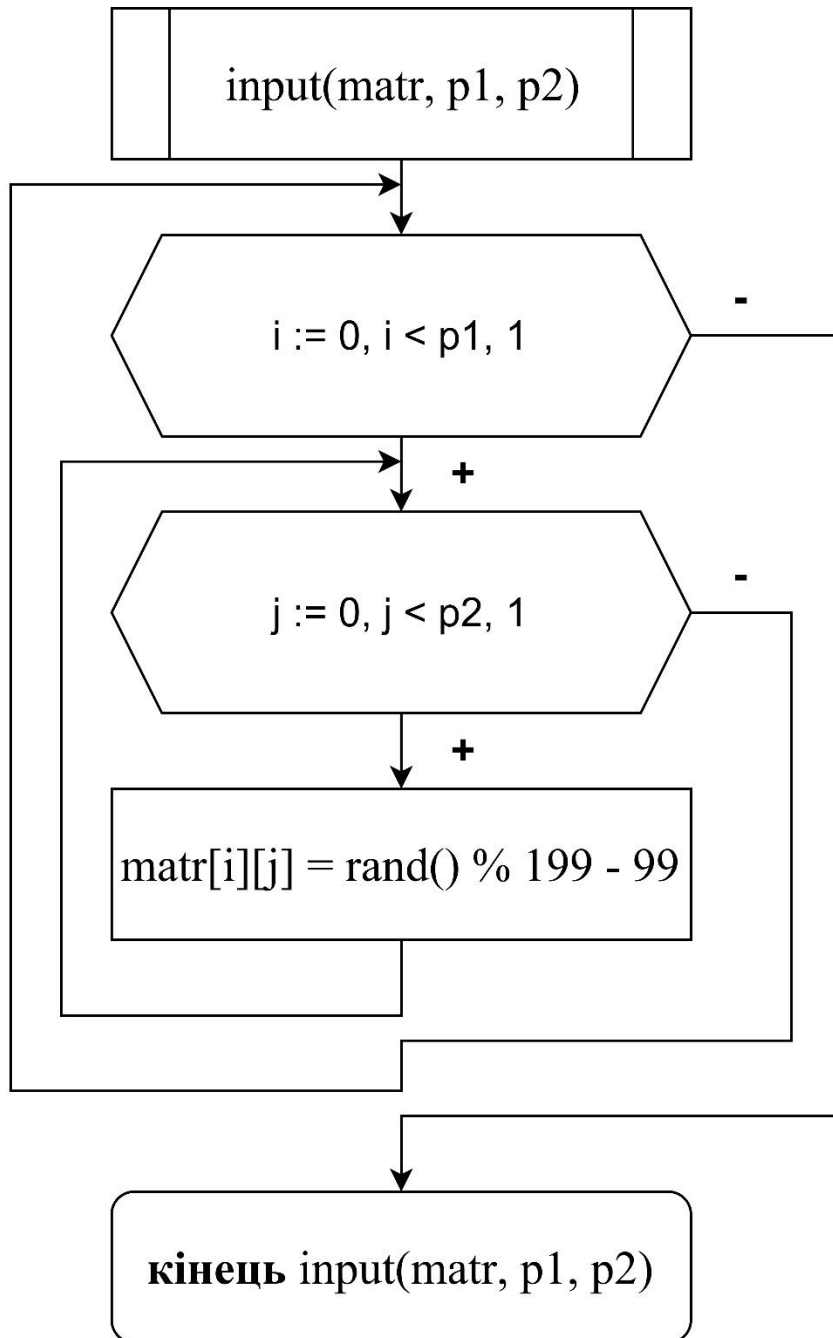
кінець sort_shell(arr, p1)

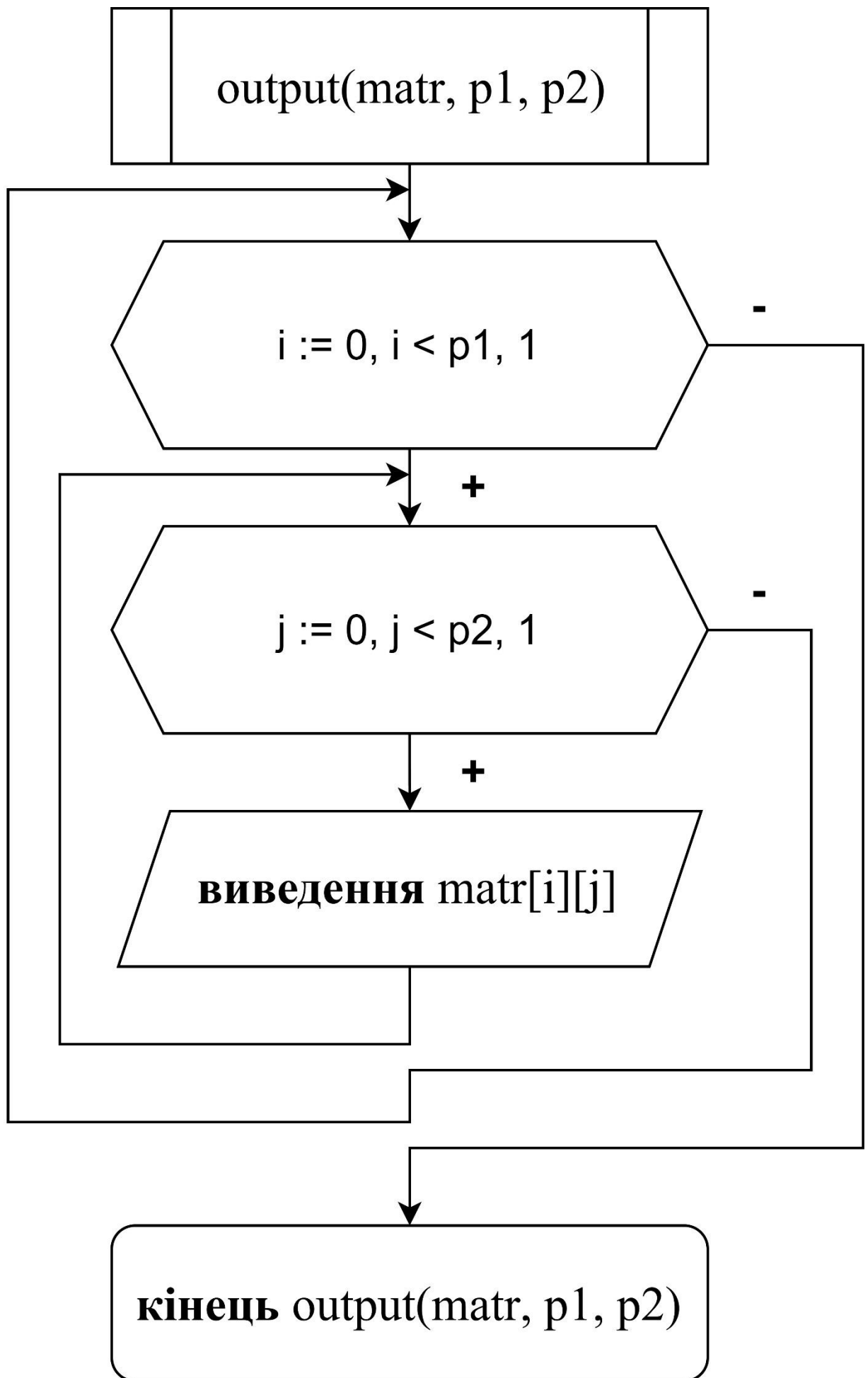
Блок-схема

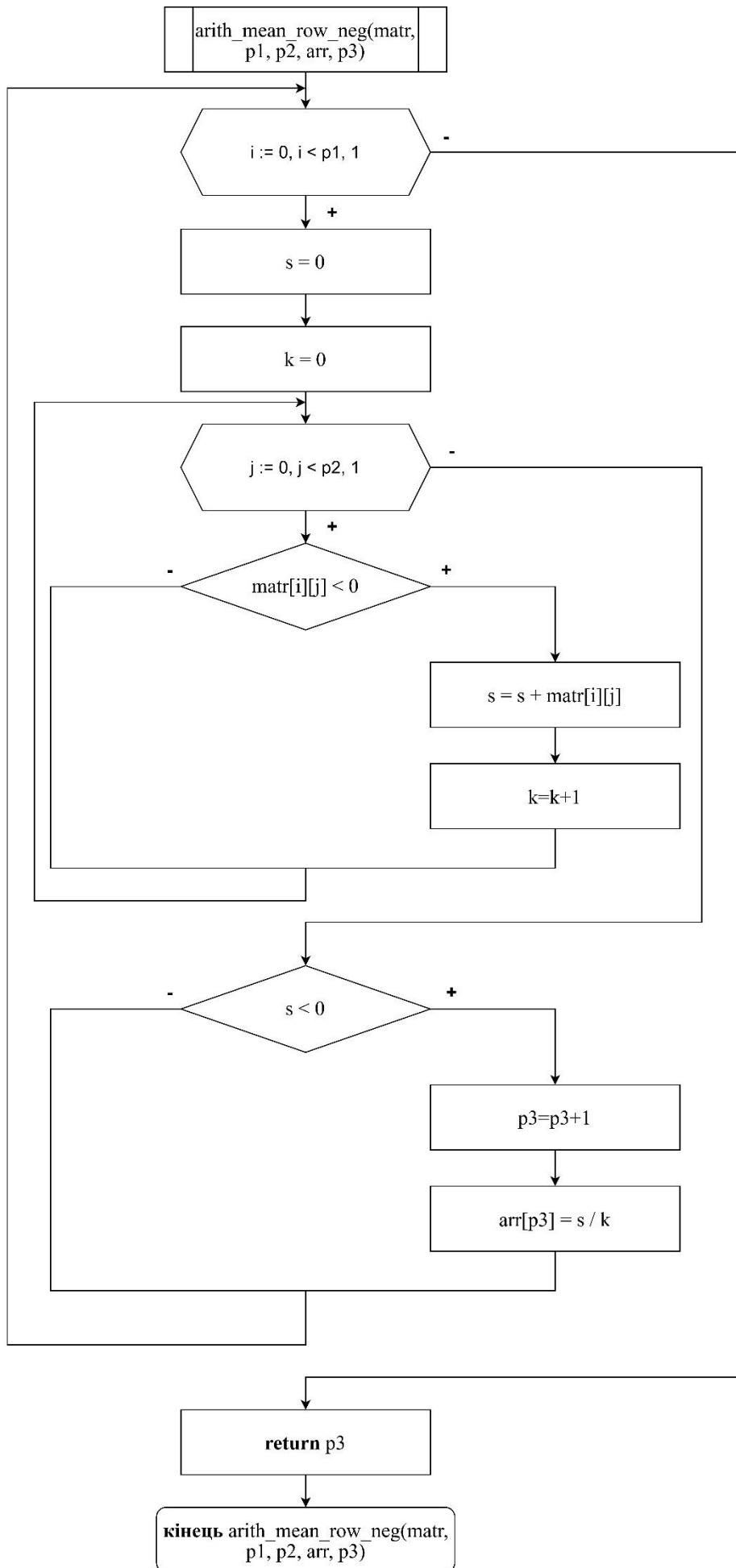
Основна програма

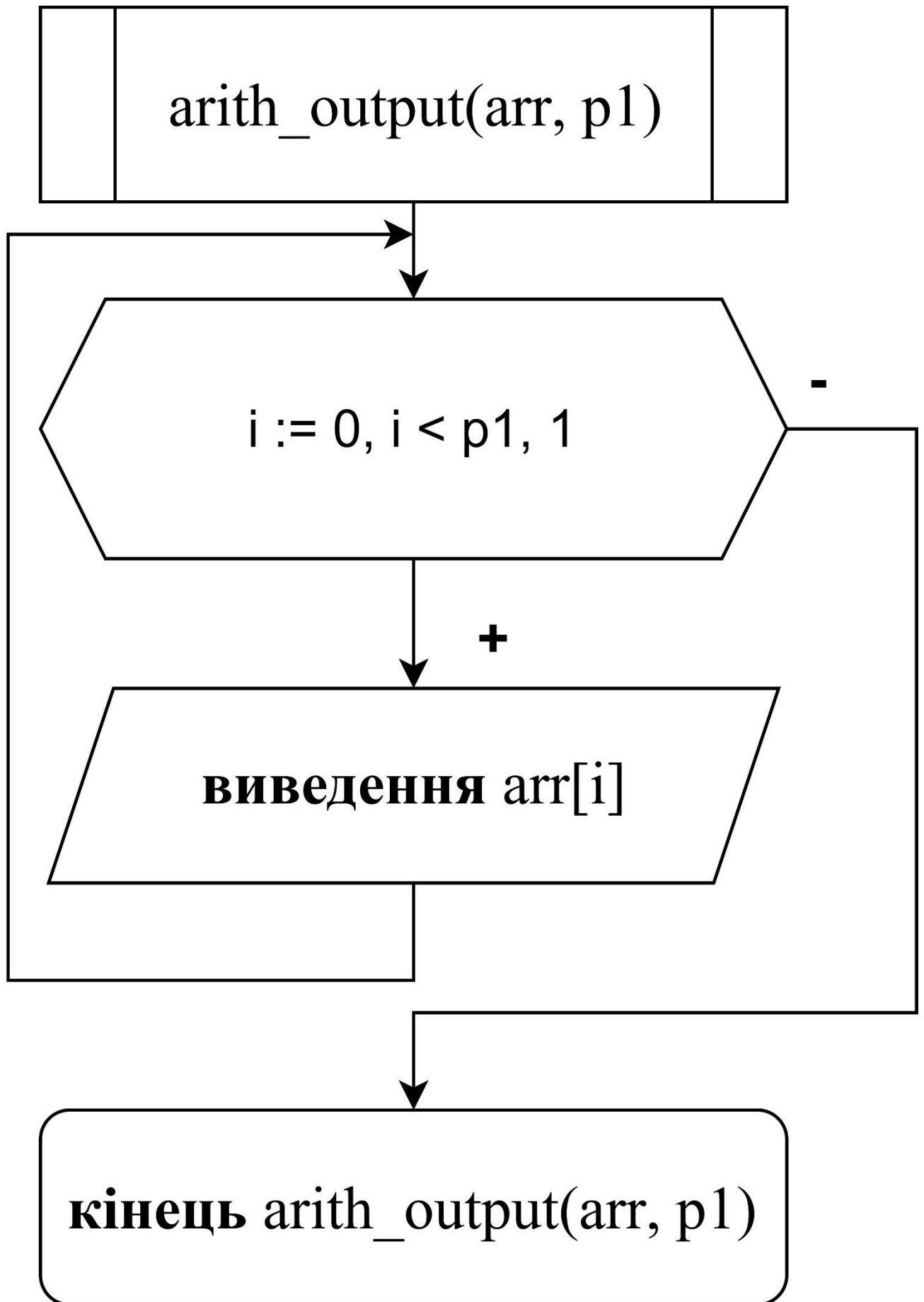


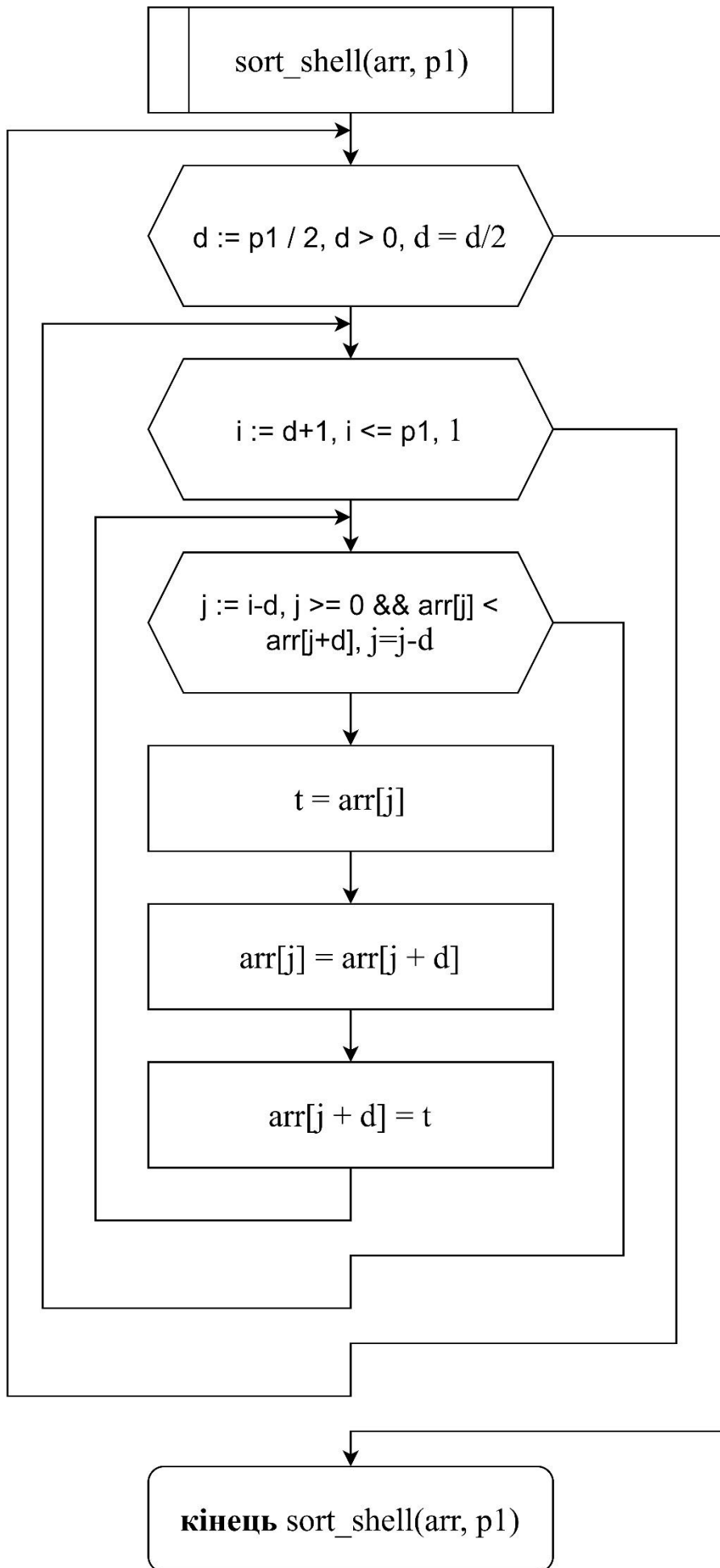
Підпрограми











Koð

```
#include <iostream>
#include <iomanip>
#include <time.h>
using namespace std;
typedef int Matrix[10][10];
typedef int Array[10];
Matrix A;
Array B;
void input(Matrix, int, int);
void output(Matrix, int, int);
void arith_mean_row_neg(Matrix, int, int, Array, int&);
void arith_output(Array, int);
void sort_shell(Array, int);
int main()
{
    srand(time(NULL));
    int m = 5;
    int n = 4;
    cout << "Matrix A:" << endl;
    input(A, m, n);
    output(A, m, n);
    int l = -1;
    cout << "Matrix B before sorting:" << endl;
    arith_mean_row_neg(A, m, n, B, l);
    arith_output(B, l);
    sort_shell(B, l);
    cout << "Matrix B after sorting:" << endl;
    arith_output(B, l);
    system("pause");
```

```

}
void input(Matrix matr, int p1, int p2)
{
for (int i = 0; i < p1; i++)
{
    for (int j = 0; j < p2; j++)
    {
        matr[i][j] = rand() % 199 - 99;
    }
}
}

void output(Matrix matr, int p1, int p2)
{
for (int i = 0; i < p1; i++)
{
    for (int j = 0; j < p2; j++)
    {
        cout << setw(4) << matr[i][j];
    }
    cout << endl;
}
cout << endl;
}

void arith_mean_row_neg(Matrix matr, int p1, int p2, Array arr, int &p3)
{
for (int i = 0; i < p1; i++)
{
    int s = 0;
    int k = 0;
    for (int j = 0; j < p2; j++)

```



```

        {
            if (matr[i][j] < 0)
            {
                s = s + matr[i][j];
                k++;
            }
        }
        if (s < 0)
        {
            p3++;
            arr[p3] = s / k;
        }
    }
}

void arith_output(Array arr, int p1)
{
    for (int i = 0; i <= p1; i++)
    {
        cout << setw(4) << arr[i];
    }
    cout << endl;
}

void sort_shell(Array arr, int p1)
{
    int t;
    for (int d = p1 / 2; d > 0; d /= 2)
    {
        for (int i = d; i <= p1; ++i)
        {
            for (int j = i-d; j >= 0 && arr[j] < arr[j+d]; j -= d)

```

```

        {
            t = arr[j];
            arr[j] = arr[j + d];
            arr[j + d] = t;
        }
    }
}
}

```

Тестування

```

Matrix A:
  95 -44  -7  76
 -29 -66  68  45
  17 -13 -27  72
 -17 -54  -2 -69
   2  47  -9  82

Matrix B before sorting:
 -25 -47 -20 -35  -9
Matrix B after sorting:
  -9 -20 -25 -35 -47
Для продовження натисніть будь-яку клавішу . . . █

```

```

Matrix A:
 -38 -97  25 -27
   0  70 -18  72
 -62  80 -68  58
 -53  94  99  68
  19  87 -38  61

Matrix B before sorting:
 -54 -18 -65 -53 -38
Matrix B after sorting:
 -18 -38 -53 -54 -65
Для продовження натисніть будь-яку клавішу . . . █

```

```
Matrix A:
 41  -7   7   7
 95  -7   9  65
-29  -8  96  20
 45  94  18  42
-82 -66  91  14

Matrix B before sorting:
 -7  -7 -18 -74
Matrix B after sorting:
 -7  -7 -18 -74
Для продовження натисніть будь-яку клавішу . . .
```

Висновки: Для побудови алгоритму розв’язання заданої задачі я створила допоміжні функції з ітераційними алгоритмами задання різних числових значень та сортування методом Шела за спаданням. Завдяки цьому я вивчила та дослідила властивості алгоритмів пошуку та сортування, отримала практичні навички використання цих алгоритмів під час складання програмних специфікацій.