#include <iostream>

using namespace std;

// Function to merge two subarrays

void merge(int arr[], int left, int mid, int right) {

int n1 = mid - left + 1; // Size of the left subarray

int n2 = right - mid; // Size of the right subarray

// Create temporary arrays

int leftArr[n1], rightArr[n2];

// Copy data to temporary arrays

for (int i = 0; i < n1; i++)

leftArr[i] = arr[left + i];

for (int j = 0; j < n2; j++)

rightArr[j] = arr[mid + 1 + j];

// Merge the temporary arrays back into arr

int i = 0, j = 0, k = left;

while (i < n1 && j < n2) {

if (leftArr[i] <= rightArr[j]) {

arr[k] = leftArr[i];

i++;

} else {

arr[k] = rightArr[j];

j++;

}

k++;

}

// Copy any remaining elements of leftArr

while (i < n1) {

arr[k] = leftArr[i];

i++;

k++;

}

// Copy any remaining elements of rightArr

while (j < n2) {

arr[k] = rightArr[j];

j++;

k++;

}

}

// Function to perform merge sort

void mergeSort(int arr[], int left, int right) {

if (left < right) {

// Find the middle point

int mid = left + (right - left) / 2;

// Recursively sort the two halves

mergeSort(arr, left, mid);

mergeSort(arr, mid + 1, right);

// Merge the sorted halves

merge(arr, left, mid, right);

}

}

// Function to print an array

void printArray(int arr[], int size) {

for (int i = 0; i < size; i++)

cout << arr[i] << " ";

cout << endl;

}

// Main function

int main() {

int arr[] = {38, 27, 43, 3, 9, 82, 10};

int size = sizeof(arr) / sizeof(arr[0]);

cout << "Original array: ";

printArray(arr, size);

mergeSort(arr, 0, size - 1);

cout << "Sorted array: ";

printArray(arr, size);

return 0;

}