// You are using GCC

#include <iostream>

#include <algorithm> // For std::swap

using namespace std;

// Function to perform bubble sort on the array and count swaps

void bubbleSort(int arr[], int n) {

    int result=0;

        bool swapped = false;

    for (int i = 0; i < n; i++) { // Adjust outer loop to n - 1

        int count = 0; // Reset swap count for this pass

        for (int j = 0; j < n-i-1; j++) { // Adjust inner loop to n - i - 1

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

                swap(arr[j], arr[j + 1]); // Swap elements

                count++;

                result++;// Increment swap counter

                swapped = true; // Mark that a swap occurred

            }

        }

        // Print the array after each pass of the outer loop

        cout << "Array after pass " << i + 1 << ": ";

        for (int k = 0; k < n; k++) {

            cout << arr[k] << " ";

        }

        cout << "| No of swaps done : " << count << endl;

         if (!swapped) break;

        //result = result + count;

    }

        cout<<result<<endl;;

}

int main() {

    int n;

    cin >> n; // Read the number of elements

    int arr[n]; // Create an array of size n

    // Read array elements

    for (int i = 0; i < n; i++) {

        cin >> arr[i];

    }

    bubbleSort(arr, n); // Call bubble sort to sort the array

    // Output the final sorted array

    cout << "Final sorted array: ";

    for (int i = 0; i < n; i++) {

        cout << arr[i] << " ";

    }

    cout << endl;

    return 0;

}

// You are using GCC

#include<iostream>

using namespace std;

void insertionSort(int arr[], int n){

    for(int i=0; i<n; i++){

        for(int j=i+1; j>0; j--){

            if(arr[j]<arr[j-1]){

              swap(arr[j], arr[j-1]);

            }else{

                break;

            }

        }

    }

    }

int main(){

    int n;

    cin>>n;

    int arr[n];

    for(int i=0; i<n; i++){

        cin>>arr[i];

    }

    insertionSort(arr,n);

    for (int i = 0; i < n; i++) {

        cout << arr[i] << " ";

    }

}

#include <iostream>

using namespace std;

// Function to perform insertion sort on the array

void insertionSort(int arr[], int n) {

    for (int i = 1; i < n; i++) {  // Start from the second element

        int key = arr[i];           // Element to be inserted

        int j = i - 1;

        // Move elements of arr[0..i-1] that are greater than key

        // to one position ahead of their current position

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

            arr[j + 1] = arr[j];

            j--;

        }

        arr[j + 1] = key;  // Insert the key into the correct position

    }

}

int main() {

    int n;

    cin >> n; // Read the number of elements

    int arr[n]; // Create an array of size n

    // Read array elements

    for (int i = 0; i < n; i++) {

        cin >> arr[i];

    }

    insertionSort(arr, n); // Call insertion sort to sort the array

    // Output the final sorted array

    for (int i = 0; i < n; i++) {

        cout << arr[i] << " ";

    }

    cout << endl;

    return 0;

}

#include <iostream>

using namespace std;

// Function to perform selection sort on the array

void selectionSort(int arr[], int n) {

    // Traverse through all array elements

    for (int i = 0; i < n - 1; i++) {

        // Assume the minimum is the first element of the unsorted portion

        int minIndex = i;

        // Find the index of the minimum element in the unsorted portion

        for (int j = i + 1; j < n; j++) {

            if (arr[j] < arr[minIndex]) {

                minIndex = j;

            }

        }

        // Swap the found minimum element with the first element of the unsorted portion

        if (minIndex != i) {

            swap(arr[i], arr[minIndex]);

        }

        // Print the array after each pass of the outer loop

        cout << "Array after pass " << i + 1 << ": ";

        for (int k = 0; k < n; k++) {

            cout << arr[k] << " ";

        }

        cout << endl;

    }

}

int main() {

    int n;

    cin >> n; // Read the number of elements

    int arr[n]; // Create an array of size n

    // Read array elements

    for (int i = 0; i < n; i++) {

        cin >> arr[i];

    }

    // Perform selection sort and print array after each step

    selectionSort(arr, n);

    // Output the final sorted array

    cout << "Final sorted array: ";

    for (int i = 0; i < n; i++) {

        cout << arr[i] << " ";

    }

    cout << endl;

    return 0;

}

#include <iostream>

using namespace std;

// Function to perform selection sort on the array

void selectionSort(int arr[], int n) {

    // Traverse through all array elements

    for (int i = 0; i < n - 1; i++) {

        // Assume the minimum is the first element of the unsorted portion

        int minIndex = i;

        bool swapped = false; // Flag to check if any swap occurs

        // Find the index of the minimum element in the unsorted portion

        for (int j = i + 1; j < n; j++) {

            if (arr[j] < arr[minIndex]) {

                minIndex = j;

            }

        }

        // Swap the found minimum element with the first element of the unsorted portion

        if (minIndex != i) {

            swap(arr[i], arr[minIndex]);

            swapped = true; // A swap occurred

        }

        // Print the array after each pass of the outer loop

        cout << "Array after pass " << i + 1 << ": ";

        for (int k = 0; k < n; k++) {

            cout << arr[k] << " ";

        }

        cout << endl;

        // If no swaps occurred, the array is already sorted

        if (!swapped) {

            cout << "Array is already sorted after pass " << i + 1 << endl;

            break; // Exit early

        }

    }

}

int main() {

    int n;

    cin >> n; // Read the number of elements

    int arr[n]; // Create an array of size n

    // Read array elements

    for (int i = 0; i < n; i++) {

        cin >> arr[i];

    }

    // Perform selection sort and print array after each step

    selectionSort(arr, n);

    // Output the final sorted array

    cout << "Final sorted array: ";

    for (int i = 0; i < n; i++) {

        cout << arr[i] << " ";

    }

    cout << endl;

    return 0;

}