Insertion Sort

```
#include <iostream>
using namespace std;

void insertionSort(int arr[], int n) {
    for (int i = 1; i < n; i++) {
        int key = arr[i];
        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 = j - 1;
        }
        arr[j + 1] = key;

    }
}

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] << " ";
    }

    return 0;
}
```

Insertion Sort after every pass

```
// C++ Implementation
#include <iostream>
using namespace std;

void printArray(int arr[], int n, int pass) {
    cout << "Pass " << pass << ": ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

void insertionSort(int arr[], int n) {
    for (int i = 1; i < n; i++) {
        int key = arr[i];
        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 = j - 1;
        }
        arr[j + 1] = key;

        // Print array after each pass
        printArray(arr, n, i);
    }
}

int main() {
    int n;

   cin >> n;

    int arr[n];

   for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    insertionSort(arr, n);

    cout << "\nFinal Sorted array: ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }

    return 0;
}
```

Java
Insertion Sort

```
import java.util.Scanner;

class InsertionSort {
    public static void insertionSort(int[] arr) {
        int n = arr.length;
        for (int i = 1; i < n; i++) {
            int key = arr[i];
            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 = j - 1;
            }
            arr[j + 1] = key;

            // Print array after each pass
            printArray(arr, i);
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int n = scanner.nextInt();

        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }


        insertionSort(arr);

        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        scanner.close();
    }
}
```

Insertion sort after every pass

```
import java.util.Scanner;

class InsertionSort {
    public static void printArray(int[] arr, int pass) {
        System.out.print("Pass " + pass + ": ");
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }

    public static void insertionSort(int[] arr) {
        int n = arr.length;
        for (int i = 1; i < n; i++) {
            int key = arr[i];
            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 = j - 1;
            }
            arr[j + 1] = key;

            // Print array after each pass
            printArray(arr, i);
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int n = scanner.nextInt();

        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }


        insertionSort(arr);

        scanner.close();
    }
}
```