38a)

```java
import java.util.Scanner;

public class BitonicSubsequence {

    public static int longestBitonicSubsequence(int[] nums) {
        int n = nums.length;

        int[] LIS = new int[n];
        int[] LDS = new int[n];

        for (int i = 0; i < n; i++) {
            LIS[i] = 1;
            LDS[i] = 1;
        }

        for (int i = 1; i < n; i++) {
            for (int j = 0; j < i; j++) {
                if (nums[j] < nums[i]) {
                    LIS[i] = Math.max(LIS[i], LIS[j] + 1);
                }
            }
        }

        for (int i = n - 2; i >= 0; i--) {
            for (int j = n - 1; j > i; j--) {
                if (nums[j] < nums[i]) {
                    LDS[i] = Math.max(LDS[i], LDS[j] + 1);
                }
            }
        }
        int maxLen = 0;
        for (int i = 0; i < n; i++) {
            if (LIS[i] > 1 && LDS[i] > 1) {
                maxLen = Math.max(maxLen, LIS[i] + LDS[i] - 1);
            }
        }

        return maxLen;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Take input from user
        System.out.print("Enter the size of the array: ");
        int size = sc.nextInt();

        int[] userInput = new int[size];
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < size; i++) {
            userInput[i] = sc.nextInt();
        }

        // Output the result
        int result = longestBitonicSubsequence(userInput);
        System.out.println("Length of the longest valid bitonic subsequence: " + result);
    }
}
```

38b)

```
import java.util.*;

public class SubstringCount {

    // Count overlapping occurrences of a substring
    public static int countSubstringOccurrences(String text, String pattern) {
        int count = 0;
        for (int i = 0; i <= text.length() - pattern.length(); i++) {
            if (text.substring(i, i + pattern.length()).equals(pattern)) {
                count++;
            }
        }
        return count;
    }

    public static List<Integer> countMatches(String[] A, String[] Q) {
        List<Integer> result = new ArrayList<>();
        for (String a : A) {
            int total = 0;
            for (String q : Q) {
                total += countSubstringOccurrences(a, q);
            }
            result.add(total);
        }
        return result;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // Input for array A
        System.out.print("Enter number of strings in array A: ");
        int n1 = sc.nextInt();
        String[] A = new String[n1];
        System.out.println("Enter the strings for A:");
        for (int i = 0; i < n1; i++) {
            A[i] = sc.next();
        }

        // Input for array Q
        System.out.print("Enter number of query strings in array Q: ");
        int n2 = sc.nextInt();
        String[] Q = new String[n2];
        System.out.println("Enter the query strings for Q:");
        for (int i = 0; i < n2; i++) {
            Q[i] = sc.next();
        }

        // Compute and print result
        List<Integer> output = countMatches(A, Q);
        System.out.println("Output: " + output);
    }
}
```