Implementing stack using queue

<https://leetcode.com/problems/implement-stack-using-queues/solution/>

class MyStack

{

Queue<Integer> q1;

Queue<Integer> q2;

/\*\* Initialize your data structure here. \*/

public MyStack()

{

q1=new LinkedList<Integer>();

q1=new LinkedList<Integer>();

}

/\*\* Push element x onto stack. \*/

public void push(int x)

{

q1.add(x);

}

/\*\* Removes the element on top of the stack and returns that element. \*/

public int pop()

{

while(q1.size()>1)

{

q2.add(q1.remove());

}

Queue<Integer> temp=q1;

q1=q2;

q2=temp;

return q1.remove();

}

/\*\* Get the top element. \*/

public int top()

{

return q1.remove();

}

/\*\* Returns whether the stack is empty. \*/

public boolean empty()

{

return q1.isEmpty();

}

}

/\*\*

\* Your MyStack object will be instantiated and called as such:

\* MyStack obj = new MyStack();

\* obj.push(x);

\* int param\_2 = obj.pop();

\* int param\_3 = obj.top();

\* boolean param\_4 = obj.empty();

\*/