

# Chapter 7 Problems Part Three

1. Write a method called `lastIndexOf`.

```
import java.util.Arrays;
public class test {
    public static void main(String[] args) {
        int b = 5;
        int[] arr = new int[] {1,2,3,4,0,6,7,4};
        System.out.println(lastIndexOf(b,arr));
    }
    public static int lastIndexOf(int a, int[] intArray) {
        int result = 0;
        for (int i = 0; i < intArray.length; i++) {
            if (intArray[i] == a) {
                result = i;
            }
            else {
                ;
            }
        }
        if (result!=0) {
            return result;
        }
        else {
            return -1;
        }
    }
}
```

2. `range`

```
import java.util.Arrays;
public class test {
    public static void main(String[] args) {
        int intArray[] = new int[] {10,11,88,2,12,120};
        int range = getMax(intArray) - getMin(intArray) + 1;
        System.out.println(range);
    }
    public static int getMax(int[] intArray) {
        int max = intArray[0];
        for(int i = 1; i < intArray.length; i++) {
            if (intArray[i] > max) {
                max = intArray[i];
            }
        }
        return max;
    }
    public static int getMin(int[] intArray) {
        int min = intArray[0];
        for(int i = 1; i < intArray.length; i++) {
            if (intArray[i] < min) {
                min = intArray[i];
            }
        }
        return min;
    }
}
```

3. `countInRange`

```
import java.util.Arrays;
public class test {
    public static void main(String[] args) {
        int[] array = new int[] {1,2,3,4,5,56,9,2,3,54};
        int maximum = 8;
        int minimum = 0;
        System.out.println(countInRange(array,minimum,maximum));
    }
    public static int countInRange(int[] arr, int min, int max) {
        int counter = 0;
        for (int i = 0; i < arr.length; i++) {
            if ((min < arr[i]) && (max > arr[i])) {
                counter++;
            }
            else {
                ;
            }
        }
    }
}
```

```

    }
    }
    return counter;
}
}

```

#### 4. isSorted

```

import java.util.Arrays;
public class test {
    public boolean isSorted(int[] arr) {
        for (int i = 0; i < arr.length - 1; i++) {
            if (arr[i] > arr[i+1]) {
                return false;
            }
        }
        return true;
    }
}

```

#### 5. mode

```

public static int mode(int[] list)
{
    int[] counts = new int[100];
    for(int i = 0; i < list.length; i++)
    {
        counts[list[i]]++;
    }
    int maxPos = counts[0];
    int maxNumber = 0;
    for(int i = 1; i < counts.length; i++)
    {
        if(counts[i] > maxPos)
        {
            maxPos = counts[i];
            maxNumber = i;
        }
    }
    return maxNumber;
}

```

#### 6. stdev

```

public static double stdev(int[] a)
{
    if(a.length == 0)
    {
        return 0;
        double total = 0;
    }
    for(int i = 0; i < a.length; i++)
    {
        total += a[i];
    }
    double average = total / a.length;
    total = 0;
    for(int i = 0; i < a.length; i++)
    {
        total += Math.pow((a[i] - average), 2.0);
    }
    double sdv = Math.sqrt(total / (a.length - 1));
    return sdv;
}

```

#### 8. Write a method called median.

```

public static int median(int[] arr) {
    Arrays.sort(arr);
    return arr[arr.length/2];
}

```

#### 10. percentEven

```
public static int minGap(int[] arr){  
    int placeholder = arr[1] - arr[0];  
    for(int i = 0; i < arr.length; i++) {  
        if (arr[i+1] - arr[i] < placeholder) {  
            placeholder = arr[i+1] - arr[i];  
        }  
    }  
    return placeholder;  
}
```