

Quick Sort Code In C++

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

// Function to swap two elements
void swap(int &a, int &b) {
    int temp = a;
    a = b;
    b = temp;
}

// Partition function (Lomuto partition scheme)
int partition(int arr[], int low, int high) {
    int pivot = arr[high]; // Choosing last element as pivot
    int i = low - 1;       // Index of smaller element

    for (int j = low; j < high; j++) {
        if (arr[j] <= pivot) {
            i++;
            swap(arr[i], arr[j]); // Place smaller element on left
        }
    }
    swap(arr[i + 1], arr[high]); // Place pivot in correct position
    return i + 1;
}

// Quick Sort function (recursive)
void quickSort(int arr[], int low, int high) {
    if (low < high) {
        // Partition the array
        int pi = partition(arr, low, high);

        // Recursively sort elements before and after partition
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}
```

```
// Driver code
int main() {
    int arr[] = {10, 7, 8, 9, 1, 5};
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Original Array: ";
    for (int i = 0; i < n; i++) cout << arr[i] << " ";
    cout << endl;

    quickSort(arr, 0, n - 1);

    cout << "Sorted Array: ";
    for (int i = 0; i < n; i++) cout << arr[i] << " ";
    cout << endl;

    return 0;
}
```

Output-

Original Array: 10 7 8 9 1 5
Sorted Array: 1 5 7 8 9 10

www.tpcglobal.in