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

// Partition function to rearrange elements around the pivot
int partition(int arr[], int start, int end) {
    int pivot = arr[start];
    int count = 0;
    for (int i = start + 1; i <= end; i++) {
        if (arr[i] <= pivot)
            count++;
    }
    int pivotIndex = start + count;
    swap(arr[pivotIndex], arr[start]);
    int i = start, j = end;
    while (i < pivotIndex && j > pivotIndex) {
        while (arr[i] <= pivot) {
            i++;
        }
        while (arr[j] > pivot) {
            j--;
        }
        if (i < pivotIndex && j > pivotIndex) {
            swap(arr[i++], arr[j--]);
        }
    }
    return pivotIndex;
}

// Recursive QuickSort function
void quickSort(int arr[], int start, int end) {
    if (start >= end)
        return;
    int p = partition(arr, start, end);
    quickSort(arr, start, p - 1);
    quickSort(arr, p + 1, end);
}

int main() {
    //int arr[] = {9, 3, 4, 2, 1, 8};
    //int n = 6; // Dimension array.
    int arr[] = {9, 3, 5, 4, 2, 7, 8, 0, 6, 1};
    int n = 10;
    quickSort(arr, 0, n - 1);
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    return 0;
}
```