Merge sort

```c
#include<stdio.h>
#include<time.h>

int arr[20], size;

void merge_arrays(int[], int, int, int);
void merge_sort(int[], int, int);

int main() {
   int i;
   clock_t start_time, end_time;
   double elapsed_time;

   printf("Enter the number of elements: ");
   scanf("%d", &size);

   printf("Enter the elements: ");
   for (i = 0; i < size; i++) {
      scanf("%d", &arr[i]);
   }

   start_time = clock();
   merge_sort(arr, 0, size - 1);
   end_time = clock();

   elapsed_time = (double)(end_time - start_time) / CLOCKS_PER_SEC;

   printf("Sorted array: ");
   for (i = 0; i < size; i++) {
      printf("%d ", arr[i]);
   }
   printf("\n");

   printf("Time taken: %f seconds\n", elapsed_time);

   return 0;
}

void merge_sort(int arr[], int start, int end) {
   if (start < end) {
      int middle = (start + end) / 2;
```

```
        merge_sort(arr, start, middle);
        merge_sort(arr, middle + 1, end);
        merge_arrays(arr, start, middle, end);
    }
}

void merge_arrays(int arr[], int start, int middle, int end) {
    int i = start, j = middle + 1, k = 0;
    int temp[end - start + 1];

    while (i <= middle && j <= end) {
        if (arr[i] < arr[j]) {
            temp[k++] = arr[i++];
        } else {
            temp[k++] = arr[j++];
        }
    }

    while (i <= middle) {
        temp[k++] = arr[i++];
    }

    while (j <= end) {
        temp[k++] = arr[j++];
    }

    for (i = start, k = 0; i <= end; i++, k++) {
        arr[i] = temp[k];
    }
}
```

```
Output
Enter the number of elements: 5
Enter the elements: 12 3 5 7 9
Sorted array: 3 5 7 9 12
Time taken: 0.000001 seconds
```