

# Основи програмування – 1. Алгоритми та структури даних

---

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 13

Виконав студент

ІП-15 Конденко Іван Ігорович

(шифр, прізвище, ім'я, по батькові)

Перевірів(-ла)

Вечерковська Анастасія Сергіївна

(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.

#### Індивідуальне завдання

#### Варіант 13

#### Постановка задачі

#### Завдання

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).

13	6 x 6	Цілий	Із додатних значень елементів головної діагоналі двовимірного масиву. Відсортувати методом бульбашки за зростанням.
----	-------	-------	---

#### Математична модель

Змінна	Тип	Ім'я	Призначення
Двовимірний масив	Цілочисельний[6][6]	mass1	Вхідні дані
Одновимірний масив	Цілочисельний[6]	mass2	Вхідні дані
Розмірність масиву	Цілочисельний (константа)	roz	Вхідні дані
Ф-ія для побудови масивів	Пустий	bud_mass(mass1[],mass2[])	Проміжні дані
Ф-ія для сортування масиву	Пустий	Sort(mass2[],n)	Проміжні дані
Ітератор	Цілочисельний	i	Проміжні дані
Ітератор	Цілочисельний	j	Проміжні дані

Двовимірний масив будується за допомогою ф-ії bud\_mass в якій знаходиться цикл, який заповнює масив випадковими числами від 0 до 100. rand – заповнення випадковими числами. Одновимірний масив будується за

## Основи програмування – 1. Алгоритми та структури даних

---

допомогою циклу в якому порівнюються індекси двовимірного масиву і якщо вони співпадають, то цей елемент додається до одновимірного масиву. У ф-ії виконується сортування одновимірного масиву методом бульбашки. Його сутність полягає в тому, що по черзі йде порівняння двох сусідніх елементів. Якщо вони не впорядковані між собою, вони змінюють положення. Обмін повторюється стільки ж разів скільки елементів у масиві. Сама ж функція приймає масив та його розмірність.

### Розв'язання

#### Псевдокод

#### Початок

**Введення** mass1, mass2, roz

bud\_mass(mass1, mass2, n)

Sort(mass2, roz)

#### Кінець

#### Функція bud\_mass(char mass1[6], char mass2[6], n)

**повторити** для і від 0 до  $i < n$

**повторити** для j від 0 до  $j < n$

mass1[i][j] = rand

**все повторити**

**все повторити**

**повторити** для і від 0 до  $i < n$

**повторити** для j від 0 до  $j < n$

mass2[i] = mass1[i=j][j=i]

**все повторити**

**все повторити**

#### Функція Sort(mass2[6], n)

**повторити** для і від 0 до  $i < n$

**повторити** для j від 0 до  $j < n$

**якщо** mass2[j+1] < mass2[j] **то**

tmp = mass2[j+1]

mass2[j+1] = mass2[j]

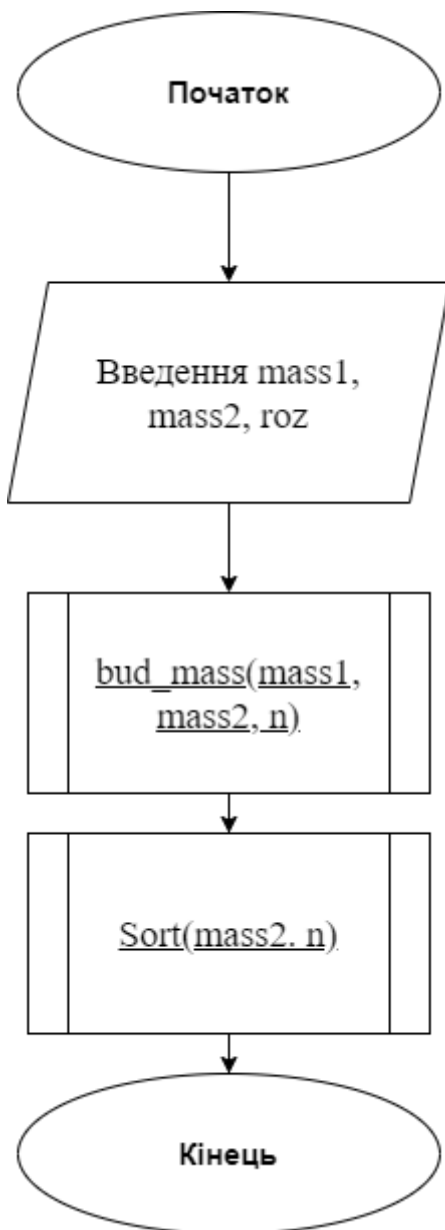
mass2[j] = tmp

**все повторити**

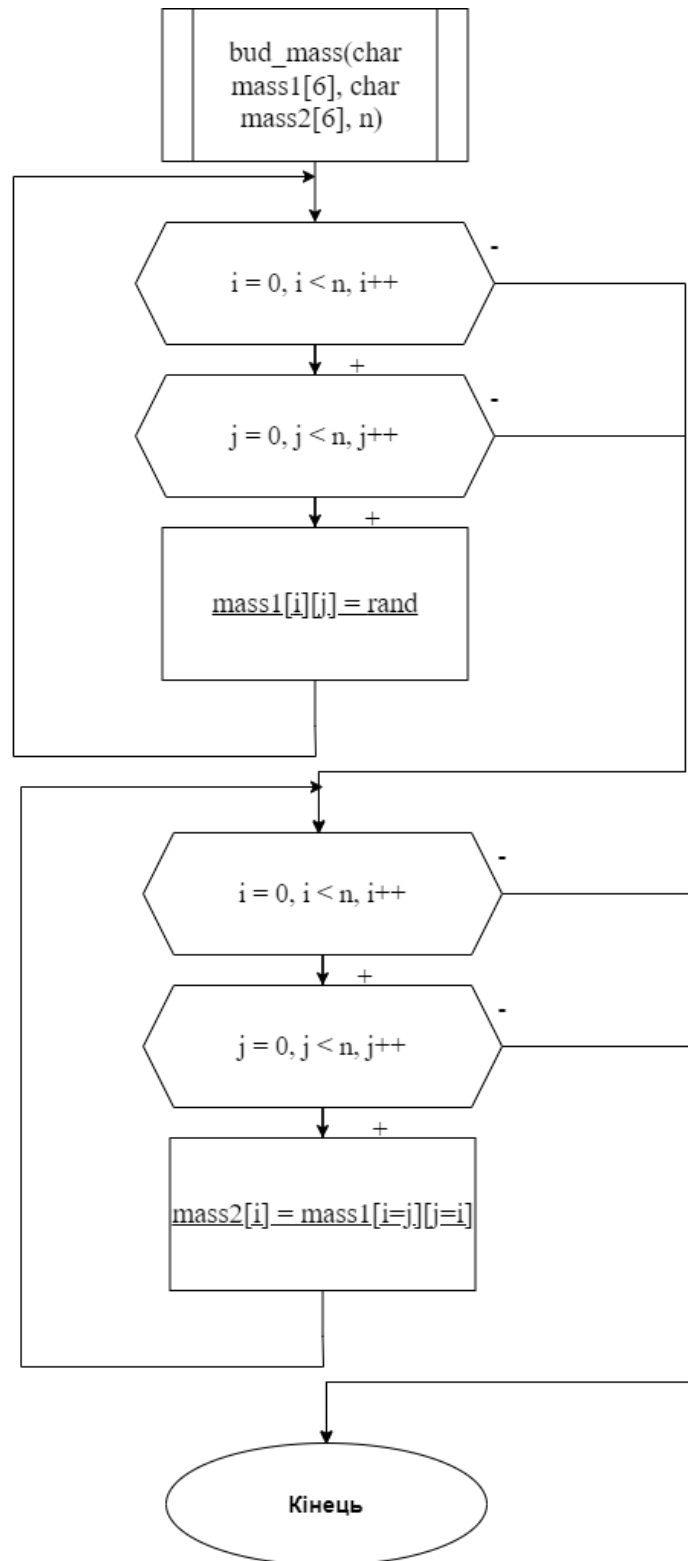
**все повторити**

Блок схема

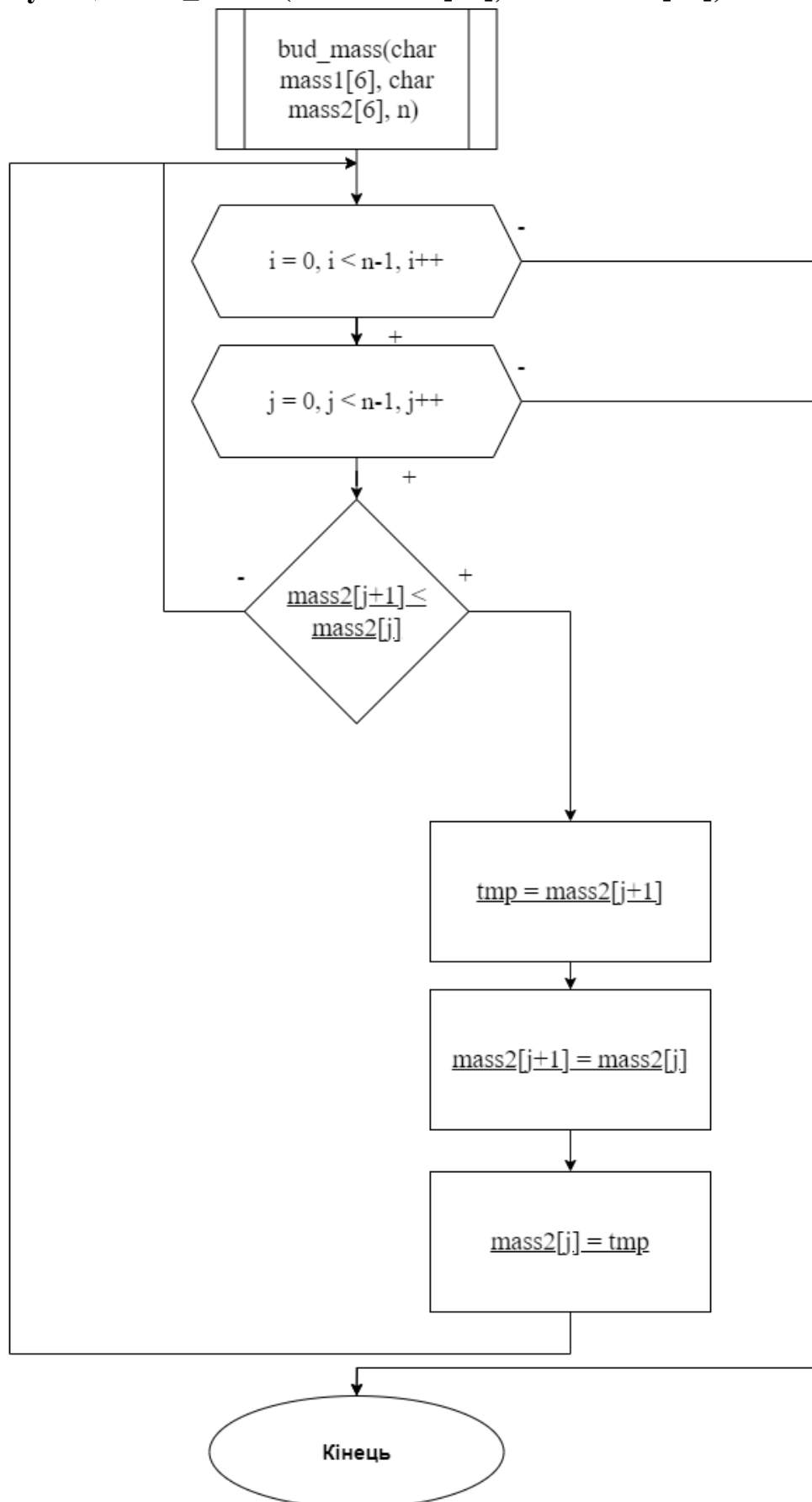
Основна програма



Функція bud\_mass(char mass1[6], char mass2[6], n)



Функція `bud_mass3(char mass1[10], char mass2[10], char mass3[10])`



## Код програми

```
#include <iostream>

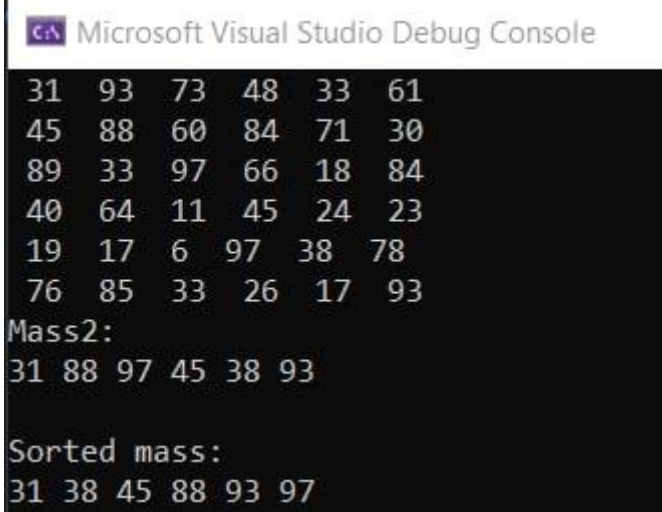
using namespace std;

void bud_mass(int mass1[6][6], int mass2[6], int n) {
    srand((int)time(0));
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            mass1[i][j] = rand() % 100;
            cout << " " << mass1[i][j] << " ";
        }
        cout << endl;
    }
    cout << "Mass2:" << endl;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++) {
            mass2[i] = mass1[i = j][j = i];
            cout << mass2[i] << " ";
        }
    cout << endl;
}

void Sort(int mass2[6], int n)
{
    for (int i = 0; i < n-1; ++i)
    {
        for (int j = 0; j < n-1; ++j)
        {
            if (mass2[j+1] < mass2[j])
            {
                int tmp = mass2[j+1];
                mass2[j+1] = mass2[j];
                mass2[j] = tmp;
            }
        }
    }
    cout << "\nSorted mass:\n";
    for (int k = 0; k < n; ++k) {
        cout << mass2[k] << " ";
    }
}

int main()
{
    int const roz = 6;
    int mass1[roz][roz];
    int mass2[roz];
    bud_mass(mass1, mass2, roz);
    Sort(mass2, roz);
}
```

## Випробування програми



```
Microsoft Visual Studio Debug Console

31 93 73 48 33 61
45 88 60 84 71 30
89 33 97 66 18 84
40 64 11 45 24 23
19 17 6 97 38 78
76 85 33 26 17 93
Mass2:
31 88 97 45 38 93
Sorted mass:
31 38 45 88 93 97
```

## Висновки

Ми дослідили алгоритми пошуку та сортування, набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій. У результаті ми отримали програму, яка знаходить елементи головної матриці двохвимірного масиву, переносить їх в одновимірний масив та сортує його методом бульбашки.