```
import java.util.Scanner;
import java.util.Collections;
import java.util.PriorityQueue;

public class heaps1 {

    // heap approach to get the K largest element from the array
    public static int get_KLargest(int[] arr, int k) {
        PriorityQueue<Integer> min_heap = new PriorityQueue<>();
        for (int i=0; i<arr.length; ++i) {
            min_heap.add(arr[i]);
            if (min_heap.size() > k) {
                min_heap.poll();
            }
        }
        return min_heap.peek();
    }

    // similarly, we'll find the K smallest elemenst from the array
    private static int get_KSmallest(int[] arr, int k) {
        PriorityQueue<Integer> max_heap = new PriorityQueue<>(Collections.
reverseOrder());
        for (int i=0; i<arr.length; i++) {
            max_heap.add(arr[i]);
            if (max_heap.size() > k) {
                max_heap.poll();
            }
        }
        return max_heap.peek();
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int sx = input.nextInt();
        int[] arr = new int[sx];

        for (int i=0; i<sx; ++i) arr[i] = input.nextInt();
        int K = input.nextInt();

        System.out.println("K Smallest :" + get_KSmallest(arr, K));
        System.out.println("\nK Largest :" + get_KLargest(arr, K));
        input.close();
    }
}
```