**package** com;

**import** java.util.Arrays;

**class** MergeSort

{

**void** merge(**int** nums[], **int** left, **int** m, **int** right)

{

**int** n1 = m - left + 1;

**int** n2 = right - m;

**int** Left\_part\_arra[] = **new** **int** [n1];

**int** Right\_part\_arra[] = **new** **int** [n2];

**for** (**int** i=0; i<n1; ++i)

Left\_part\_arra[i] = nums[left + i];

**for** (**int** j=0; j<n2; ++j)

Right\_part\_arra[j] = nums[m + 1+ j];

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

**int** k = left;

**while** (i < n1 && j < n2)

{

**if** (Left\_part\_arra[i] <= Right\_part\_arra[j])

{

nums[k] = Left\_part\_arra[i];

i++;

}

**else**

{

nums[k] = Right\_part\_arra[j];

j++;

}

k++;

}

**while** (i < n1)

{

nums[k] = Left\_part\_arra[i];

i++;

k++;

}

**while** (j < n2)

{

nums[k] = Right\_part\_arra[j];

j++;

k++;

}

}

// merge()

**void** sort(**int** numbers[], **int** left, **int** right)

{

**if** (left < right)

{

// Find the middle point

**int** m = (left+right)/2;

// Sort first halve

sort(numbers, left, m);

// Sort second halve

sort(numbers , m+1, right);

// Merge the sorted halves

merge(numbers, left, m, right);

}

}

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

{

MergeSort ob = **new** MergeSort();

**int** numbers[] = {4,-9,14,25,52,89,32};

System.***out***.println("Original Array:");

System.***out***.println(Arrays.*toString*(numbers));

ob.sort(numbers, 0, numbers.length-1);

System.***out***.println("Sorted Array:");

System.***out***.println(Arrays.*toString*(numbers));

}

}

