```java
package stacksqueues;

// Java program to implement Queue using
// two stacks with costly enQueue()
import java.util.*;

class GFG
{
    static class Queue
    {
        static Stack<Integer> s1 = new Stack<Integer>();
        static Stack<Integer> s2 = new Stack<Integer>();

        static void enQueue(int x)
        {
            // Move all elements from s1 to s2
            while (!s1.isEmpty())
            {
                s2.push(s1.pop());
                //s1.pop();
            }

            // Push item into s1
            s1.push(x);

            // Push everything back to s1
            while (!s2.isEmpty())
            {
                s1.push(s2.pop());
                //s2.pop();
            }
        }

        // Dequeue an item from the queue
        static int deQueue()
        {
            // if first stack is empty
            if (s1.isEmpty())
            {
                System.out.println("Q is Empty");
                System.exit(0);
            }

            // Return top of s1
            int x = s1.peek();
            s1.pop();
            return x;
        }
    };

    // Driver code
    public static void main(String[] args)
    {
        Queue q = new Queue();
        q.enQueue(1);
        q.enQueue(2);
        q.enQueue(3);

        System.out.println(q.deQueue());
        System.out.println(q.deQueue());
        System.out.println(q.deQueue());
    }
}

// This code is contributed by Prerna Saini
```