#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;

}

// Partition function used in QuickSort to partition the array around a pivot

int partition(vector<int>& arr, int low, int high) {

    int pivot = arr[high];

    int i = low - 1;

    for (int j = low; j < high; j++) {

        if (arr[j] < pivot) {

            i++;

            swap(arr[i], arr[j]);

        }

    }

    swap(arr[i + 1], arr[high]);

    return i + 1;

}

// QuickSort function to sort an array using the quick sort algorithm

void quickSort(vector<int>& arr, int low, int high) {

    if (low < high) {

        int pi = partition(arr, low, high);

        quickSort(arr, low, pi - 1);

        quickSort(arr, pi + 1, high);

    }

}

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();

    quickSort(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 QuickSort is:" << endl;

    print(arr);

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

    return 0;

}