```java
//Write a Java Program to reverse the Queue upto Kth Elements//
import java.util.Scanner;

class CircularQueue {
    private int maxSize;
    private int front;
    private int rear;
    private int currentSize;
    private int[] queue;

    public CircularQueue(int size) {
        maxSize = size;
        front = 0;
        rear = -1;
        currentSize = 0;
        queue = new int[maxSize];
    }

    public void enqueue(int item) {
        if (isFull()) {
            System.out.println("Queue is full. Overflow condition!");
            return;
        }

        rear = (rear + 1) % maxSize;
        queue[rear] = item;
        currentSize++;

        System.out.println("Enqueued: " + item);
    }

    public int dequeue() {
        if (isEmpty()) {
            System.out.println("Queue is empty. Underflow condition!");
            return -1;
        }

        int item = queue[front];
        front = (front + 1) % maxSize;
        currentSize--;

        return item;
    }

    public boolean isEmpty() {
        return (currentSize == 0);
    }

    public boolean isFull() {
        return (currentSize == maxSize);
    }

    public void display() {
        if (isEmpty()) {
            System.out.println("Queue is empty.");
            return;
        }

        System.out.print("Queue: ");
        int i = front;
        while (i != rear) {
            System.out.print(queue[i] + " ");
            i = (i + 1) % maxSize;
        }
        System.out.println(queue[rear]);
    }

    public void kReverse(int k) {
        if (isEmpty() || k <= 0 || k > currentSize) {
            System.out.println("Invalid value of K.");
            return;
        }

        int[] temp = new int[k];
        for (int i = 0; i < k; i++) {
            temp[i] = dequeue();
        }

        for (int i = k - 1; i >= 0; i--) {
            enqueue(temp[i]);
        }

        System.out.println("Reversed first " + k + " elements and enqueued them.");
    }
}

class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the size of the circular queue: ");
        int size = scanner.nextInt();
        CircularQueue circularQueue = new CircularQueue(size);

        System.out.print("Enter the number of elements to enqueue: ");
        int n = scanner.nextInt();
        System.out.println("Enter the elements to enqueue:");

        for (int i = 0; i < n; i++) {
            int element = scanner.nextInt();
            circularQueue.enqueue(element);
        }

        System.out.print("Enter the value of K: ");
        int k = scanner.nextInt();
        circularQueue.kReverse(k);

        System.out.println("\nFinal Queue:");
        circularQueue.display();

        scanner.close();
    }
}
```