Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

Кафедра ІПЗ

**Лабораторна робота № 6**

**«Методи сортування. Сортування в масивах**

**Основні алгоритми реалізації »**

Виконав

ст. гр. ІП-22-1

Курчій С.В.

Перевірив

доцент Процюк В.Р.

Івано-Франківськ

2023

**Мета:** ознайомлення з основними методами сортування, а також дослідження алгоритмів сортування та вивчення способів їх реалізації за допомогою мови програмування С++

**Умова:**

1.1 Побудувати у середовищі Draw.io таблично-графічну схему роботи методу сортування бульбашкою за зростанням).

1.2 Побудувати у середовищі таблично-графічну схему сортування за зростанням згідно з методом вставок (представлення початку виконання + етап завершення сортування).

2. Заповнити одновимірний цілочисельний масив А(його розмірність=30) з допомогою генератора випадкових чисел (діапазон від 10 до 100). Підготувати окремі відповідні програми реалізації вказаних методів для сортування за зростанням отриманого масиву А ( Наприклад, мовою C++).

**Розв’язок**

**Завдання 1.**

#include <iostream>

void printArray(int arr[], int size)

{

    for (int i = 0; i < size; i++)

    {

        std::cout << arr[i] << " ";

    }

    std::cout << std::endl;

}

void bubbleSort(int arr[], int size)

{ // Зовнішній цикл проходить через усі елементи масиву.

    for (int i = 0; i < size - 1; i++)

    {

        for (int j = 0; j < size - i - 1; j++)

        { // Порівняння двох сусідніх елементів.

            if (arr[j] > arr[j + 1])

            {

                // Якщо поточний елемент більший за наступний, то вони обмінюються місцями.        int temp = arr[j];

                arr[j] = arr[j + 1];

                arr[j + 1] = temp;

            }

        }

    }

}

void insertionSort(int arr[], int size)

{

    for (int i = 1; i < size; i++)

    {

        int key = arr[i]; // Збереження значення поточного елемента масиву

        int j = i - 1; // Вказує на позицію перед поточним лементом

        // Перемістити елементи arr[0..i-1], які більші за ключ

// на одну позицію попереду поточної позиції

while (j >= 0 && arr[j] > key)

        {

            arr[j + 1] = arr[j]; // Зрушення елемента вправо

            j = j - 1;

        }

        arr[j + 1] = key; // Key Вставляється на відповідне місце

    }

}

int main()

{

    // Створення масиву

    int arr[] = {80, 20, 60, 10, 50, 30};

    int size = sizeof(arr) / sizeof(arr[0]);

    // Виведення початкового масиву

    std::cout << "Initial array: ";

    printArray(arr, size);

    // Сортування бульбашкою

    bubbleSort(arr, size);

    std::cout << "Array after bubble sort: ";

    printArray(arr, size);

    // Створення нового масиву для сортування вставками

    int arrInsertion[] = {80, 20, 60, 10, 50, 30};

    // Виведення початкового масиву для сортування вставками

    std::cout << "Initial array for insertion sort: ";

    printArray(arrInsertion, size);

    // Сортування вставками

    insertionSort(arrInsertion, size);

    std::cout << "Array after insertion sort: ";

    printArray(arrInsertion, size);

    return 0;

}

**Завдання 2.**

#include <iostream>

#include <cstdlib>

#include <ctime>

const int ARRAY\_SIZE = 30;

void fillArray(int arr[], int size)

{

    for (int i = 0; i < size; ++i)

    {

        arr[i] = rand() % 91 + 10; // Генерує випадкове число в діапазоні [10, 100]

    }

}

void printArray(const int arr[], int size)

{

    for (int i = 0; i < size; ++i)

    {

        std::cout << arr[i] << " ";

    }

    std::cout << std::endl;

}

void insertionSort(int arr[], int size)

{

    for (int i = 1; i < size; ++i)

    {

        int key = arr[i]; // Збереження значення поточного елементу масиву

        int j = i - 1; // Вказує на позицію перед поточним елементом

        while (j >= 0 && arr[j] > key)

        {

            arr[j + 1] = arr[j]; // Зрушення елемента вправо

            --j;

        }

        arr[j + 1] = key; // Key Вставляється на відповідне місце

    }

}

int main()

{

    // Ініціалізація генератора випадкових чисел

    srand(static\_cast<unsigned>(time(0)));

    int A[ARRAY\_SIZE];

    // Заповнення та вивід початкового масиву

    std::cout << "Initial array A:\n";

    fillArray(A, ARRAY\_SIZE);

    printArray(A, ARRAY\_SIZE);

    // Сортування та вивід відсортованого масиву

    insertionSort(A, ARRAY\_SIZE);

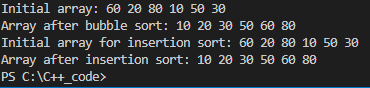
    std::cout << "Array A after ascending sorting:\n";

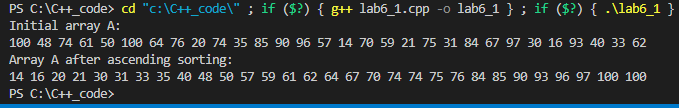
    printArray(A, ARRAY\_SIZE);

    return 0;

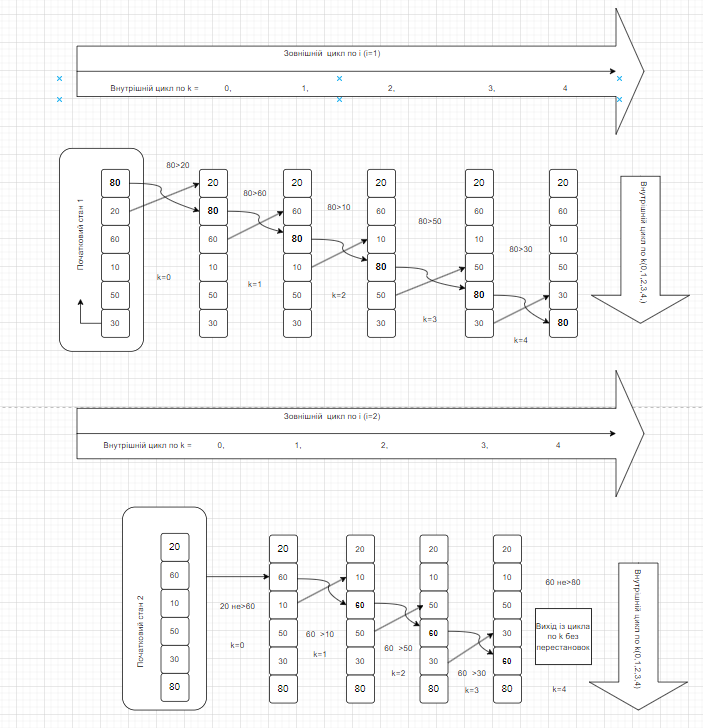
}

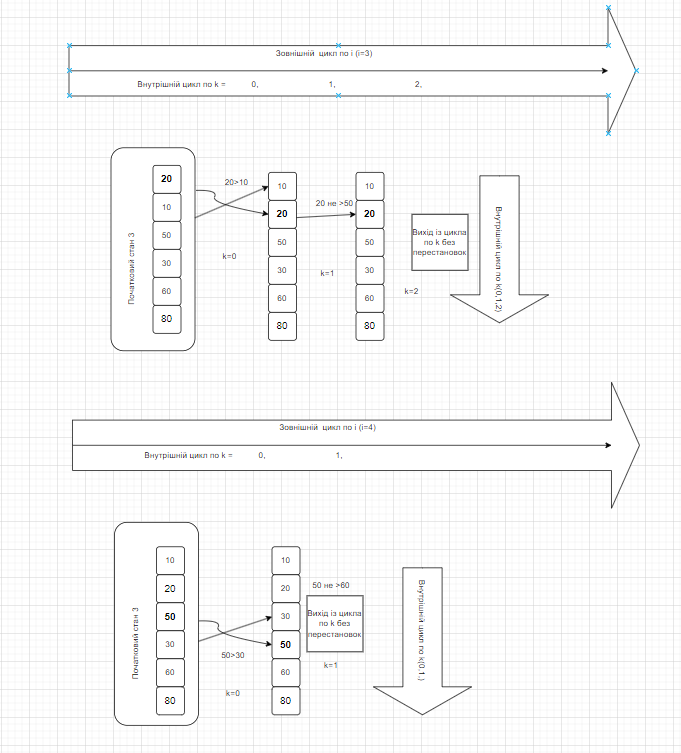
**Результат:**

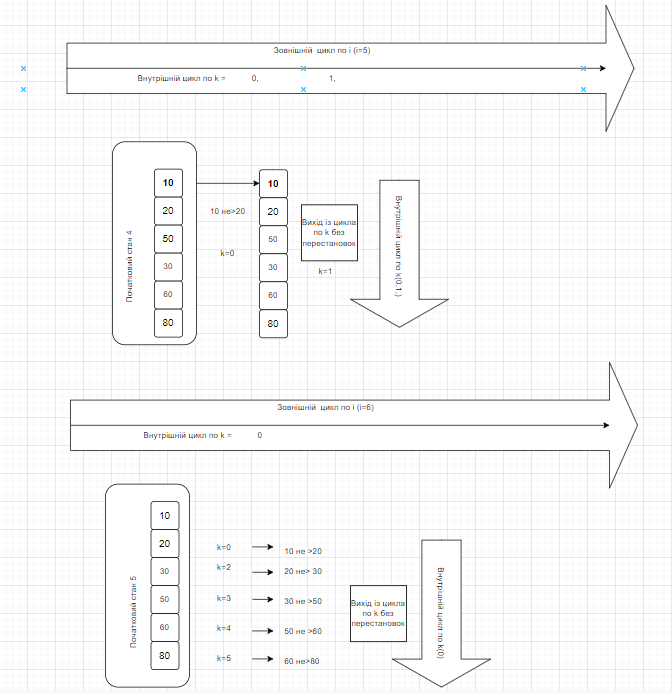




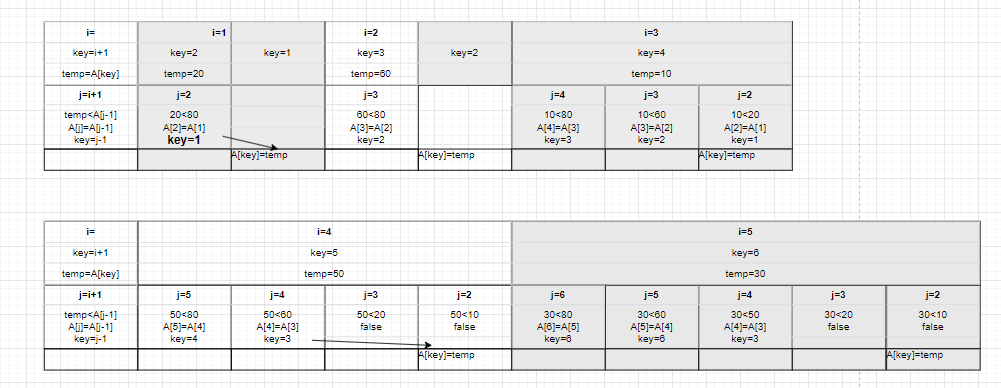
**Сотування бульбашкою:**

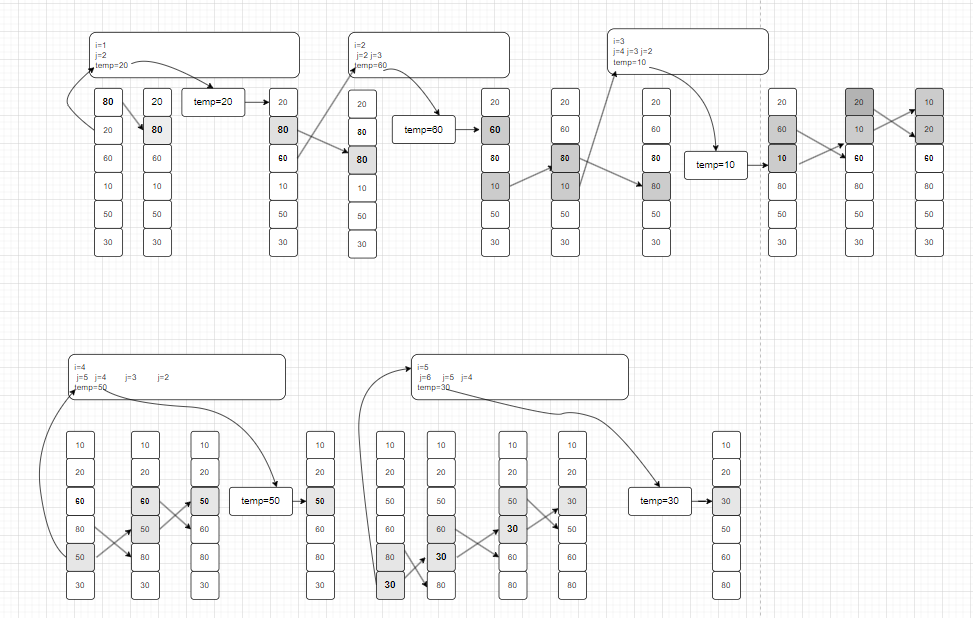






**Сортування вставками:**

****

****

**Висновок:** Вивчення основних методів сортування та їх реалізації за допомогою мови програмування C++ є ключовим етапом для розуміння та вдосконалення алгоритмічних навичок. Основні методи сортування включають в себе алгоритми, такі як сортування вставками, сортування бульбашкою.