```java
package stacksqueues;

//Java program to print next
//greater element using stack

public class _2NGE
{
    static class stack
    {
        int top;
        int items[] = new int[100];

        // Stack functions to be used by printNGE
        void push(int x)
        {
            if (top == 99)
            {
                System.out.println("Stack full");
            }
            else
            {
                items[++top] = x;
            }
        }

        int pop()
        {
            if (top == -1)
            {
                System.out.println("Underflow error");
                return -1;
            }
            else
            {
                int element = items[top];
                top--;
                return element;
            }
        }

        boolean isEmpty()
        {
            return (top == -1) ? true : false;
        }
    }

    /* prints element and NGE pair for
    all elements of arr[] of size n */
    static void printNGE(int arr[], int n)
    {
        int i = 0;
        stack s = new stack();
        s.top = -1;
        int element, next;

        /* push the first element to stack */
        s.push(arr[0]);

        // iterate for rest of the elements
        for (i = 1; i < n; i++)
        {
            next = arr[i];

            if (s.isEmpty() == false)
            {
                // if stack is not empty, then
                // pop an element from stack
                element = s.pop();

                /* If the popped element is smaller than
                next, then a) print the pair b) keep
                popping while elements are smaller and
                stack is not empty */
                while (element < next)
                {
                    System.out.println(element + " --> " + next);
                    if (s.isEmpty() == true)
                        break;
                    element = s.pop();
                }

                /* If element is greater than next, then
                push the element back */
                if (element > next)
                    s.push(element);
            }

            /* push next to stack so that we can find next
            greater for it */
            s.push(next);
        }

        /* After iterating over the loop, the remaining
        elements in stack do not have the next greater
        element, so print -1 for them */
        while (s.isEmpty() == false)
        {
            element = s.pop();
            next = -1;
            System.out.println(element + " -- " + next);
        }
    }

    public static void main(String[] args)
    {
        int arr[] = { 13, 11,9,8, 21, 3 };
        int n = arr.length;
        printNGE(arr, n);
    }
}

// Thanks to Rishabh Mahrsee for contributing this code
```