```java
package TestingAndComplexity;
public class IntSortSearch
{

    private static int numComps = 0;
    //-------------------------------------------------
    public static int getNumComps()
    {
        return numComps;
    }
    //-------------------------------------------------
    public static int linearSearch(int[] a, int x)
    {
        numComps = 0;
        for (int k = 0; k < a.length; k++)//------Line 1
        {
            numComps++;
            if (a[k] == x)//-----------------------Line 2
            {
                System.out.println("Num Comps = "+numComps);
                return k;//--------------------------Line 3
            }
        }
        System.out.println("Num Comps = "+numComps);
        return -1;//-------------------------------Line 4
    }
    //-------------------------------------------------
    private static int select(int[] a, int k)
    {
        //find the location of the smallest element in the array
        //between position k and the end of the array
        int smallestPos = k;    //-------------------Line 5
        for (int j = k; j < a.length; j++)
        {
            numComps++;
            if (a[j] < a[smallestPos])//----------Line 6
                smallestPos = j;//-----------------Line 7
        }
        return smallestPos;//--------------------Line 8
    }
    //-------------------------------------------------
    public static void selectionSort(int[] a){
        int pos;                //position of the smallest elelemt
        int temp;       //temp variable for swap
        numComps = 0;

        for (int k = 0; k < a.length-1; k++)//---Line 9
        {
            //find the smallest element
            pos = select(a, k);//-----------------Line 10
```

```
51
52              //swap it with k-th element
53              temp = a[k];
54              a[k] = a[pos];
55              a[pos] = temp;
56          }
57          System.out.println("Num Comps = "+numComps);//Line 11
58      }
59      //-------------------------------------------------
60      public static int binarySearch(int[] a, int x)
61      {
62          int lowerLimit = 0;
63          int upperLimit = a.length-1;
64          int middle = 0;
65          numComps = 0;
66          while (lowerLimit <= upperLimit)//-------------Line 12
67          {
68              numComps++;
69              middle = (lowerLimit+upperLimit) / 2;
70              if (a[middle] == x){//---------------------Line 13
71                  System.out.println("Num Comps = "+numComps);
72                  return middle;//-----------------------Line 14
73              }
74              if (a[middle] < x){//----------------------Line 15
75                  lowerLimit = middle+1;//----------------Line 16
76              } else {
77                  upperLimit = middle-1;//----------------Line 17
78              }
79          }
80          System.out.println("Num Comps = "+numComps);
81          return -1;//----------------------------------Line 18
82      }
83      //-------------------------------------------------
84  }
```