```
/*************************************************
//  This is an implementation of merge sort in
//  Java.
//
//  It wants a filename followed by a number of elements
//  which must be comparable by a simple < = or >
//  So, it is called like this:
//      java Merge /path/to/file number
//
//  Jordan Thomas
//  February 2007
*************************************************/

import java.io.*;

public class Merge
{

        public static int[] x;
        public static long count = 0;
//      public static int[] randoms;

        //pass in how many numbers and a filename
        public static void main(String[] args)
        {

                Integer temp = Integer.parseInt(args[1]);
                int length = (temp).intValue();
                String file = args[0];
                readFromDisk(file, length);

                System.out.println("******************************");
                System.out.println("Merge Sort of " + length + " numbers:");
                System.out.println("_____________________________________");
                System.out.println("Started sort at  :  " + Time.getDate() + "   ||");
                mergeSort(0, length - 1);       //do the sort
                //make sure the items got sorted
//              for(int i = 0; i < length; i++)
//              {
//                      System.out.println(x[i]);
//              }
                System.out.println("Finished sort at :  " + Time.getDate() + "   ||");
                System.out.println("_____________________________________");
                System.out.println("comparisons:  " + count );
                System.out.println("\n\n");

                //added so user has the option to put the now sorted numbers in a file
                if(args.length >= 3)
                {
                        WriteFile.write(x, length, args[2]);
                }
        }

        //simply load all the ints from a file into our array
        public static void readFromDisk(String filename, int length)
        {
                try
                {
                FileReader fr = new FileReader(filename);//Reads the file name
                BufferedReader br = new BufferedReader(fr);//Checks the file.
                String line = br.readLine();//Reads first line of text.

                x = new int[length];
                int i = 0;
                Integer readFromLine = Integer.parseInt(line);
```

```
                while(line != null)//while there is something on the line.
                {
                        x[i] = readFromLine.intValue();
                        line = br.readLine();
//                      System.out.println(line);
                        i++;
                        readFromLine = Integer.parseInt(line);
                }
                br.close();//Closes the buffer.
                }
                catch(FileNotFoundException fnfe)
                {
                        System.out.println("bad file name");
                }
                catch(IOException ioe)
                {
                        System.out.println("Input / Output Exception found:\n\r"+ioe.getMessage());
                }
                catch(NumberFormatException nfe)
                {
                        //System.out.println("done\n");
                }

        }

        //recursive merge sort algorithm
        //basically, just keeping halving the array until you get to
        //blocks of two.  Sort it and then merge it back up the
        //recursive call tree.
        public static void mergeSort(int a, int b)
        {
                int temp, m;
                if(b == a + 1) // case of two elements
                {
                        if(x[b] < x[a])
                        {
                                //if swapping is needed, do it
                                temp = x[a];
                                x[a] = x[b];
                                x[b] = temp;
                        }
                }
                // otherwise we need to merge each half, over and over.
                else if(b > a + 1)
                {
                        m = (a + b) / 2;
                        mergeSort(a,m);
                        mergeSort(m+1,b);
                        merge(a,m,b);
                }
        }

        //actually merges portions of the array so it will be sorted
        public static void merge(int a, int m, int b)
        {
                int i,j,k;
                int[] y = new int[b-a+1];
                i=a;
                j=m+1;
                k=0;
                while(i <= m && j<=b)
                {
                        count++;
                        if(x[i] <= x[j])
                        {
                                y[k] = x[i];
                                i++;
```

```
                                k++;
                        }
                        else
                        {
                                y[k] = x[j];
                                j++;
                                k++;
                        }
                }
                while(j <= b)
                {
                        y[k] = x[j];
                        j++;
                        k++;
                }
                while(i<=m)
                {
                        y[k] = x[i];
                        i++;
                        k++;
                }
                while(i <=m)
                {
                        y[k] = x[i];
                        i++;
                        k++;
                }
                        j=a;
                for(i=0; i<=b-a; i++)
                {
                        x[j] = y[i];
                        j++;
                }
        }
}
```