Imp. Stack By use of 2 QUEUE

```
import java.util.*;

class Q {

    Stack<Integer> in = new Stack<>();

    Stack<Integer> out = new Stack<>();


    void add(int x) {

        in.push(x);}


    int remove() {

        shift();

        return out.pop();    }


    int peek() {

        shift();

        return out.peek();}


    boolean isEmpty() {

        return in.isEmpty() && out.isEmpty();}


    void shift() {

        if (out.isEmpty())

            while (!in.isEmpty())

                out.push(in.pop());}}
```

Queue by 2 stack

```
import java.util.*;
class Stack {
    Queue<Integer> q1 = new LinkedList<>();
    Queue<Integer> q2 = new LinkedList<>();

    void push(int x) {
        q2.add(x);
        while (!q1.isEmpty()) {
            q2.add(q1.remove());}

        // swap
        Queue<Integer> temp = q1;
        q1 = q2;
        q2 = temp;}

    int pop() {
        return q1.remove();}

    int peek() {
        return q1.peek();}

    boolean isEmpty() {
        return q1.isEmpty();}}
```