```
1
     package StackAndQueue;
2
3
     class ImplementStackWithLinkedList<E> {
4
5
         private Node<E> peekNode;
6
7
         public ImplementStackWithLinkedList() {
8
             peekNode = null;
9
10
11
         private class Node<E>{
12
             E val;
13
             Node<E> next;
14
             Node (E val) {
15
                  this.val = val;
16
                  this.next = null;
17
             }
18
         }
19
20
         public void push(E val) {
21
22
             Node<E> newPeekNode = new Node<E>(val);
23
             newPeekNode.next = peekNode;
24
             peekNode = newPeekNode;
25
26
27
         public E pop() {
28
29
             if ( peekNode == null ) {
30
                  return null;
31
             } else {
32
33
                  E val = peekNode.val;
34
                  peekNode = peekNode.next;
35
                  return val;
36
             }
37
         }
38
39
         public E peek() {
40
41
             if (peekNode == null) {
42
                  return null;
43
             } else {
44
                  return peekNode.val;
45
             }
46
         }
47
48
49
         public boolean isEmpty() {
50
             return peekNode == null;
51
52
53
         public static void main(String[] args) {
54
55
             ImplementStackWithLinkedList<Integer> stack
56
                      = new ImplementStackWithLinkedList<>();
57
58
             Integer result1 = stack.pop();
59
             Integer result2 = stack.peek();
60
61
             for (int i = 0; i < 9; i++) {
62
                  stack.push(i);
63
             }
64
65
             for (int i = 0; i < 10; i++) {
66
                  System.out.println(stack.pop());
67
             }
68
         }
69
     }
```