```java
/**
 * header
 */
import java.lang.System;
import java.lang.Math;

public class ArrayFunHouseTwo
{
    /*
    goingUp() will return true if all numbers in numArray are in increasing order
    [1,2,6,9,23]
    @param numArray is an array of integers
    @return true if the values are in increasing order, false if they are not
     */
    public static boolean goingUp(int[] numArray)
    {
        for(int i = 1; i < numArray.length; i++)
            if( numArray[i] < numArray[i - 1] )
                return false;
        return true;
    }

    /*
    goingDown() will return true if all numbers in numArray are in decreasing order
    @param numArray is an array of integers
    @return true if the values are in decreasing order, false if they are not
     */
    public static boolean goingDown(int[] numArray)
    {
        for(int i = 1; i < numArray.length; i++)
            if( numArray[i] > numArray[i - 1] )
                return false;
        return true;
    }

    /*
    getValuesBiggerThanX will return an array that contains count number of values that
    are larger than parameter x
    @param numArray is an array of integers
    @param count is the number of values larger than x that we want to return
    @param x is the comparison value
    @return array of count size that contains integers that are bigger than x from the
    original array
     */
    public static int[] getCountValuesBiggerThanX(int[] numArray, int count, int x)
    {
        int lengthOfArray = 0;
        for(int i = 0; i < numArray.length; i++)
            if (numArray[i] > x) {
                lengthOfArray++;
                if(lengthOfArray == 7)
                    break;//NEVER rely on a break to exit a loop, unless you HAVE to
            }

        int[] response = new int[lengthOfArray];
        int c = 0;

        for(int i = 0; i < numArray.length; i++)
            if (numArray[i] > x) {
                if (c > lengthOfArray - 1)
                    break;//same comment as above, to lazy to retype
                response[c] = numArray[i];
                c++;
            }

        return response;
```

```
64          }
65      }
66
```