```java
public class B59 {

    /*
     * //1. 滑动窗口应当是队列，但为了得到滑动窗口的最大值，队列序可以从两端删除元素，因此使用双端队列。
//2. 对新来的元素k，将其与双端队列中的元素相比较， 前面比k小的，直接移出队列（因为不再可能成为后面滑动窗口的最大值了！
//3. 前面比k大的X，比较两者下标，判断X是否已不在窗口之内，不在了，直接移出队列。队列的第一个元素是当前滑动窗口中的最大值
     */
    public ArrayList<Integer> maxInWindows(int [] num, int size) {
        ArrayList<Integer> arr = new ArrayList<>();
        if(num == null)
            return arr;
        if(num.length < size || size < 0)
            return arr;
        LinkedList<Integer> queue = new LinkedList<>();
        for(int i = 0; i < size - 1; i ++) {
            while (!queue.isEmpty() && num[i] >= num[queue.getLast()]) {
                queue.removeLast(); //这里就保证了存在队列中的数字一定是递减的。
            }
                queue.addLast(i);
        }

        for(int i = size - 1; i < num.length; i ++) {
            while (!queue.isEmpty() && num[i] > num[queue.getLast()]) {
                queue.removeLast();
            }
            queue.addLast(i);
            if(i - queue.getFirst() + 1 > size) {
                queue.removeFirst();
            }
            arr.add(num[queue.getFirst()]);//因为数字是递减的，从而满足条件的第一个就是最大的
        }

        return arr;
    }

    public static void main(String [] args) {
        LinkedList<Integer> l1 = new LinkedList<>();
        l1.addLast(1);
        l1.addLast(6);
```