Quick Sort Algorithm

Quick Sort is a highly efficient sorting algorithm based on the divide-and-conquer approach. It works by selecting a 'pivot' element from the array and partitioning the other elements into two sub-arrays, according to whether they are less than or greater than the pivot. The sub-arrays are then sorted recursively.

Steps of Quick Sort:

1. **Choose a Pivot**: Select an element from the array as the pivot. Common choices include the first element, the last element, or a random element.
2. **Partitioning**: Rearrange the array so that all elements less than the pivot come before it, and all elements greater than the pivot come after it.
3. **Recursively Apply**: Apply the above steps recursively to the sub-arrays of elements with smaller and larger values.

Implementation in C

Here is a simple implementation of Quick Sort in C without using any user-defined functions:

```
#include <stdio.h>

void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pivot = arr[high]; // pivot
        int i = (low - 1); // Index of smaller element

        for (int j = low; j <= high - 1; j++) {
            if (arr[j] < pivot) {
                i++; // increment index of smaller element
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
        int temp = arr[i + 1];
        arr[i + 1] = arr[high];
        arr[high] = temp;
        int pi = i + 1;

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

int main() {
    int arr[] = {10, 7, 8, 9, 1, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    quickSort(arr, 0, n - 1);
    printf("Sorted array: \n");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

Explanation:

- **Pivot Selection**: The last element of the array is chosen as the pivot.

- **Partitioning**: Elements are rearranged such that elements less than the pivot are on the left, and elements greater than the pivot are on the right.
- **Recursive Sorting**: The `quickSort` function is called recursively for the sub-arrays formed by partitioning.

This implementation sorts the array in-place and does not require any additional user-defined functions. It's a straightforward and efficient way to perform Quick Sort in C.