

```
1 //
2 // Created by hfwei on 2023/11/15.
3 //
4
5 #include <stdio.h>
6
7 #define LEN 7
8
9 /**
10  * @brief Sort nums[left .. right] using merge sort.
11  * @param nums
12  * @param left
13  * @param right
14  */
15 void MergeSort(int nums[], int left, int right);
16
17 /**
18  * @brief Merge nums[left .. mid] and nums[mid + 1 .. right]
19  * @param nums
20  * @param left
21  * @param mid
22  * @param right
23  */
24 void Merge(int nums[], int left, int mid, int right);
25
26 /**
27  * @brief Copy src[left .. right] to dest[left .. right]
28  * @param src
29  * @param dest
30  * @param left
31  * @param right
32  */
33 void Copy(const int src[], int dest[], int left, int right);
34
35 int main() {
36     int numbers[LEN] = {38, 27, 43, 3, 9, 82, 10};
37     MergeSort(numbers, 0, LEN - 1);
38
39     for (int i = 0; i < LEN; i++) {
40         printf("%d ", numbers[i]);
41     }
42
43     return 0;
44 }
45
46 void MergeSort(int nums[], int left, int right) {
47     if (left == right) {
48         return;
49     }
50
51     int mid = (left + right) / 2;
52     MergeSort(nums, left, mid);    // Call the Mirror
53     MergeSort(nums, mid + 1, right); // Call the Mirror
```

```
54
55 Merge(nums, left, mid, right);
56 }
57
58 void Merge(int nums[], int left, int mid, int right) {
59     static int copy[LEN] = {0};
60
61     int left_index = left;
62     int right_index = mid + 1;
63     int copy_index = left;
64
65     while (left_index <= mid && right_index <= right) {
66         if (nums[left_index] <= nums[right_index]) {
67             copy[copy_index] = nums[left_index];
68             left_index++;
69         } else {
70             copy[copy_index] = nums[right_index];
71             right_index++;
72         }
73
74         copy_index++;
75     }
76
77     while (left_index <= mid) {
78         copy[copy_index] = nums[left_index];
79         left_index++;
80         copy_index++;
81     }
82
83     while (right_index <= right) {
84         copy[copy_index] = nums[right_index];
85         right_index++;
86         copy_index++;
87     }
88
89     Copy(copy, nums, left, right);
90 }
91
92 void Copy(const int src[], int dest[], int left, int right) {
93     for (int i = left; i <= right; ++i) {
94         dest[i] = src[i];
95     }
96 }
```