# 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 19

Виконав студент: ІП-11 Лисенко Андрій Юрійович

Перевірила: Вітковська Ірина Іванівна

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.

**Індивідуальне завдання**

**Варіант 19**

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.

|  |  |  |  |
| --- | --- | --- | --- |
| 19 | 4 x 8 | Дійсний | Із добутку значень елементів стовпців  двовимірного масиву. Відсортувати методом  Шела за зростанням. |

**Постановка задачі**

Для алгоритму сортування Шела використаємо просту функцію swap() для виконання дії обміна значень між двома змінними. Функція get\_arr() обчислить масив дійсний чисел розмірністю 8, одним з параметрів цієї функції є згенерований випадковим чином матриця. Для генерування дійсних випадкових чисел використаємо модифіковану функцію rand() - rand\_float().

**Побудова математичної моделі**

Складемо таблицю змінних

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Розмірність матриці x, y | *int* | *size\_x, size\_y* | Використати для умови виходу із циклів |
| Точність | *int* | *precision* | Використати як аргумент функції rand\_float() |
| Діапазон | *int* | *range* | Зберігати дані про діапазон генерування випадкових чисел |
| Матриця | *double\*\** | *matrix* | Зберігати дані про елементи матриці |
| Масив | *double\** | *array* | Зберігати дані про елементи масиву |

Складемо таблицю підпрограм:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Підпрограма | Тип | Ім’я | Параметри | Призначення |
| Опис матриці | *double\*\** | *declr\_mtx* | *size\_x, size\_y* | Виділити пам'ять для матриці |
| Описі масиву | *double\** | *declr\_arr* | *size\_x* | Виділити пам'ять для масиву |
| Генерація матриці | *void* | *gen\_mtx* | *range, precision* | Згенерувати матрицю |
| Обчислення масиву | *void* | *get\_arr* | *matrix, size\_x, size\_y, range, precision* | Обчислити масив як добуток стовбців матриці |
| Виведення матриці | *void* | *out\_mtx* | *matrix, size\_x, size\_y, range, precision* | Виведення матриці |
| Виведення масиву | *void* | *out\_arr* | *array, size\_x, range, precision* | Виведення масиву |
| Обмін | *void* | *swap* | *a, b* | Обміняти значення двох змінних |
| Cортування Шела | *void* | *sort\_shell* | *array, size* | Виконати сортування шела |

**Розв’язання**

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

**Крок 1:** Визначення основних дій

**Крок 2:** Конкретизування дії генерування матриці

**Крок 3:** Конкретизування дії обчислення масиву

**Крок 4:** Конкретизування дії сортування масиву

**Псевдокод**

**Основна програма:**

**Початок**

*matrix = declr\_mtx(size\_x, size\_y);*

*array = declr\_mtx(size\_x);*

**Введення** *range, precision***;**

*gen\_mtx(matrix, size\_x, size\_y, range, precision);*

*get\_arr(matrix, array, size\_x, size\_y);*

*out\_mtx(matrix, size\_x, size\_y, range, precision);*

*out\_arr(array, size\_x, range, precision);*

*sort\_shell(array, size\_x);*

*out\_arr(array, size\_x, range, precision);*

**Кінець**

**Підпрограма *declr\_mtx():***

**Початок *declr\_mtx()***

*matrix = new double\* [size\_x];*

**Повторити для** *i* **від** *0* **до** *size\_x:*

*\*(matrix + i) = new double [size\_y];*

**Все повторити**

**Кінець *declr\_mtx()***

**Підпрограма *declr\_arr():***

**Початок *declr\_arr()***

*array = new double [size\_x];*

**Повернути** *array;*

**Кінець*****declr\_arr()***

**Підпрограма*****rand\_double():***

**Початок *rand\_double()***

*rand\_int = rand() % (2 \* range) - range;*

*rand\_frac = rand() % int(pow(10, precision));*

*random = rand\_int + rand\_frac \* pow(0.1, precision);*

**Повернути** *random;*

**Кінець *rand\_double()***

**Підпрограма *gen\_mtx():***

**Початок *gen\_mtx()***

**Повторити для***i***від***0***до** *size\_x****:***

**Повторитидля***j***від***0***до***size\_y****:***

*\*(\*(matrix + i) + j) = rand\_double(range, precision);*

**Все повторити**

**Все повторити**

**Кінець *gen\_mtx()***

**Підпрограма *get\_arr():***

**Початок *get\_arr()***

**Повторити для***i***від***0***до** *size\_x****:***

*\*(array + i) = 1;*

**Повторитидля***j***від***0***до***size\_y****:***

*\*(array + i) \*= \*(\*(matrix + i) + j);*

**Все повторити**

**Все повторити**

**Кінець *get\_arr()***

**Підпрограма *out\_mtx():***

**Початок *out\_mtx()***

**Повторитидля***j***від***0***до***size\_y****:***

**Повторити для***i***від***0***до** *size\_x****:***

**Виведення** *\*(\*(matrix + i) + j);*

**Все повторити**

**Все повторити**

**Кінець *out\_mtx()***

**Підпрограма *out\_arr():***

**Початок *out\_arr()***

**Повторити для***i***від***0***до** *size\_x****:***

**Виведення** *\*(array + i);*

**Все повторити**

**Кінець *out\_arr()***

**Підпрограма *swap():***

**Початок *swap()***

*c = \*a;*

*\*a = \*b;*

*\*b = c;*

**Кінець *swap()***

**Підпрограма *sort\_shell():***

**Початок** ***sort\_shell()***

**Повторитидля***d***від***size / 2***до***1* **з кроком** */2****:***

**Повторити для***i***від***d***до** *size****:***

**Повторити для***j***від***i* **поки** *j >= d && array[j - d] > array[j]****:***

*swap(array + j, array + j - d);*

**Все повторити**

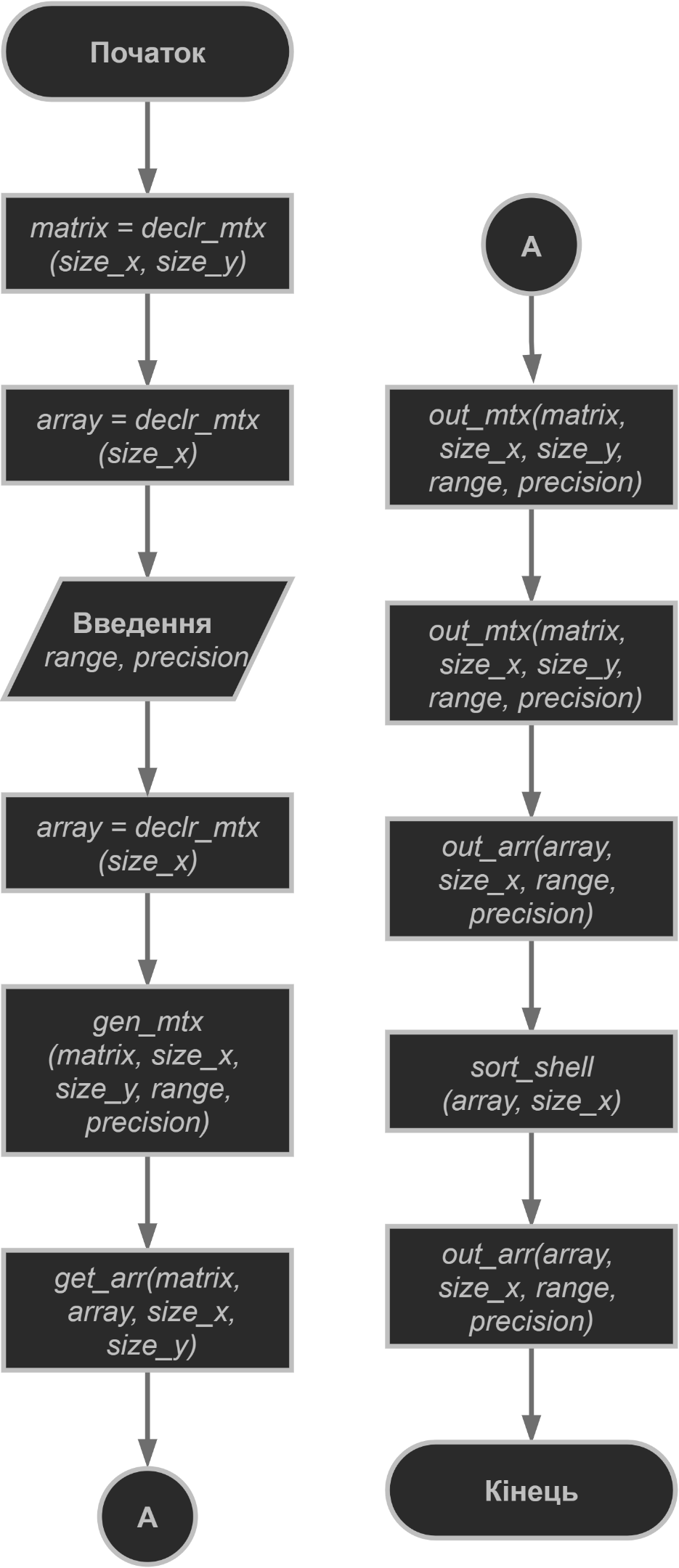
**Все повторити**

**Все повторити**

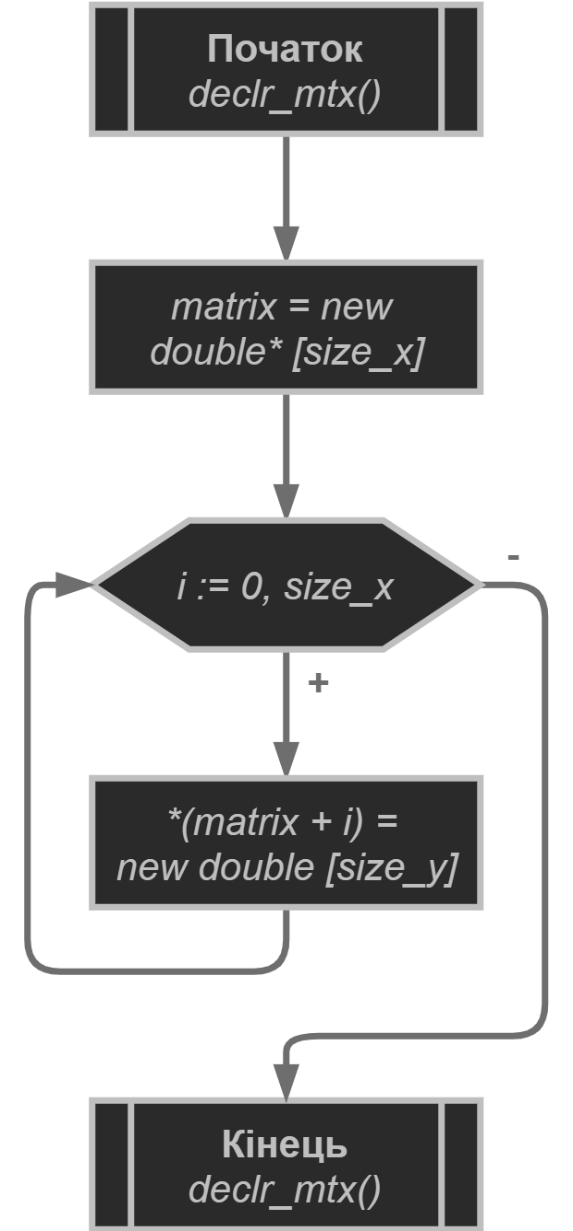
**Кінець *sort\_shell()***

**Блок-схема**

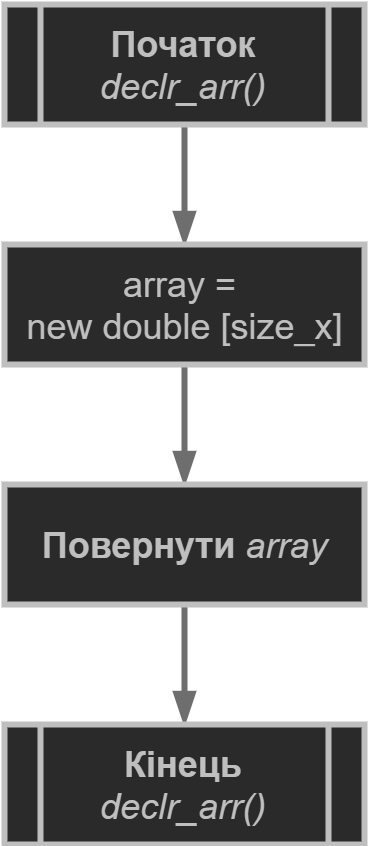
**Основна програма:**

****

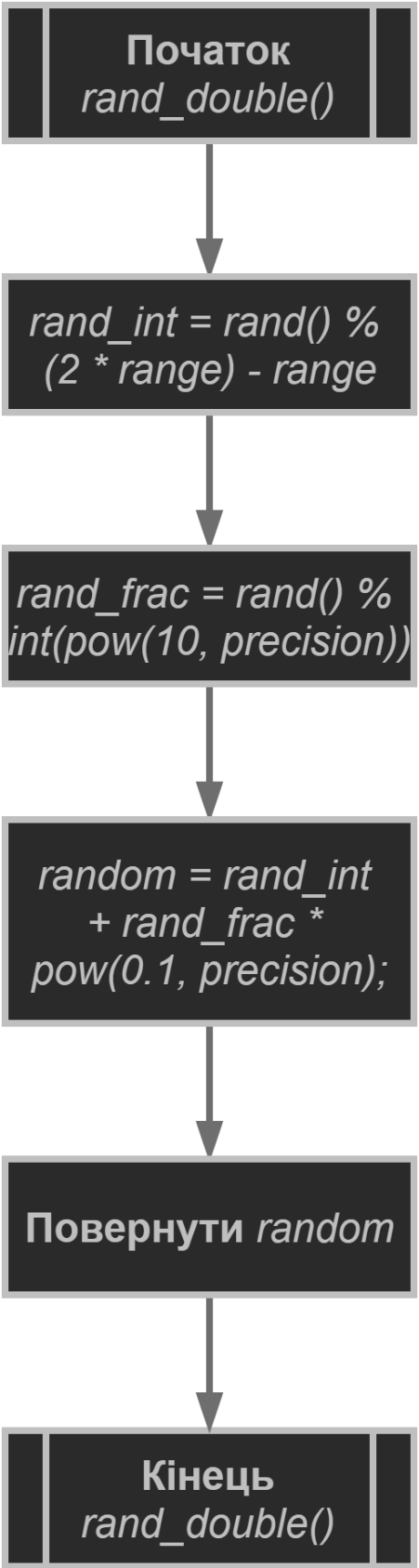
**Підпрограма declr\_mtx():**

****

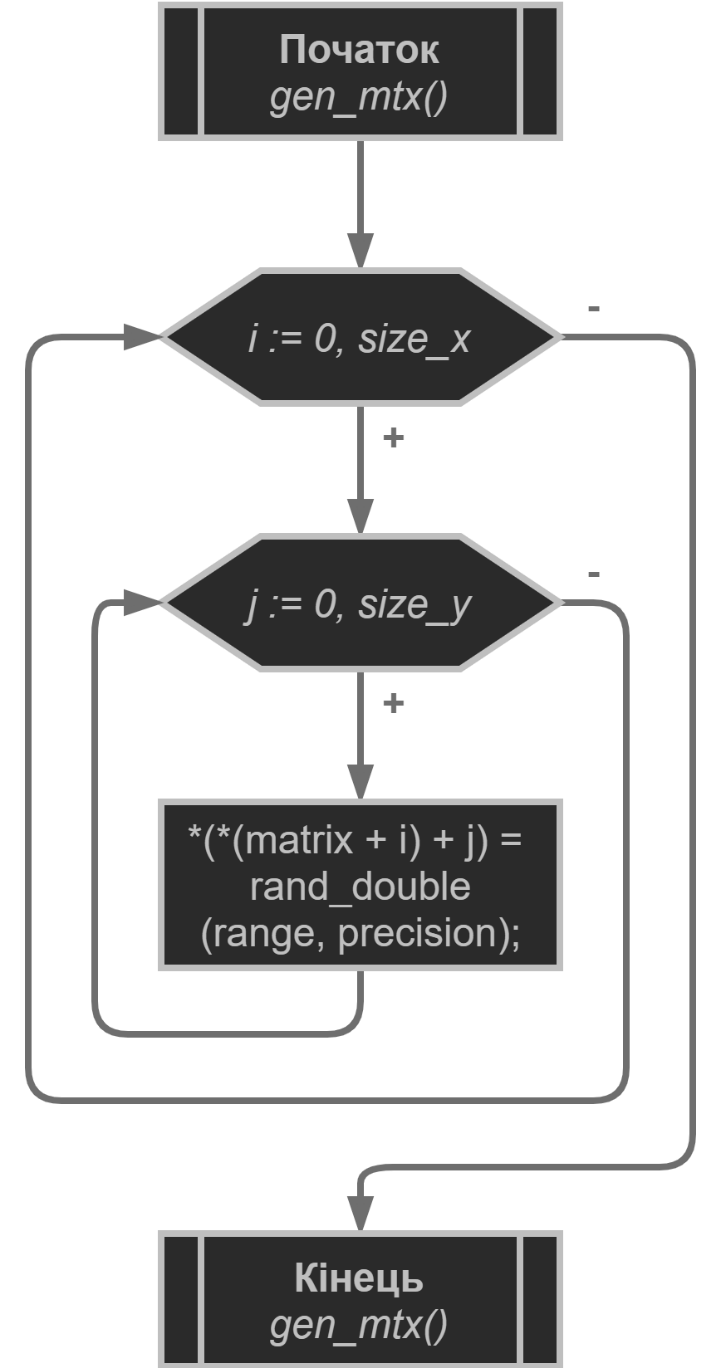
**Підпрограма declr\_arr():**

****

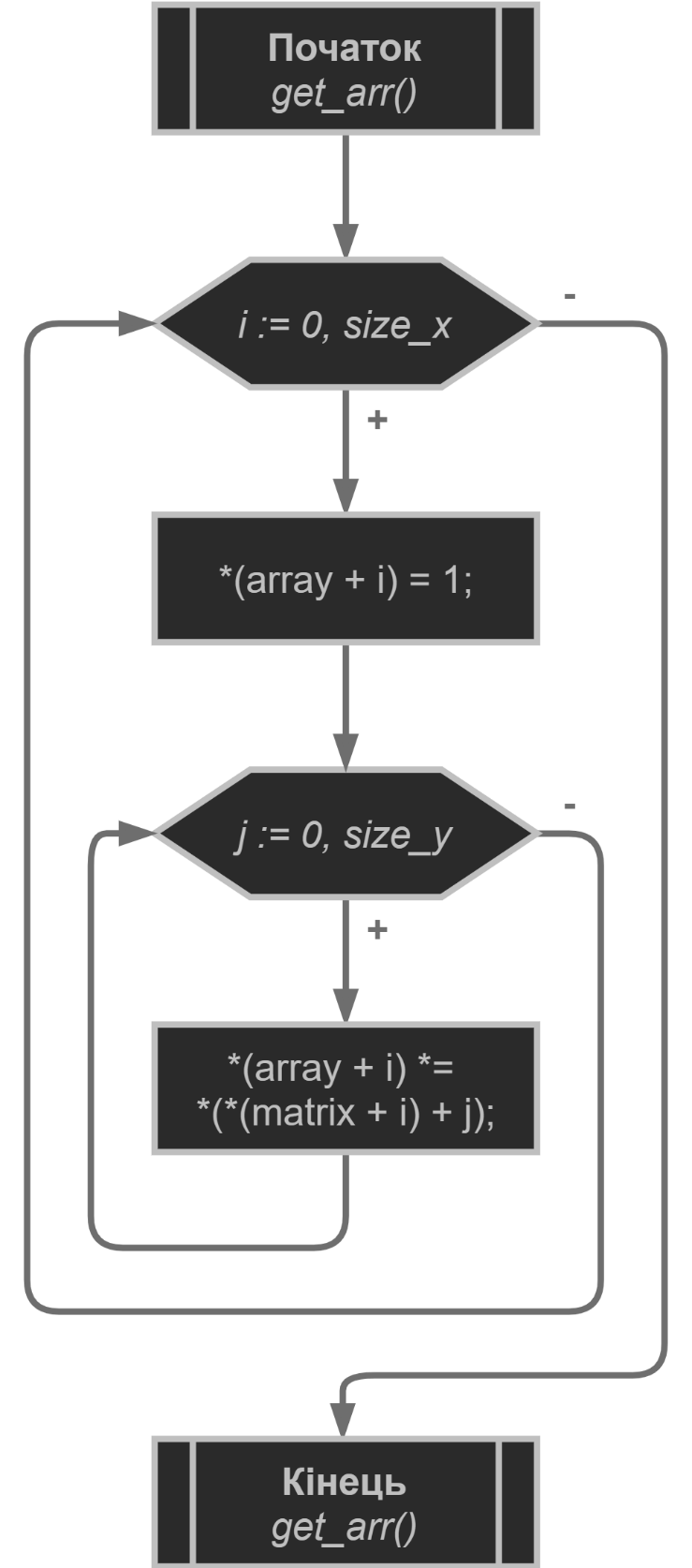
**Підпрограма rand\_double()**

****

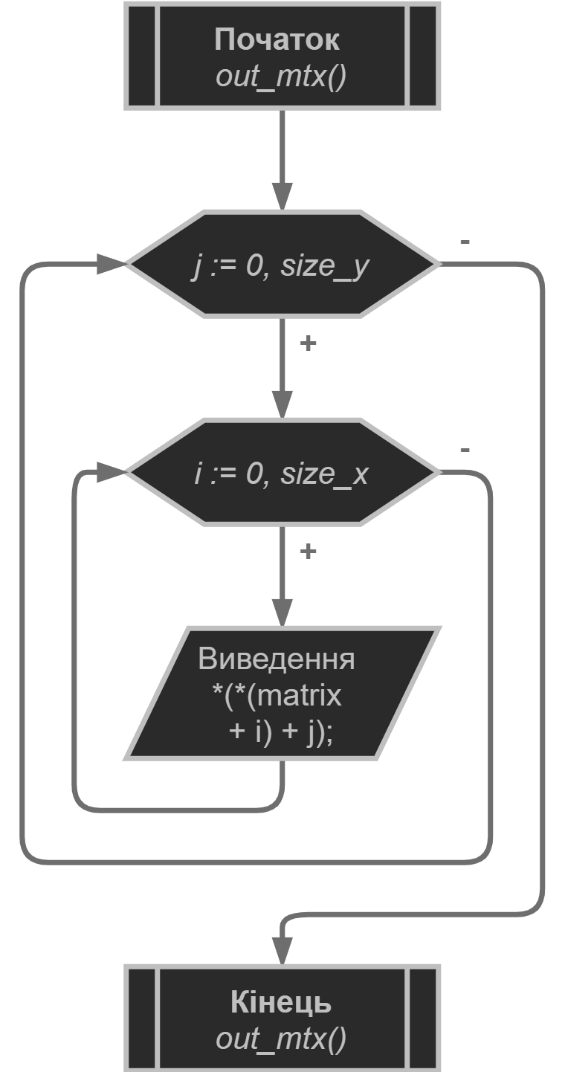
**Підпрограма gen\_mtx()**

****

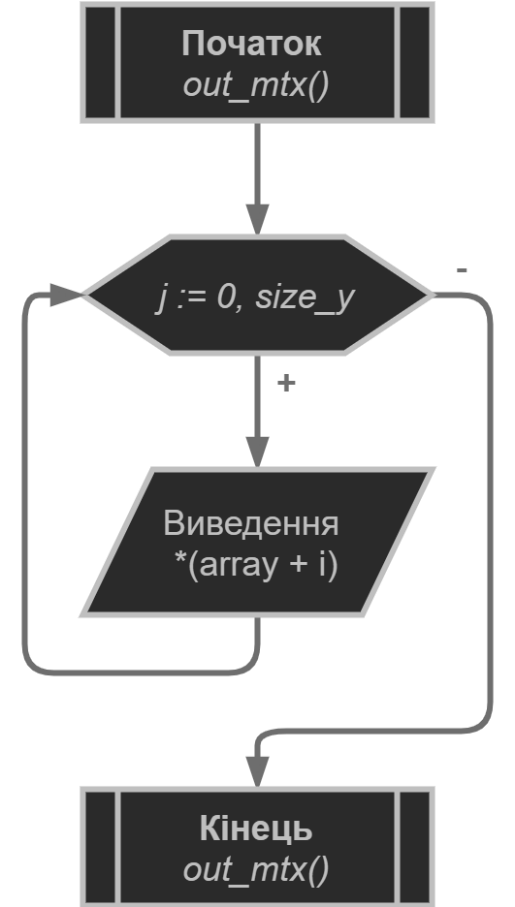
**Підпрограма get\_arr()**

****

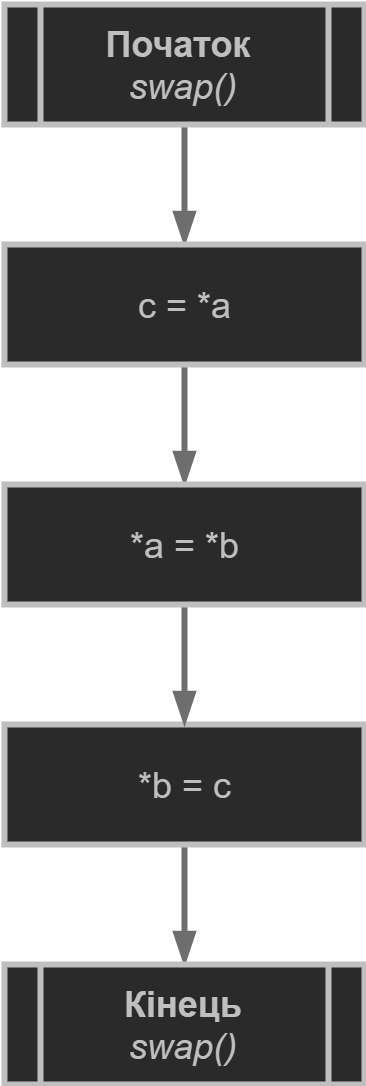
**Підпрограма out\_mtx()**

****

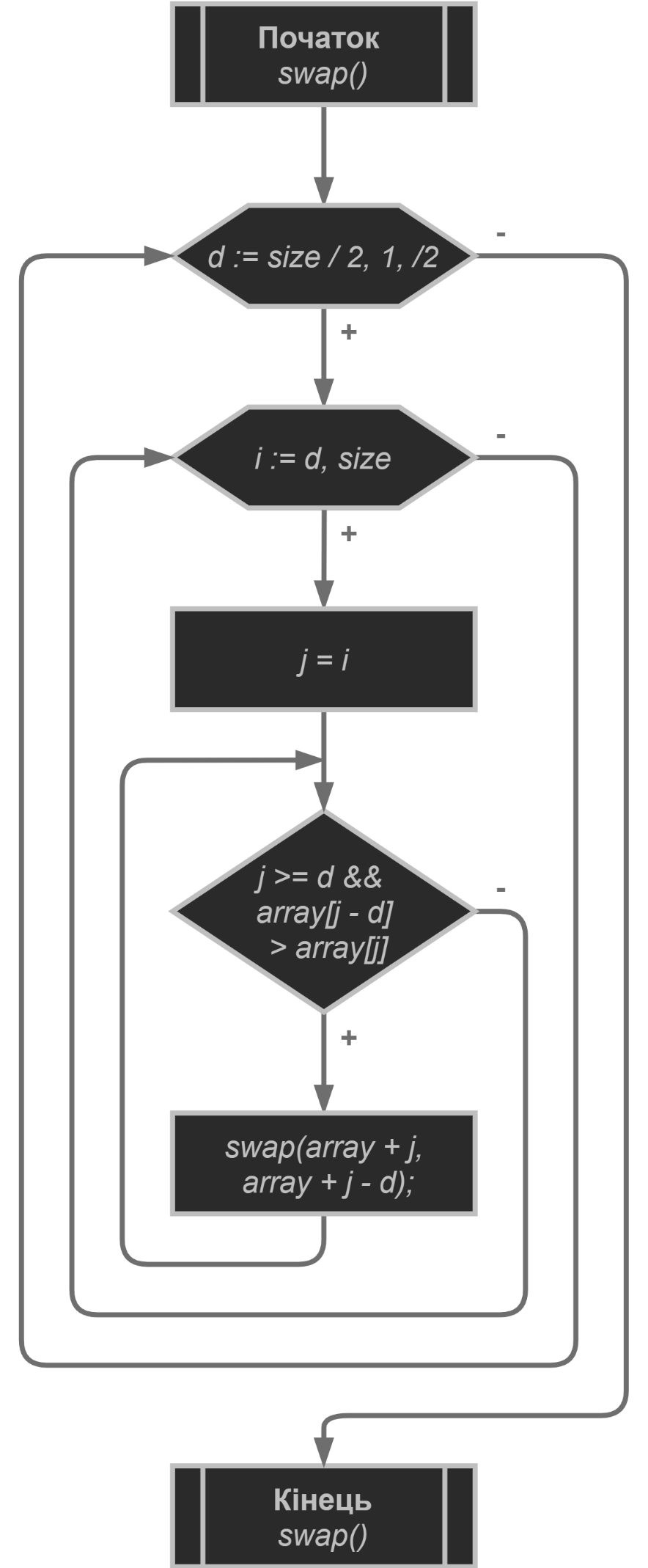
**Підпрограма out\_arr()**

****

**Підпрограма swap()**

****

**Підпрограма sort\_shell()**

****

**Код на С++**

#include <iostream>

#include <stdlib.h>

#include <math.h>

#include <cmath>

#include <iomanip>

#include <stdio.h>

using namespace std;

double\*\* declr\_mtx(int, int);

double\* declr\_arr(int);

void gen\_mtx(double\*\*, int, int, int, int);

void get\_arr(double\*\*, double\*, int, int);

void out\_mtx(double\*\*, int, int, int, int);

void out\_arr(double\*, int, int, int);

void swap(double\*, double\*);

void sort\_shell(double\*, int);

int main() {

int size\_x = 8, size\_y = 4;

double\*\* matrix = declr\_mtx(size\_x, size\_y);

double\* array = declr\_arr(size\_x);

int range, precision;

cout << "Enter range (<=16383): "; cin >> range;

cout << "Enter number of significant figures after decimal point (<=4): "; cin >> precision;

srand((unsigned int)time(NULL));

gen\_mtx(matrix, size\_x, size\_y, range, precision);

get\_arr(matrix, array, size\_x, size\_y);

cout << "\nGenerated matrix is: \n";

out\_mtx(matrix, size\_x, size\_y, range, precision);

cout << "Resulting array is: \n";

out\_arr(array, size\_x, range, precision);

sort\_shell(array, size\_x);

cout << "Resulting sorted array is: \n";

out\_arr(array, size\_x, range, precision);

system("pause");

return 0;

}

double\*\* declr\_mtx(int size\_x, int size\_y) {

double\*\* matrix = new double\* [size\_x];

for (int i = 0; i < size\_x; i++) {

\*(matrix + i) = new double [size\_y];

}

return matrix;

}

double\* declr\_arr(int size\_x) {

double\* array = new double[size\_x];

return array;

}

double rand\_double(int range, int precision) {

double random;

int rand\_int, rand\_frac;

rand\_int = rand() % (2 \* range) - range;

rand\_frac = rand() % int(pow(10, precision));

random = rand\_int + rand\_frac \* pow(0.1, precision);

return random;

}

void gen\_mtx(double\*\* matrix, int size\_x, int size\_y, int range, int precision) {

for (int j = 0; j < size\_y; j++)

for (int i = 0; i < size\_x; i++)

\*(\*(matrix + i) + j) = rand\_double(range, precision);

}

void get\_arr(double\*\* matrix, double\* array, int size\_x, int size\_y) {

for (int i = 0; i < size\_x; i++) {

\*(array + i) = 1;

for (int j = 0; j < size\_y; j++)

\*(array + i) \*= \*(\*(matrix + i) + j);

}

}

void out\_mtx(double\*\* matrix, int size\_x, int size\_y, int range, int precision) {

int width = (int)log10(range) + precision + 4;

int width\_full = width \* size\_x + 3;

cout << "\n";

for (int j = 0; j < size\_y; j++) {

cout << "|";

for (int i = 0; i < size\_x; i++)

cout << setw(width) << setprecision(precision) << fixed

<< \*(\*(matrix + i) + j);

cout << " |\n";

if (j != size\_y - 1)

cout << "|"

<< setw(width\_full)

<< "|\n";

}

cout << "\n";

}

void out\_arr(double\* array, int size\_x, int range, int precision) {

cout << "\n[ ";

for (int i = 0; i < size\_x; i++)

cout << setprecision((long long)4 \* precision)

<< \*(array + i) << " ";

cout << "]\n\n";

}

void swap(double\* a, double\* b) {

double c = \*a;

\*a = \*b;

\*b = c;

}

void sort\_shell(double\* array, int size) {

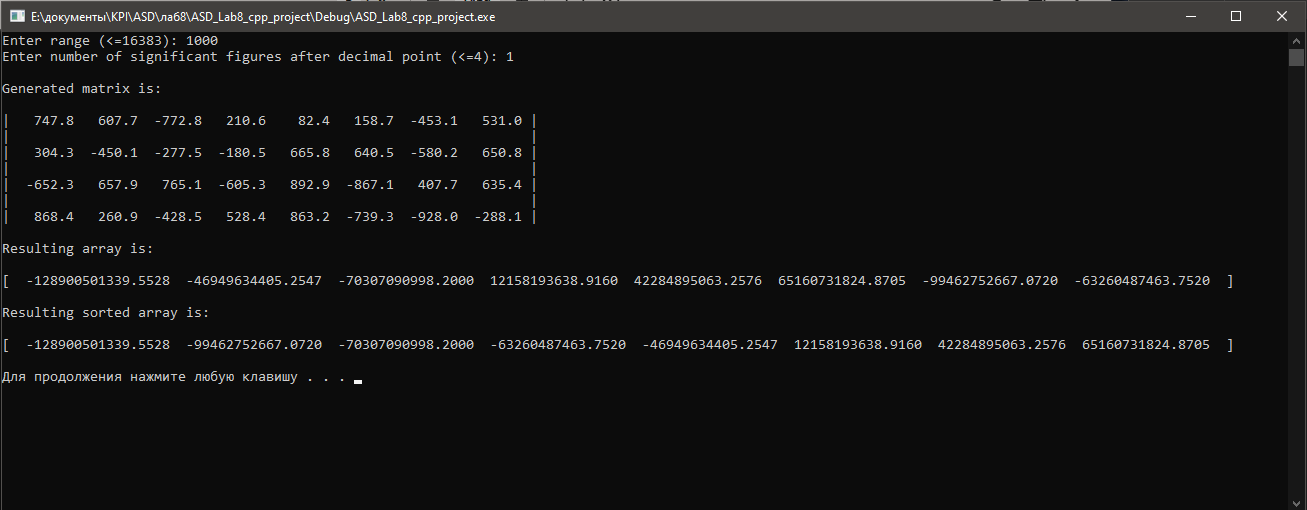
for (int d = size / 2; d >= 1; d /= 2)

for (int i = d; i < size; i++)

for (int j = i; j >= d && array[j - d] > array[j]; j -= d)

swap(array + j, array + j - d);

}

****

**Перевірка**

|  |  |
| --- | --- |
| Блок | Дія |
| 0 | Початок |
| 1 | Згенерована матриця :1 1 -1 0 1 0 -1 -1  1 -2 1 -1 -2 -2 -2 -2  0 -2 1 1 0 -1 -1 1  0 -2 0 -2 -1 -1 0 -1 |
| 2 | Обчислений масив: [ 0 -8 -0 0 0 -0 -0 -2 ] (масив обчислено правильно) |
| 3 | Сортований масив: [ -8 -2 0 -0 0 -0 -0 0 ] (масив відсортовано правильно) |
| 4 | Кінець |

**Висновок**

Отже, ми використали функцію sort\_shell для сортування масиву даних за зростанням, дослідили алгоритми пошуку та набули практичних

навичок використання цих алгоритмів під час складання програмних

специфікацій.