public class MergeSortTest {

private void mergeSort (int [] array, int left, int right){

if (left < right){

//Set Middle

int middle = (left + right) / 2;

//Recursively sort halves of the array

mergeSort(array, left, middle);

mergeSort(array, middle + 1, right);

//Finally merge after all sort's are returned

merge(array, left, middle, right);

}

}

private void merge (int [] array, int left, int middle, int right){

int sub1 = middle - left + 1;

int sub2 = right - middle;

int [] smallLeft = new int [sub1];

int [] smallRight = new int[sub2];

for (int i = 0; i < sub1; ++i)

smallLeft[i] = array[left + i];

for (int j = 0; j < sub2; ++j)

smallRight[j] = array[middle + 1 + j];

int i = 0, j = 0;

int k = left;

while (i < sub1 && j < sub2){

if (smallLeft[i] <= smallRight[j]){

array[k] = smallLeft[i];

i++;

}

else{

array[k] = smallRight[j];

j++;

}

k++;

}

while (i < sub1){

array[k] = smallLeft[i];

i++;

k++;

}

while (j < sub2){

array[k] = smallRight[j];

j++;

k++;

}

}

//Method to print entire array

private void printArray (int [] array){

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

System.out.print(array[i] + " ");

}

}

public static void main(String[] args) {

//Create example array

int [] array = {7, 45, 88, 12, 15, 6, 1, 67, 81, 99};

//Create instance because none of the methods are static

MergeSortTest test = new MergeSortTest();

//Print unsorted array first

System.out.println("Before MergedSort: ");

test.printArray(array);

test.mergeSort(array, 0, array.length - 1);

//Finally Print sorted array

System.out.println("\nAfter MergedSort: ");

test.printArray(array);

}

}