#include <iostream>

#include <vector>

#include <omp.h>

using namespace std;

// Merge Sort

void merge(vector<int>& arr, int l, int m, int r) {

    int n1 = m - l + 1;

    int n2 = r - m;

    vector<int> L(n1), R(n2);

    for (int i = 0; i < n1; i++)

        L[i] = arr[l + i];

    for (int j = 0; j < n2; j++)

        R[j] = arr[m + 1 + j];

    int i = 0, j = 0, k = l;

    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++;

    }

}

void mergeSort(vector<int>& arr, int l, int r) {

    if (l < r) {

        int m = l + (r - l) / 2;

        #pragma omp parallel sections

        {

            #pragma omp section

            mergeSort(arr, l, m);

            #pragma omp section

            mergeSort(arr, m + 1, r);

        }

        merge(arr, l, m, r);

    }

}

// Bubble Sort

void bubbleSort(vector<int>& arr) {

    int n = arr.size();

    for (int i = 0; i < n - 1; i++) {

        bool swapped = false;

        #pragma omp parallel for shared(swapped)

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

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

                swap(arr[j], arr[j + 1]);

                swapped = true;

            }

        }

        if (!swapped) break; // If no two elements were swapped, the array is already sorted

    }

}

int main() {

    int n;

    cout << "Enter the number of elements: ";

    cin >> n;

    vector<int> arr(n);

    cout << "Enter " << n << " elements: ";

    for (int i = 0; i < n; i++)

        cin >> arr[i];

    // Merge Sort

    cout << "Merge Sort:";

    mergeSort(arr, 0, n - 1);

    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";

    cout << endl;

    // Bubble Sort

    cout << "Bubble Sort:";

    bubbleSort(arr);

    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";

    cout << endl;

    return 0;

}