#include <iostream>

using namespace std;

// Function to partition the array

int partition(int arr[], int low, int high) {

int pivot = arr[high]; // Choose the last element as pivot

int i = low - 1; // Index for smaller element

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

if (arr[j] < pivot) {

i++;

swap(arr[i], arr[j]); // Move smaller element to the left

}

}

swap(arr[i + 1], arr[high]); // Place the pivot in the correct position

return i + 1; // Return the partition index

}

// Quicksort function

void quickSort(int arr[], int low, int high) {

if (low < high) {

int pi = partition(arr, low, high); // Partition the array

// Recursively sort the left and right subarrays

quickSort(arr, low, pi - 1);

quickSort(arr, pi + 1, high);

}

}

// Utility function to print the array

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

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

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

}

cout << endl;

}

int main() {

int arr[] = {8, 3, 1, 7, 0, 10, 2};

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

cout << "Original array: ";

printArray(arr, n);

quickSort(arr, 0, n - 1);

cout << "Sorted array: ";

printArray(arr, n);

return 0;

}