```
/*************************************************
//  This is insertion sort implemented in Java.
//  It wants a number of integers and a filename
//  It uses these to load the file into an array
//  and then applies insertion sort to the array.
//
//
//  Jordan Thomas
//  February 2007
*************************************************/
import java.io.*;

public class Insertion
{

        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("Insertion Sort of " + length + " numbers:");
                System.out.println("_____________________________________");
                System.out.println("Started sort at  :  " + Time.getDate() + "   ||");
                insertionSort(length);
                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]);
                }

        }


    // Sorts the specified array of numbers using the insertion
    // sort algorithm.  Insertion sort just goes through the list
    // comparing everything to the item after it and switching them
    // if needed.  It does this n -1 times and then everything is
    // sorted
   public static void insertionSort (int n)
   {
      int temp;
      int position;


      for (int index = 1; index < n; index++)
      {
        count++;
         temp = x[index];
         position = index;

         // shift larger values to the right
         while (position > 0 && x[position-1] > temp )
         {
```

```
           count++;
            x[position] = x[position-1];
            position--;
         }

         x[position] = temp;
      }
        }

        //Load up the array with the numbers in our file
        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");
                }

        }
}
```