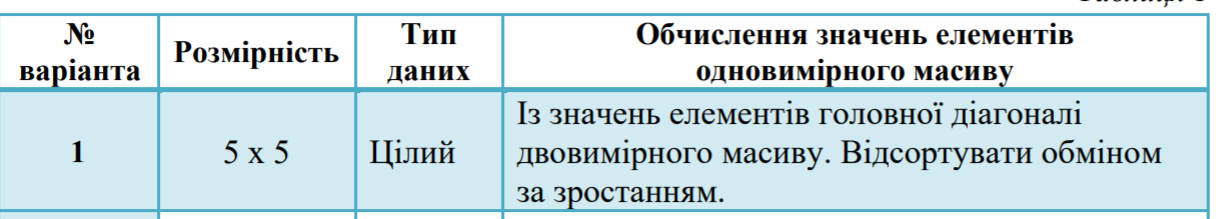
**Лабораторна робота 8**

**Дослідження алгоритмів пошуку та сортування**

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

**Постановка задачі**



Для сортування одновимірного масиву було використано сортування бульбашкою

**Побудова математичної моделі**

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім'я | Призначення |
| Массив | Цілий | a | Матриця чисел |
| Массив | Цілий | a2 | масив головної діагоналі |
| Допоміжна змінна | Цілий | k | Змінна для бульбашково сортування |
| Лічільники | Цілий | I,j | Лічільники |
| Функція | Цілий | iniz | Функція ініціалізації одновимірного масиву |
| Функція | Цілий | result | Бульбашкове сортування |
| Функція | Цілий | random | Функція генерування матриці |
|  |  |  |  |

Крок 1 Функція Ініціалізації матриці

Крок 2 Функція створення масиву на основі головної діагоналі матриці

Крок 3 Функція бульбашкового сортування

Random(a[5][5])

Початок

Для i Від 0 до 5 з кроком 1

Для j Від 0 до 5 з кроком 1

a[i][j] = rand() % 20 + 5;

Все повторити

Все повторити

Кінець

**Псевдокод**

Крок 1

Початок

Random(a)

Функція створення масиву на основі головної діагоналі матриці

Функція бульбашкового сортування

Кінець

Крок 2

iniz(a[5][5],b[5])

Початок

Для i Від 0 до 5 з кроком 1

b[i]=a[i][i]

Все повторити

Кінець

Початок

Random(a)

Iniz(a,a2)

Функція для бульбашкового сортування

Кінець

Крок 3

Result(a[5])

Початок

Для i Від 0 до 4 з кроком 1

Для j Від 0 до 4-i з кроком 1

K=a[j]

A[j]=a[j+1]

A[j+1]=k

Все повторити

Все повторити

Кінець

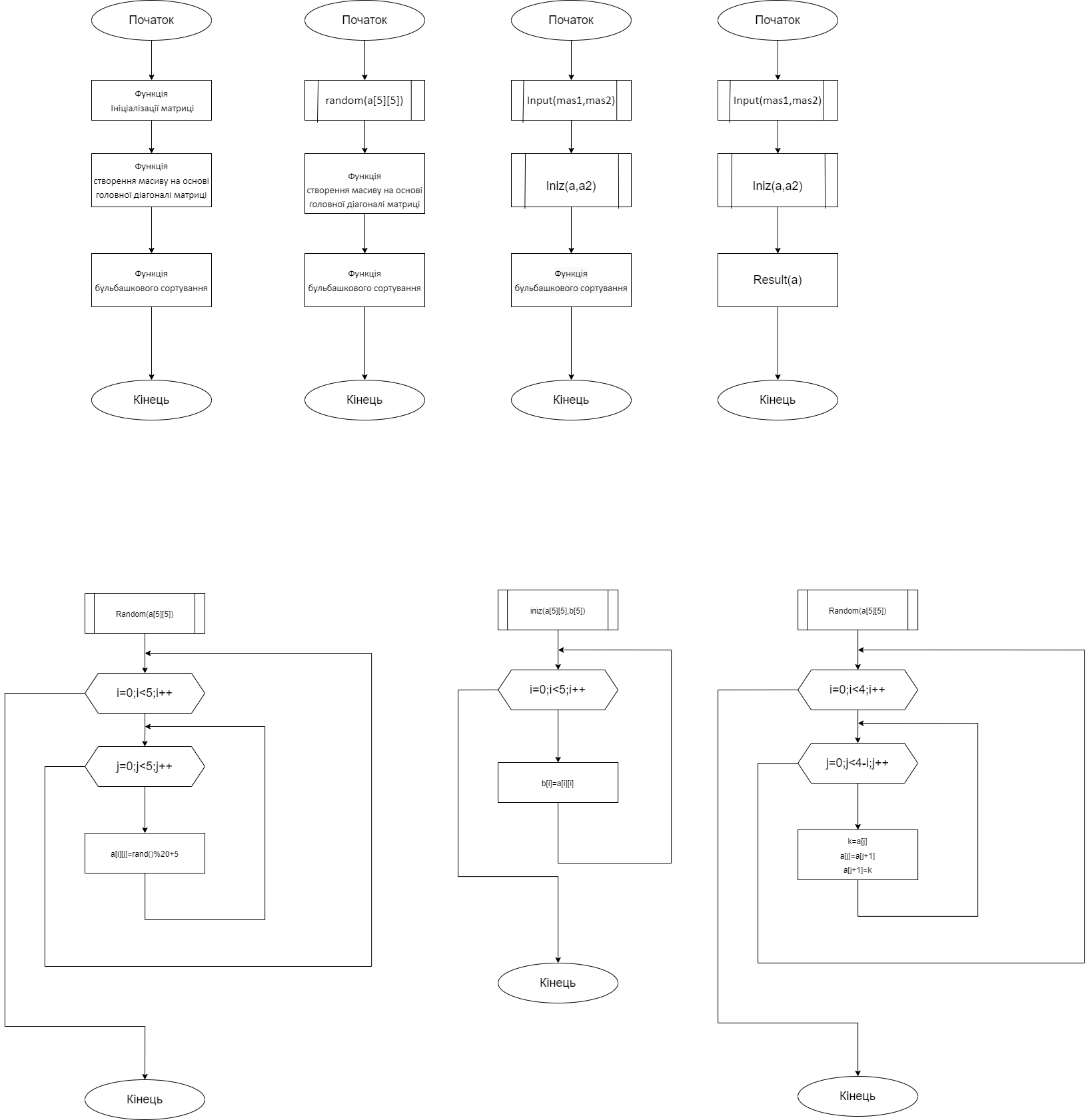
Початок

Random(a)

Iniz(a,a2)

Result(a)

Кінець

**Блок**-**схема**

**Код програми**

#include <iomanip>

#include <iostream>

using namespace std;

void random(int a[5][5]);

void iniz(int a[5][5], int b[5]);

void result(int a[5]);

void output(int a[5]);

int main()

{

srand(time(NULL));

int a[5][5];

int a2[5];

random(a);

iniz(a, a2);

output(a2);

result(a2);

output(a2);

}

void random(int a[5][5]) {

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

a[i][j] = rand() % 20 + 5;

cout << setw(3) << a[i][j];

}

cout << endl;

}

cout << endl;

}

void iniz(int a[5][5], int b[5]) {

for (int i = 0; i < 5; i++)

b[i] = a[i][i];

}

void result(int a[5]) {

for(int i=0;i<4;i++)

for(int j=0;j<4-i;j++)

if (a[j] > a[j + 1]) {

int k = a[j];

a[j] = a[j + 1];

a[j + 1] = k;

}

}

void output(int a[5]){

for (int i = 0; i < 5; i++) {

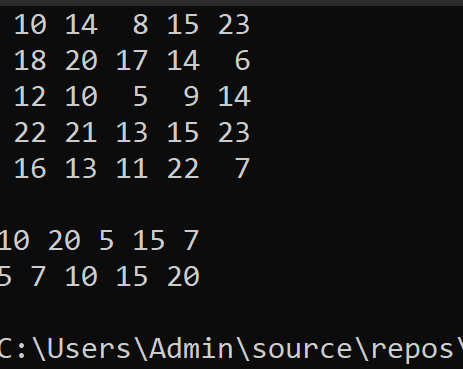
cout << a[i] << " ";

}

cout << endl;

}

**Випробування**



**Висновок**

У цій роботі було досліджено алгоритми пошуку та сортування, набуто практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Специфікою завдання було створення окремого масиву створеного з головної діагоналі, та сортування методом обміну.