```java
package stacksqueues;

/* Java Program to implement a stack using
two queue */
import java.util.*;

class GfG {

    static class Stack {
        // Two inbuilt queues
        static Queue<Integer> q1 = new LinkedList<Integer>();
        static Queue<Integer> q2 = new LinkedList<Integer>();

        // To maintain current number of
        // elements
        static int curr_size;

        Stack()
        {
            curr_size = 0;
        }

        static void push(int x)
        {
            curr_size++;

            // Push x first in empty q2
            q2.add(x);

            // Push all the remaining
            // elements in q1 to q2.
            while (!q1.isEmpty()) {
                q2.add(q1.peek());
                q1.remove();
            }

            // swap the names of two queues
            Queue<Integer> q = q1;
            q1 = q2;
            q2 = q;
        }

        static void pop()
        {

            // if no elements are there in q1
            if (q1.isEmpty())
                return;
            q1.remove();
            curr_size--;
        }

        static int top()
        {
            if (q1.isEmpty())
                return -1;
            return q1.peek();
        }

        static int size()
        {
            return curr_size;
        }
    }

    // driver code
```

```
    public static void main(String[] args)
    {
        Stack s = new Stack();
        s.push(1);
        s.push(2);
        s.push(3);

        System.out.println("current size: " + s.size());
        System.out.println(s.top());
        s.pop();
        System.out.println(s.top());
        s.pop();
        System.out.println(s.top());

        System.out.println("current size: " + s.size());
    }
}
// This code is contributed by Prerna
```