

```
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    }
```

```

1      aux[k] = a[pointer_left];
2      pointer_left++;
3
4  } else {
5      aux[k] = a[pointer_right];
6      pointer_right++;
7  }
8  }
9
10 for (k = i; k <= j; k++) {
11     a[k] = aux[k];
12 }
13
14 }
15
16 int main() {
17     int a[100], aux[100], n, i, d, swap;
18     printf("Enter number of elements in the array:\n");
19     scanf("%d", &n);
20     printf("Enter %d numbers\n", n);
21     for (i = 0; i < n; i++)
22         scanf("%d", &a[i]);
23     clock_t begin = clock();
24     merge_sort(0, n - 1, a, aux);
25     printf("Printing the sorted array:\n");
26     for (i = 0; i < n; i++){
27         printf("%d\t", a[i]);
28     }
29     clock_t end = clock();
30     double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
31     printf("\nExecution Time : %.10fseconds\n", time_spent);
32     return 0;
33 }

```


Enter number of elements in the array:

5

Enter 5 numbers

20 15 5 10 25

Printing the sorted array:

5 10 15 20 25

Execution Time : 0.0000170000seconds

...Program finished with exit code 0

Press ENTER to exit console.