

초급자를 골랐지만 팀선정문제를 풀고 스터디를 진행하면서 숙련자 문제를 풀었습니다.

Malloc은 문제 이해를 못하여 문제풀이를 실패하였고, 달팽이 리스트, 에디터 문제로 진행하였습니다.

에디터 문제는 스터디를 통해 해결책을 Iterator라는 내용을 알게 되어 풀 수 있었습니다. 또 Iterator를 몰랐다면 스택과 같은 자료구조를 이용하여 접근했어도 가능한 것을 알 수 있었습니다.

그리고 이번 스터디로 가장 중요하게 배운 점은 ArrayList와 LinkedList의 차이입니다. ArrayList는 인덱스의 흐름 순서의 삽입, 삭제 시에 LinkedList보다 훨씬 빠른 처리 속도를 보여주고, 중간의 인덱스에 값을 삽입, 삭제시에는 LinkedList의 속도가 훨씬 빠릅니다. 이처럼 상황별로 활용할 수 있는 환경이 따로 있다는 것에 대해서 배울 수 있었습니다.

노드 개수 N, 회전 횟수 M, 바퀴당 회전 수 V

입력: (10) (10) (3)

1 2 3 4 5 6 7 8 9 10
13 2 5 11 7 8 2 4 9 10

결과값: (K) 1 2 3 8 9 10 11 16 17 49

일단, 사이클이 V의 위치를 포함하기 때문에 V-1을 해준다.

for (i=1 ~ i<=M i++), 10번 반복 (M번)

첫 N의 사이클은 N번 회전하기 때문에
if (K < N) 때는 저장중인 리스트의 K값 인덱스를 그대로 쓴다.
즉, list.get(K)

그리고 if (K >= N) 이라면 N을 넘어가기 때문에 사이클이
진행된다. 그렇다면 N의 첫 번째 N만큼은 K에서 빼주어야 (K-N)
이걸 사이클의 잔여인 (N-V)로 나눈 그 나머지에서 앞에 채워주
못한 V만큼을 채워주기 위해 +V를 해준다.
즉 list.get((K-N) % (N-V) + V) 이러한 수식이 가능하다.

아래 list를 ArrayList가 아닌 LinkedList를 사용하면
시간 초과가 나며 이는 왜냐하면 부러 삽입, 검색하는 것은 ArrayList,
중간에 삽입, 검색은 하는 것은 LinkedList의 효율이 좋지 때문인 것 같다

```

package day0819;

import java.io.BufferedReader;

public class SWEA_달팽이_리스트 {

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

        ArrayList<Integer> list = new ArrayList<>();
        int N, M, V;

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());

        N = Integer.parseInt(st.nextToken());
        M = Integer.parseInt(st.nextToken());
        V = Integer.parseInt(st.nextToken());

        st = new StringTokenizer(br.readLine());

        for(int i=0; i<N; i++) {
            list.add(i, Integer.parseInt(st.nextToken()));
        }

        V = V-1;
        for(int i=1; i<=M; i++) {
            int K = Integer.parseInt(br.readLine());
            if(K<N) {
                System.out.println(list.get(K));
            }

            if(K>=N) {
                System.out.println(list.get((K-N)%(N-V)+V));
            }
        }
    }
}

```

LinkedList면 시간 초과!

```

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

    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();
}

```

ArrayList 사용

Iterator 사용!