using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Parallel\_Task

{

internal class Program

{

static void Main(string[] args)

{

int n = Convert.ToInt32(Console.ReadLine());

Func<object, int[]> func1 = new Func<object, int[]>(GetArray);

Task<int[]> task1 = new Task<int[]>(func1, n);

Func<Task<int[]>, int[]> func2 = new Func<Task<int[]>, int[]>(SortArray);

Task<int[]> task2 = task1.ContinueWith<int[]>(func2);

Action<Task<int[]>> action = new Action<Task<int[]>>(PrintArray);

Task task3 = task2.ContinueWith(action);

task1.Start();

Console.ReadKey();

}

static int[] GetArray(object a)

{

int n = (int)a;

int[] array = new int[n];

Random random = new Random();

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

{

array[i] = random.Next(0, 100);

}

return array;

}

static int[] SortArray(Task<int[]> task)

{

int[] array = task.Result;

for (int i = 0; i < array.Count() - 1; i++)

{

for (int j = i + 1; j < array.Count(); j++)

{

if (array[i] > array[j])

{

int t = array[i];

array[i] = array[j];

array[j] = t;

}

}

}

return array;

}

static void PrintArray(Task<int[]> task)

{

int[] array = task.Result;

for (int i = 0; i < array.Count(); i++)

{

Console.Write($"{array[i]} ");

}

}

}

}