Q1) Sorting Elements of an Array by Frequency

Solution:

```
package TEST;
public class SortByFrequency {

        public static void main(String[] args) {
            int[] array = { 2, 2, 3, 4, 5, 12, 2, 3, 3, 3, 12, };
            sortByCount(array);
        }
        //Array element's count
        static int elementsCount(int ar[], int n)
        {
            int count = 0;
            for(int i : ar)
            {
                int max = n;
                if(i == max)
                {
                    count++;
                }
                max = i;
            }
            return count;
        }
        static void sortByCount(int ar[])
        {
            int temp = 0;
            for(int i = 0; i < ar.length; i++)
            {
                for(int j = i + 1; j < ar.length; j++)
                {
                    if(elementsCount(ar, ar[i]) < elementsCount(ar, ar[j]))
                    {
                        temp = ar[i];
                         ar[i] = ar[j];
                         ar[j] = temp;
                    }
                }
            }
            printArray(ar);
        }
        static void printArray(int[] ar)
        {
            for(int i : ar)
            {
                System.out.print(i+" ");
            }

        }
}
```

Output: -

```
3 3 3 3 2 2 2 12 12 4 5
```

Q2) Longest consecutive subsequence

Solution:

```
package TEST;
import java.util.Arrays;
public class LongCons {

	public static void main(String[] args) {
		int[] num_array = { 49, 1, 3, 200, 2, 4, 70, 5 };
		Arrays.sort(num_array);
		int k = 1, i;
		for (i = 0; i < num_array.length - 1; i++) {
			if ((num_array[i + 1] - num_array[i]) == 1) {
				k++;
			}
		}
		System.out.println("Original array is: " + Arrays.toString(num_array));
		System.out.println("The array Length is " + num_array.length);
		System.out.println("The longest consecutive subsequence of this array is: " + k);
	}
}
```

Output:-

```
Original array is: [1, 2, 3, 4, 5, 49, 70, 200]
The array Length is 8
The longest consecutive subsequence of this array is: 5
```