

```

1 //Write a Java Program to reverse the Queue upto Kth Elements//
2 import java.util.Scanner;
3
4 class CircularQueue {
5     private int maxSize;
6     private int front;
7     private int rear;
8     private int currentSize;
9     private int[] queue;
10
11     public CircularQueue(int size) {
12         maxSize = size;
13         front = 0;
14         rear = -1;
15         currentSize = 0;
16         queue = new int[maxSize];
17     }
18
19     public void enqueue(int item) {
20         if (isFull()) {
21             System.out.println("Queue is full. Overflow condition!");
22             return;
23         }
24
25         rear = (rear + 1) % maxSize;
26         queue[rear] = item;
27         currentSize++;
28
29         System.out.println("Enqueued: " + item);
30     }
31
32     public int dequeue() {
33         if (isEmpty()) {
34             System.out.println("Queue is empty. Underflow condition!");
35             return -1;
36         }
37
38         int item = queue[front];
39         front = (front + 1) % maxSize;
40         currentSize--;
41
42         return item;
43     }
44
45     public boolean isEmpty() {
46         return (currentSize == 0);
47     }
48
49     public boolean isFull() {
50         return (currentSize == maxSize);
51     }
52
53     public void display() {
54         if (isEmpty()) {
55             System.out.println("Queue is empty.");
56             return;
57         }
58
59         System.out.print("Queue: ");
60         int i = front;
61         while (i != rear) {
62             System.out.print(queue[i] + " ");
63             i = (i + 1) % maxSize;
64         }
65         System.out.println(queue[rear]);
66     }
67

```

```

68     public void kReverse(int k) {
69         if (isEmpty() || k <= 0 || k > currentSize) {
70             System.out.println("Invalid value of K.");
71             return;
72         }
73
74         int[] temp = new int[k];
75         for (int i = 0; i < k; i++) {
76             temp[i] = dequeue();
77         }
78
79         for (int i = k - 1; i >= 0; i--) {
80             enqueue(temp[i]);
81         }
82
83         System.out.println("Reversed first " + k + " elements and enqueued them.");
84     }
85 }
86
87 class Main {
88     public static void main(String[] args) {
89         Scanner scanner = new Scanner(System.in);
90         System.out.print("Enter the size of the circular queue: ");
91         int size = scanner.nextInt();
92         CircularQueue circularQueue = new CircularQueue(size);
93
94         System.out.print("Enter the number of elements to enqueue: ");
95         int n = scanner.nextInt();
96         System.out.println("Enter the elements to enqueue:");
97
98         for (int i = 0; i < n; i++) {
99             int element = scanner.nextInt();
100             circularQueue.enqueue(element);
101         }
102
103         System.out.print("Enter the value of K: ");
104         int k = scanner.nextInt();
105         circularQueue.kReverse(k);
106
107         System.out.println("\nFinal Queue:");
108         circularQueue.display();
109
110         scanner.close();
111     }
112 }
113

```