#include <iostream>

#include <vector>

#include <cstdlib>

#include <chrono>

using namespace std;

// Function to print the elements of an array

void print(const vector<int>& arr) {

    for (int i = 0; i < arr.size(); i++) {

        cout << arr[i] << " ";

    }

    cout << endl;

}

// Merge function used in MergeSort to merge two halves

void merge(vector<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

    vector<int> L(n1), R(n2); // Temporary arrays to hold the two halves

    for (int i = 0; i < n1; i++) {

        L[i] = arr[left + i];

    }

    for (int j = 0; j < n2; j++) {

        R[j] = arr[mid + 1 + j];

    }

    int i = 0, j = 0, k = left; // Initial indexes of subarrays and merged array

    while (i < n1 && j < n2) {

        if (L[i] <= R[j]) {

            arr[k] = L[i];

            i++;

        } else {

            arr[k] = R[j];

            j++;

        }

        k++;

    }

    while (i < n1) {

        arr[k] = L[i];

        i++;

        k++;

    }

    while (j < n2) {

        arr[k] = R[j];

        j++;

        k++;

    }

}

// MergeSort function to sort an array using the merge sort algorithm

void mergeSort(vector<int>& arr, int left, int right) {

    if (left < right) {

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

        mergeSort(arr, left, mid);

        mergeSort(arr, mid + 1, right);

        merge(arr, left, mid, right);

    }

}

int main() {

    ios\_base::sync\_with\_stdio(false);

    vector<int> arr(300);

    for (int i = 0; i < 300; i++) {

        arr[i] = 1 + rand() % 1000;

    }

    cout << "Given array is:" << endl;

    print(arr);

    auto start\_time = chrono::high\_resolution\_clock::now();

    mergeSort(arr, 0, arr.size() - 1);

    auto stop\_time = chrono::high\_resolution\_clock::now();

    chrono::duration<double> elapsed = stop\_time - start\_time;

    cout << "\nSorted array using MergeSort is:" << endl;

    print(arr);

    cout << "\nElapsed time for MergeSort: " << elapsed.count() << " s\n";

    return 0;

}