

```

import java.util.ArrayList;
import java.util.Scanner;

public class MergeSort {

    ArrayList<Integer> getInput(ArrayList<Integer> al) {
        al = new ArrayList<Integer>();
        Scanner x = new Scanner(System.in); //creating scanner class to get
user input
        System.out.println("Enter size of arraylist:");
        int size= x.nextInt();
        for (int i = 1; i <= size; i++) //loop for creating array upto size of
arraylist
        {
            System.out.println("Input element for arraylist:");
            int element= x.nextInt();
            al.add(element); //add element in al

        }
        System.out.println("Unsorted arraylist:"); //print unsorted arraylist
        System.out.println(al);
        return al;
    }

    void getOutput(ArrayList<Integer> al) {
        System.out.println("Sorted arraylist:" + al);
    }

    void merge(ArrayList<Integer> al, ArrayList<Integer> beg,
ArrayList<Integer> end) {
        int begsize = beg.size(); //beg =begining side
        int endsize = end.size(); //end= end side
        int i = 0, j = 0, k = 0;
        //running the loop until elements of one side is finished
        while (i < begsize && j < endsize) {
            if (beg.get(i) <= end.get(j)) { //checking which one side is
smaller
                al.set(k, beg.get(i)); //adding the smaller one first
                i++;
            } else {
                al.set(k, end.get(j));
                j++;
            }
            k++;
        }
        //if the elements are left over in either sides
        while (i < begsize) {
            al.set(k, beg.get(i));
            i++;
            k++;
        }
        while (j < endsize) {
            al.set(k, end.get(j));
            j++;
            k++;
        }
    }
}

```

```

void sort(ArrayList<Integer> al) {
    ArrayList<Integer> beg = new ArrayList<Integer>();
    ArrayList<Integer> end = new ArrayList<Integer>();
    int size = al.size();
    if (size == 1) {
        //when only one element is present in arraylist it is already
sorted
        return;//so return the same element
    }
    int mid = size / 2;//calculating mid of arraylist
    for (int i = 0; i < mid; i++) {
        // loop to sort the arraylist of begining side
        beg.add(al.get(i));
    }
    for (int i = mid; i < size; i++) {
        // loop to sort the arraylist of another side
        end.add(al.get(i));
    }
    sort(beg); //call the sort until it convert into single list
    sort(end); //call the sort in another side until it convert into
single list
    merge(al, beg, end); //merge the divide side or list to create a
sorted list
}

public static void main(String[] args) {
    ArrayList<Integer> al = new ArrayList<>();
    MergeSort ms = new MergeSort();
    al = ms.getInput(al);//call to ask user input
    ms.sort(al);//call to sort and merge
    ms.getOutput(al);//call to get output
}
}

```