1. 에디터

1) 스택으로 구현

```java
public class Editor {
    ///스택을 사용한 버전
    public static void main(String[] args) throws IOException {

        Stack<Character> left = new Stack<>();
        Stack<Character> right = new Stack<>();
        BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
        String s = bf.readLine();
        String rev="";
        int n = Integer.parseInt(bf.readLine());
        for(int i=0;i<s.length();i++) {
            left.push(s.charAt(i));
        }

        for(int i=0;i<n;i++) {
            s = bf.readLine();
            StringTokenizer st=new StringTokenizer(s);
            char order = st.nextToken().charAt(0);
            if(order=='P') { //삽입
                char ch = st.nextToken().charAt(0);

                left.push(ch);
            }
            else if(order=='L') {//커서 왼쪽으로 한칸 옮김
                if(!left.empty())
                right.add(left.pop());
            }
            else if(order=='D') {//커서 오른쪽으로 한칸 옮김
                if(!right.empty())
                left.push(right.pop());
            }
            else if(order=='B') {//커서 왼쪽 문자 하나 삭제
                if(!left.empty())left.pop();
            }

        }
        //마지막으로 출력을 위해 오른쪽에 있는 것을 왼쪽으로 옮김
        while(!left.empty()) {
            right.push(left.pop());
        }
        BufferedWriter bw= new BufferedWriter(new OutputStreamWriter(System.out));
        while(!right.empty() ) {
            bw.write(right.pop());
        }
        bw.flush();
        bw.close();

    }

}
```

2) 리스트로 구현 -> 시간 초과

```
public static void main(String[] args) throws IOException {
    //리스트를 이용한 것. 그러나 시간 초과

    BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
    String s = bf.readLine();
    int curser = s.length();
    String rev="";
    int n = Integer.parseInt(bf.readLine());
    List<Character> lists = new ArrayList<>();
    for(int i=0;i<s.length();i++) {
    lists.add(s.charAt(i));

    }

    for(int i=0;i<n;i++) {
        s = bf.readLine();
        StringTokenizer st=new StringTokenizer(s);
        char order = st.nextToken().charAt(0);
        if(order=='P') { //삽입
            char ch = st.nextToken().charAt(0);
            lists.add(curser,ch);
            curser++;
        }
        else if(order=='L') {//커서 왼쪽으로 한칸 옮김
            if(curser!=0) curser--;
        }
        else if(order=='D') {//커서 오른쪽으로 한칸 옮김
            if(curser!=s.length()+1) curser++;
        }
        else if(order=='B') {//커서 왼쪽 문자 하나 삭제
            if(lists.size()==0) continue;
            if(curser!=0) lists.remove(curser-1);
            curser--;
            if(curser<=0) curser=0;
        }


    }

    Iterator<Character> iter = lists.iterator();
    BufferedWriter bw= new BufferedWriter(new OutputStreamWriter(System.out));
    while(iter.hasNext()) {
        bw.write(iter.next());
    }
    bw.flush();
    bw.close();
}
```

2. 달팽이 리스트

```java
public class Snail {

    static class Node{
        int point; //가르키는 다음 인덱스
        int value; //노드의 값
        Node(int point,  int value){
            this.point = point;

            this.value = value;
        }
    }

    public static void main(String[] args) throws IOException {


        List<Node> snails = new ArrayList<>();
        BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
        String s =bf.readLine();
        StringTokenizer st =new StringTokenizer(s);
        int N =Integer.parseInt(st.nextToken());
        int M = Integer.parseInt(st.nextToken());
        int v = Integer.parseInt(st.nextToken());
        s= bf.readLine();
        st = new StringTokenizer(s);
        for(int i=0;i<N-1;i++) {
        int value = Integer.parseInt(st.nextToken());
            snails.add(new Node(i+1,value)); //마지막노드 N-1전까지 다음노드를 가리킴
        }

        snails.add(new Node(v-1,Integer.parseInt(st.nextToken()))); //마지막 노드가 v-1을 가리킴
        for(int i=0;i<M;i++) {

            int k = Integer.parseInt(bf.readLine());

            if(k>N-1) { //k가 끝노드보다 클때
                k=(k-(v-1))%(N-(v-1))+(v-1); // 바퀴수(k-(v-1) % (N-(v-1))) + 인덱스(v-1)
            }

            System.out.println(snails.get(k).value); //인덱스일때의 값 출력


        }


    }

}
```