

```
1  #include <stdio.h>
2  #include <time.h>
3  void merge_sort(int i, int j, int a[], int aux[]) {
4      if (j <= i) {
5          return;
6      }
7      int mid = (i + j) / 2;
8      merge_sort(i, mid, a, aux);
9      merge_sort(mid + 1, j, a, aux);
10     int pointer_left = i;
11     int pointer_right = mid + 1;
12     int k;
13     for (k = i; k <= j; k++) {
14         if (pointer_left == mid + 1) {
15             aux[k] = a[pointer_right];
16             pointer_right++;
17         } else if (pointer_right == j + 1) {
18             aux[k] = a[pointer_left];
19             pointer_left++;
20         } else if (a[pointer_left] < a[pointer_right]) {
21             aux[k] = a[pointer_left];
22             pointer_left++;
23         } else {
24             aux[k] = a[pointer_right];
25             pointer_right++;
26         }
27     }
28     for (k = i; k <= j; k++) {
29         a[k] = aux[k];
30     }
31 }
32 int main() {
33     int a[100], aux[100], n, i, d, swap;
34     printf("Enter number of elements in the array:\n");
35     scanf("%d", &n);
36     printf("Enter %d integers\n", n);
37     for (i = 0; i < n; i++)
38         scanf("%d", &a[i]);
39     clock_t begin = clock();
40     merge_sort(0, n - 1, a, aux);
41     clock_t end = clock();
```

```
10     int pointer_left = i;
11     int pointer_right = mid + 1;
12     int k;
13     for (k = i; k <= j; k++) {
14         if (pointer_left == mid + 1) {
15             aux[k] = a[pointer_right];
16             pointer_right++;
17         } else if (pointer_right == j + 1) {
18             aux[k] = a[pointer_left];
19             pointer_left++;
20         } else if (a[pointer_left] < a[pointer_right]) {
21             aux[k] = a[pointer_left];
22             pointer_left++;
23         } else {
24             aux[k] = a[pointer_right];
25             pointer_right++;
26         }
27     }
28     for (k = i; k <= j; k++) {
29         a[k] = aux[k];
30     }
31 }
32 int main() {
33     int a[100], aux[100], n, i, d, swap;
34     printf("Enter number of elements in the array:\n");
35     scanf("%d", &n);
36     printf("Enter %d integers\n", n);
37     for (i = 0; i < n; i++)
38         scanf("%d", &a[i]);
39     clock_t begin = clock();
40     merge_sort(0, n - 1, a, aux);
41     clock_t end = clock();
42     printf("Printing the sorted array:\n");
43     for (i = 0; i < n; i++){
44         printf("%d\t", a[i]);
45     }
46     double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
47     printf("\n\nEXECUTION TIME : %.10fseconds\n", time_spent);
48     return 0;
49 }
50
```

"C:\web developement(html.css.js)\merge sort.exe"

Enter number of elements in the array:

6

Enter 6 integers

5 3 4 8 9 2

Printing the sorted array:

2 3 4 5 8 9

EXECUTION TIME : 0.000000000seconds

Process returned 0 (0x0) execution time : 10.849 s

Press any key to continue.

