

```
1  /* Problem 3--Roman Orders
2     This is a variant of a classic problem, the classic solution of which
3     is to count inversions while simultaneously running a merge sort. */
4
5  import java.io.*;
6  import java.util.*;
7
8  public class prob3 {
9
10     private static Scanner in;
11     private static PrintWriter out;
12     private static int cs;
13     private static int sz;
14     private static int[] A;
15     private static int c;
16
17     public static void main (String[] args) throws Exception {
18
19         in = new Scanner (new File ("prob3.in"));
20         out = new PrintWriter ("prob3.out");
21         cs = 1;
22         while (true) {
23             sz = in.nextInt();
24             if (sz==0) break;
25             ReadIn (); //Read In the Data
26             out.printf ("Case %d: There are %d inversion(s).\r\n\r\n",cs++,
27                         InvCount (A)); //Count inversions
28         }
29         in.close();
30         out.close();
31     }
32
33     //ReadIn reads in the data
34     public static void ReadIn () throws Exception {
35
36         c = in.nextInt();
37         A = new int[sz];
38         for (int i=0; i < sz; i++) A[i] = in.nextInt();
39     }
40
41     //InvCount runs a merge sort while counting inversions
42     public static int InvCount (int[] A) throws Exception {
43
44         if (A.length < 2) return 0; //split list in half
45         int[] B = new int[A.length/2], C = new int[(A.length+1)/2];
46         for (int i=0; i < B.length; i++) B[i] = A[i];
47         for (int i=0; i < C.length; i++) C[i] = A[B.length+i];
48         int ic = InvCount (B); //Count inversions in first half
49         ic += InvCount (C); //Count inversions in second half
50         int i=0, j=0; //Count inversions between lists
51         while (i < B.length && j < C.length)
52             if (B[i] > c*C[j]) { //Everything after me in B
53                 ic += B.length-i; //must also beat C[j]
54                 j++;
55             } else i++;
56         i = 0; j = 0; //Merge Sort for real
57         for (int k=0; k < A.length; k++)
58             if (i==B.length || j < C.length && B[i] > C[j])
59                 A[k] = C[j++];
60             else A[k] = B[i++];
61         return ic;
62     }
63 }
64
```