using System;

using System.Collections.Generic;

using System.IO;

class Solution

{

private static void copy(List<int> list, int[] array, int startIndex)

{

foreach (int num in list)

{

array[startIndex++] = num;

}

}

static int partition(int[] ar, int start, int end)

{

int pivote = ar[start];

List<int> left = new List<int>();

List<int> equal = new List<int>();

List<int> rigth = new List<int>();

equal.Add(pivote);

for (int i = start + 1; i <= end; i++)

{

if (ar[i] < pivote)

{

left.Add(ar[i]);

}

else

{

rigth.Add(ar[i]);

}

}

copy(left, ar, start);

int newPivotIndex = start + left.Count;

ar[newPivotIndex] = pivote;

copy(rigth, ar, newPivotIndex + 1);

return newPivotIndex;

}

static void printArray(int[] ar, int start, int end)

{

if (start < end)

{

for (int i = start; i <= end; i++)

{

Console.Write(ar[i] + " ");

}

Console.WriteLine();

}

}

private static void quickSort(int[] array, int start, int end)

{

if (start < end)

{

int pivotIndex = partition(array, start, end);

quickSort(array, start, pivotIndex - 1);

quickSort(array, pivotIndex + 1, end);

printArray(array, start, end);

}

}

/\* Tail starts here \*/

static void Main(String[] args)

{

int \_ar\_size;

\_ar\_size = Convert.ToInt32(Console.ReadLine());

int[] \_ar = new int[\_ar\_size];

String elements = Console.ReadLine();

String[] split\_elements = elements.Split(' ');

for (int \_ar\_i = 0; \_ar\_i < \_ar\_size; \_ar\_i++)

{

\_ar[\_ar\_i] = Convert.ToInt32(split\_elements[\_ar\_i]);

}

quickSort(\_ar, 0, \_ar.Length - 1);

}

}