```
package access_modifiers;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Longest {
    public static List<Integer> longestIncreasingSubsequence(int[] sequence) {
        int n = sequence.length;
        int[] lis = new int[n];
        Arrays.fill(lis, 1);

        for (int i = 1; i < n; i++) {
            for (int j = 0; j < i; j++) {
                if (sequence[i] > sequence[j] && lis[i] < lis[j] + 1) {
                    lis[i] = lis[j] + 1;
                }
            }
        }

        int maxLength = Arrays.stream(lis).max().getAsInt();
        int maxIndex = 0;
        for (int i = 0; i < n; i++) {
            if (lis[i] == maxLength) {
                maxIndex = i;
```

```
                break;
            }
        }

        List<Integer> subsequence = new ArrayList<>();
        subsequence.add(sequence[maxIndex]);

        for (int i = maxIndex - 1; i >= 0; i--) {
            if (sequence[i] < sequence[maxIndex] && lis[i] == lis[maxIndex] - 1) {
                subsequence.add(0, sequence[i]);
                maxIndex = i;
            }
        }

        return subsequence;
    }

    public static void main(String[] args) {
        int[] sequence = {10,22,9,33,21,50,60,41};
        List<Integer> result = longestIncreasingSubsequence(sequence);
        System.out.println("Longest Increasing Subsequence: " + result);
    }
}
```