#include<iostream>

using namespace std;

/////////////// support functions ///////////////

void fill(int\* arr, int n) {

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

arr[i] = rand() % 10;

}

}

void show(int\* arr, int n) {

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

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

}

cout << endl;

}

/////////////// algorithm of merge sort ///////////////

void toMerge(int\* arr, int begin, int end) {

int mid = begin + (end - begin) / 2,

i = begin,

k = 0,

j = mid + 1;

int\* tempArr = new int[end];

while (i <= mid && j <= end) {

if (arr[i] <= arr[j]) {

tempArr[k++] = arr[i++];

}

else {

tempArr[k++] = arr[j++];

}

}

while (i <= mid) {

tempArr[k++] = arr[i++];

}

while (j <= end) {

tempArr[k++] = arr[j++];

}

for (i = 0; i < k; i++) {

arr[begin + i] = tempArr[i];

}

}

void mergeSort(int\* arr, int left, int right) {

if (left < right)

if (right - left == 1) {

if (arr[right] < arr[left]) {

int temp = arr[left];

arr[left] = arr[right];

arr[right] = temp;

}

}

else {

mergeSort(arr, left, left + (right - left) / 2);

mergeSort(arr, left + (right - left) / 2 + 1, right);

toMerge(arr, left, right);

}

}

/////////////// test function ///////////////

void testSort() {

int n = rand() % 100;

int\* arr = new int[n];

bool cheker = true;

fill(arr, n);

show(arr, n);

mergeSort(arr, 0, n - 1);

show(arr, n);

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

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

cheker = false;

break;

}

}

cheker ? cout << "test passed" << endl << endl

: cout << "test failed" << endl << endl;

}

/////////////// main function ///////////////

void main() {

testSort();

testSort();

testSort();

}