DAY 11

import java.util.Scanner;

import java.util.Arrays;

class Sorting {

static Scanner sc = new Scanner(System.in);

static void input(int arr[]) {

System.out.print("Enter the array elements: ");

for(int i = 0; i < arr.length; i++) {

arr[i] = sc.nextInt();

}

}

static void display(int arr[]) {

System.out.print("Array elements are: ");

for(int i : arr) {

System.out.print(i + " ");

}

}

static int partition(int A[], int p, int r) {

int x, i, j, temp;

x = A[r]; //pivot element

i = p - 1;

for(j = p; j < r; j++) {

if(A[j] <= x) {

i++;

temp = A[i];

A[i] = A[j];

A[j] = temp;

}

}

temp = A[i + 1];

A[i + 1] = A[r];

A[r] = temp;

return i + 1;

}

static void quickSort(int arr[], int p, int r) {

int q;

if(p < r) {

q = partition(arr, p, r);

quickSort(arr, p, q - 1);

quickSort(arr, q + 1, r);

}

}

static void merge(int A[], int p, int q, int r) {

int n1, n2, i, j, k, L[], R[];

n1 = q - p + 1;

n2 = r - q;

L = new int[n1 + 1];

R = new int[n2 + 1];

for(i = 0; i < n1; i++) {

L[i] = A[p + i];

}

for(j = 0; j < n2; j++) {

R[j] = A[q + 1 + j];

}

L[n1] = R[n2] = 999999;

i = j = 0;

for(k = p; k <= r; k++) {

if(L[i] < R[j]) {

A[k] = L[i++];

}

else {

A[k] = R[j++];

}

}

}

static void mergeSort(int arr[], int p, int r) {

int q;

if(p < r) {

q = (p + r) / 2;

mergeSort(arr, p, q);

mergeSort(arr, q + 1, r);

merge(arr, p, q, r);

}

}

public static void main(String[] args) {

int array[], n;

System.out.print("Enter the array size: ");

n = sc.nextInt();

array = new int[n];

input(array);

//quickSort(array, 0, n - 1);

//Arrays.sort(array);

mergeSort(array, 0, n - 1);

display(array);

}

}