#include <iostream>

#include <vector>

#include <omp.h>

void parallelBubbleSort(std::vector<int>& arr) {

int n = arr.size();

bool swapped = true;

#pragma omp parallel num\_threads(n)

{ while (swapped) {

swapped = false;

#pragma omp for

for (int i = 0; i < n - 1; i++) {

if (arr[i] > arr[i + 1]) {

#pragma omp critical

std::swap(arr[i], arr[i + 1]);

swapped = true; }}}}}

int main() {

std::vector<int> arr = {5, 2, 8, 12, 1, 6, 3, 9};

parallelBubbleSort(arr);

for (int num : arr)

std::cout << num << " ";

std::cout << std::endl;

return 0;}