WAP for merge sort using recursion

**import** java.util.Scanner;

**public** **class** MergeSort {

**public** **static** **int**[] takeInput() {

Scanner s = **new** Scanner(System.***in***);

**int** size = s.nextInt();

**int** arr[] = **new** **int**[size];

**for** (**int** i = 0; i < size; i++) {

arr[i] = s.nextInt();

}

**return** arr;

}

**public** **static** **void** printArray(**int** input[]) {

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

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

}

}

**public** **static** **void** main(String[] args) {

**int**[] input = *takeInput*();

*mergeSort*(input);

*printArray*(input);

}

**public** **static** **void** mergeSort(**int**[] input){

// Write your code here

*mergeSort*(input,0,input.length-1);

}

**private** **static** **void** mergeSort(**int**[] input, **int** si, **int** ei) {

// **TODO** Auto-generated method stub

**if**(si>=ei) {

**return** ;

}

**int** mid=(si+ei)/2;

*mergeSort*(input,si,mid);

*mergeSort*(input,mid+1,ei);

*merge*(input,si,mid,ei);

}

**private** **static** **void** merge(**int**[] input, **int** si,**int** mid, **int** ei) {

**int** l = mid - si + 1;

**int** r = ei - mid;

**int** LeftArray[] = **new** **int** [l];

**int** RightArray[] = **new** **int** [r];

**for** (**int** i=0; i<l; ++i)

LeftArray[i] = input[si + i];

**for** (**int** j=0; j<r; ++j)

RightArray[j] = input[mid + 1+ j];

**int** i = 0, j = 0;

**int** k = si;

**while** (i<l&&j<r)

{

**if** (LeftArray[i] <= RightArray[j])

{

input[k] = LeftArray[i];

i++;

}

**else**

{

input[k] = RightArray[j];

j++;

}

k++;

}

**while** (i<l)

{ input[k] = LeftArray[i];

i++;

k++;

}

**while** (j<r)

{

input[k] = RightArray[j];

j++;

k++;

}

}

}