```c
#include <stdio.h>

void merge(int arr[], int left, int middle, int right) {
    int i, j, k;
    int n1 = middle - left + 1;
    int n2 = right - middle;

    // Create temporary arrays
    int L[n1], R[n2];

    // Copy data to temporary arrays L[] and R[]
    for (i = 0; i < n1; i++)
        L[i] = arr[left + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[middle + 1 + j];

    // Merge the temporary arrays back into arr[left..right]
    i = 0; // Initial index of first subarray
    j = 0; // Initial index of second subarray
    k = left; // Initial index of merged subarray

    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        } else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    // Copy the remaining elements of L[], if there are any
    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }

    // Copy the remaining elements of R[], if there are any
    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}

void mergeSort(int arr[], int left, int right) {
    if (left < right) {
        // Same as (left + right) / 2, but avoids overflow for large left and
right
        int middle = left + (right - left) / 2;
```

```
        // Sort first and second halves
        mergeSort(arr, left, middle);
        mergeSort(arr, middle + 1, right);

        // Merge the sorted halves
        merge(arr, left, middle, right);
    }
}

int main() {
    int n;

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

    int arr[n];

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

    printf("Given array is \n");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");

    // Perform merge sort
    mergeSort(arr, 0, n - 1);

    printf("Sorted array is \n");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");

    return 0;
}
```