1 solutions

```java
import java.util.Scanner;

public class ArrayRecursor2{
  public static void printArray(int array[]) {
    System.out.print("[");
    printArray(array, 0);
    System.out.print("]");
  }
  private static void printArray(int array[], int start) {
    System.out.print(array[start]);
    if(start + 1 < array.length && array[start + 1] != -1) {
      System.out.print(",");
      printArray(array, start+1);
    }
  }

  public static void reverse(int a[]) {
    reverse(a, 0, a.length-1);
  }
  private static void reverse(int a[], int start, int end) {
    if(start >= end)
      return;
    int temp = a[start];
    a[start] = a[end];
    a[end] = temp;
    reverse(a,start+1, end-1);
  }

  public static int occurances(int a[], int x) {
    return occurances(a, x, 0);
  }
  private static int occurances(int a[], int x, int index){
    if(index == a.length)
      return 0;
    if(a[index] == x)
      return 1 + occurances(a, x, index+1);
    else
      return occurances(a,x,index+1);
  }

  public static Boolean palindrome(int a[]) {
    return palindrome(a, 0, a.length-1);
  }
  private static Boolean palindrome(int a[], int start, int end) {
    if(start >= end)
      return true;
    if(a[start] == a[end])
      return true && palindrome(a, start+1, end-1);
    else
      return false;
  }

  public static Boolean isSorted(int a[]) {
    return isSorted(a, 0);
  }
  private static Boolean isSorted(int a[], int i) {
```

```
57      if (i >= a.length - 1)
58        return true;
59      if (a[i] <= a[i+1] )
60        return isSorted(a,i+1);
61      else
62        return false;
63    }
64
65    public static int [] mergeTwo(int a[], int b[]){
66      if( !isSorted(a) || !isSorted(b) )
67        return new int[0];
68      int c[] = new int[a.length+b.length];
69      mergeTwo(a,b,c,0,0,0);
70      return c;
71    }
72    private static void mergeTwo(int a[], int b[], int c[], int indexA, int
       indexB, int indexC){
73      if(indexA == a.length) {
74        if(indexB == b.length)
75          return;
76        c[indexC] = b[indexB];
77        mergeTwo(a,b,c, indexA, indexB + 1, indexC+1);
78      } else if(indexB == b.length || a[indexA] <= b[indexB] ) { //Guaranteed
       indexA != a.length
79        c[indexC] = a[indexA];
80        mergeTwo(a,b,c, indexA+1, indexB, indexC+1);
81      } else {
82        c[indexC] = b[indexB];
83        mergeTwo(a,b,c, indexA, indexB+1, indexC+1);
84      }
85    }
86
87    public static int binarySearch(int a[], int target) {
88      return binarySearch(a, target, 0, a.length);
89    }
90    private static int binarySearch(int a[], int target, int start, int end) {
91      if(start > end)
92        return -1;
93      int mid = (end + start)/2;
94      if(a[mid] == target)
95        return mid;
96      else if(a[mid] > target)
97        return binarySearch(a, target, start, mid-1);
98      else
99        return binarySearch(a, target, mid+1, end);
100   }
101
102   public static void main(String[] args) {
103     int choice = 0,c,r;
104     int arr[] = new int[0];
105     int arr2[], arr3[];
106     Scanner input = new Scanner(System.in);
107
108     do{
109       System.out.println("1) Fill new array");
110       System.out.println("2) Print current array.");
111       System.out.println("3) Reverse current array.");
112       System.out.println("4) Count occurances.");
```

```
113       System.out.println("5) Check if array is palindrome");
114       System.out.println("6) Merge with another sorted array.");
115       System.out.println("7) Search in the sorted array.");
116       System.out.println("8) Quit");
117 
118       choice = input.nextInt();
119       switch(choice) {
120         case 1:
121           System.out.println("What size is your array:");
122           c= input.nextInt();
123           arr = new int[c];
124           System.out.println("Enter your numbers, separated by white spaces:")
    ;
125           for(int j =0; j < c;j++){
126             arr[j] = input.nextInt();
127           }
128           break;
129         case 2:
130           ArrayRecursor2.printArray(arr);
131           System.out.println("");
132         break;
133         case 3:
134           ArrayRecursor2.reverse(arr);
135           ArrayRecursor2.printArray(arr);
136           System.out.println("");
137         break;
138         case 4:
139           System.out.println("Enter a number");
140           c= input.nextInt();
141           System.out.print("The number " + c + " occurs "
142             + ArrayRecursor2.occurances(arr,c) + " times in ");
143           ArrayRecursor2.printArray(arr);
144           System.out.println("");
145         break;
146         case 5:
147           if(ArrayRecursor2.palindrome(arr))
148             System.out.println("palindrome");
149           else
150             System.out.println("Not a palindrome");
151         break;
152         case 6:
153           System.out.println("What size is your array:");
154           c= input.nextInt();
155           arr2 = new int[c];
156           System.out.println("Enter your numbers, separated by white spaces:")
    ;
157           for(int j =0; j < c;j++){
158             arr2[j] = input.nextInt();
159           }
160           arr3 = ArrayRecursor2.mergeTwo(arr, arr2);
161           ArrayRecursor2.printArray(arr3);
162           System.out.println("");
163         break;
164         case 7:
165           System.out.println("Enter a number");
166           c= input.nextInt();
167           r = ArrayRecursor2.binarySearch(arr, c);
168           if(r == -1)
```

```
169             System.out.println(c + " was not found");
170           else
171             System.out.println(c + " was found at index " + r);
172         break;
173         case 8:
174           System.out.println("Bye");
175         break;
176       }
177     }while(choice != 8);
178   }
179 }
```