```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/*******************
This is code for merge sort
*******************/

// Swap two integers, of which, addresses are known
void swap(int *a, int *b){
    int temp = *a;
    *a = *b;
    *b = temp;
}

// Print an array neatly
void print(int *arr, int len){
    printf("[");
    for(int i = 0; i < len; i++){
        printf("%d", arr[i]);
        if(i < len - 1) printf(" ");
    }
    printf("]");
}

// Merge two arrays
// c: index where second array begins
// len: total length of the two arrays
void doMerge(int *arr, int c, int len){
    int ptr1 = 0, ptr2 = c;
    int *cpy = (int*)malloc(len * sizeof(int));
    for(int pos = 0; pos < len; pos++){
        if(ptr1 < c && ptr2 < len){
            if(arr[ptr1] < arr[ptr2])
                cpy[pos] = arr[ptr1++];
            else
                cpy[pos] = arr[ptr2++];
        }else if(ptr1 < c){ // Array 2 exhausted
            cpy[pos] = arr[ptr1++];
        }else if(ptr2 < len){ // Array 1 exhausted
            cpy[pos] = arr[ptr2++];
        }
    }
    for(int pos = 0; pos < len; pos++)
        arr[pos] = cpy[pos];
    free(cpy); // Done with this array - no memory leaks
}

// len: length of the array
void doMergeSort(int *arr, int len){
    if(len > 1){
        int c = (int)ceil(len/2);
        doMergeSort(arr, c);
        doMergeSort(arr + c, len - c);
        printf("Recursive Sort: ");
        print(arr, c);
        print(arr + c, len - c);
        doMerge(arr, c, len);
        printf("\nAfter Merge: ");
        print(arr, len);
        printf("\n");
    }
}

int main(){
    int arr[10] = {8,9,6,3,4,2,1,5,10,7};
    printf("Initial: ");
    print(arr, 10);
    printf("\n");
    doMergeSort(arr, 10);
    return 0;
}
```