

```
1  /* Problem 4--Parallel Bubble Sort
2     This was pretty straightforward if you just kept track of what
3     needs to be swapped. */
4
5  import java.io.*;
6  import java.util.*;
7
8  public class prob4 {
9
10     private static Scanner in;
11     private static PrintWriter out;
12     private static int cs;
13
14     public static void main (String[] args) throws Exception {
15
16         cs = 1;
17         in = new Scanner (new File ("prob4.in"));
18         out = new PrintWriter ("prob4.out");
19         while (true) { //Read in a data case
20             int ct = in.nextInt ();
21             if (ct==0) break;
22             int[] A = new int[ct];
23             for (int i=0; i < ct; i++) A[i] = in.nextInt();
24             Process (A);
25         }
26         in.close ();
27         out.close ();
28     }
29
30     /* This method does the swapping */
31     public static void Process (int[] A) throws Exception {
32
33         int offset = 0; //Some cases begin at 0, some at one
34         out.printf ("Case %d:\r\n",cs++);
35         Print (A);
36         while (!Verify (A)) { //If it's not sorted, start swapping
37             for (int i=offset; i+1 < A.length; i+=2)
38                 if (A[i] > A[i+1]) {int t = A[i]; A[i]=A[i+1]; A[i+1]=t;}
39             Print (A);
40             offset = 1-offset;
41         }
42         out.printf ("\r\n");
43     }
44
45     //This method simply prints the array
46     public static void Print (int[] A) throws Exception {
47
48         for (int i=0; i < A.length; i++) out.printf ("%d ",A[i]);
49         out.printf ("\r\n");
50     }
51
52     //This method checks whether the array is sorted */
53     public static boolean Verify (int[] A) throws Exception {
54
55         for (int i=0; i < A.length-1; i++)
56             if (A[i] > A[i+1]) return false;
57         return true;
58     }
59 }
60
```