🧩1️⃣Insertion Sort (Short Version)

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

int main() {
    int arr[] = {5, 3, 4, 1, 2}, n = 5;
    for (int i = 1; i < n; i++) {
        int key = arr[i], j = i - 1;
        while (j >= 0 && arr[j] > key) arr[j + 1] = arr[j--];
        arr[j + 1] = key;
    }
    cout << "Sorted (Insertion): ";
    for (int i = 0; i < n; i++) cout << arr[i] << " ";
}
```

🧩2️⃣Selection Sort (Short Version)

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

int main() {
    int arr[] = {64, 25, 12, 22, 11}, n = 5;
    for (int i = 0; i < n - 1; i++) {
        int minIdx = i;
        for (int j = i + 1; j < n; j++)
            if (arr[j] < arr[minIdx]) minIdx = j;
        swap(arr[i], arr[minIdx]);
    }
    cout << "Sorted (Selection): ";
    for (int i = 0; i < n; i++) cout << arr[i] << " ";
}
```

🧩3️⃣Merge Sort (Short Version)

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

void merge(int a[], int l, int m, int r) {
    int i=l, j=m+1, k=0, temp[r-l+1];
    while(i<=m && j<=r) temp[k++] = (a[i]<a[j]) ? a[i++] : a[j++];
    while(i<=m) temp[k++] = a[i++];
    while(j<=r) temp[k++] = a[j++];
    for(i=l, k=0; i<=r; i++, k++) a[i] = temp[k];
}

void mergeSort(int a[], int l, int r) {
    if(l<r) {
        int m = (l+r)/2;
        mergeSort(a, l, m);
        mergeSort(a, m+1, r);
        merge(a, l, m, r);
    }
}
int main() {
    int arr[] = {38, 27, 43, 3, 9, 82, 10}, n = 7;
    mergeSort(arr, 0, n-1);
    cout << "Sorted (Merge): ";
    for(int i=0; i<n; i++) cout << arr[i] << " ";
}
```