

题目描述：

在学校中，N 个小朋友站成一队， 第 i 个小朋友的身高为 height[i]，
第 i 个小朋友可以看到的第一个比自己身高更高的小朋友 j，那么 j 是 i 的好朋友(要求 j > i)。
请重新生成一个列表，对应位置的输出是每个小朋友的好朋友位置，如果没有看到好朋友，请在该位置用 0 代替。
小朋友人数范围是 [0, 40000]。

输入描述：

第一行输入 N，N 表示有 N 个小朋友
第二行输入 N 个小朋友的身高 height[i]，都是整数

输出描述：

输出 N 个小朋友的好朋友的位置

补充说明：

示例 1

输入：

2
100 95

输出：

0 0

说明：

第一个小朋友身高 100，站在队尾位置，向队首看，没有比他身高高的小朋友，所以输出第一个值为 0。
第二个小朋友站在队首，前面也没有比他身高高的小朋友，所以输出第二个值为 0。

示例 2

输入：

8
123 124 125 121 119 122 126 123

输出：

1 2 6 5 5 6 0 0

说明:

123 的好朋友是 1 位置上的 124

124 的好朋友是 2 位置上的 125

125 的好朋友是 6 位置上的 126

以此类推

```
import java.util.Scanner;
import java.util.*;
// 注意类名必须为 Main, 不要有任何 package xxx 信息
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        // 注意 hasNext 和 hasNextLine 的区别
        while (in.hasNextInt()) { // 注意 while 处理多个 case
            int num = in.nextInt();
            int[] heights = new int[num];
            for(int i = 0; i < num; i++){
                heights[i] = in.nextInt();
            }

            Deque<Integer> stack = new LinkedList<>();
            int[] res = new int[num];
            for(int i = num - 1; i >= 0; i--){
                while(!stack.isEmpty() && heights[stack.peek()] <= heights[i]){
                    stack.pop();
                }
                if(stack.isEmpty()){
                    res[i] = 0;
                }else{
                    res[i] = stack.peek();
                }
                stack.push(i);
            }

            StringBuilder sb = new StringBuilder();
            for(int i = 0; i < num; i++){

                sb.append(String.valueOf(res[i]));
                sb.append(" ");
            }
            sb.deleteCharAt(sb.length() - 1);
            System.out.println(sb);
        }
    }
}
```

