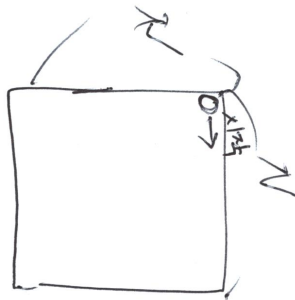


# Baekjoon 3190 뱀

📅 날짜	@2021년 8월 12일 → 2021년 8월 13일
🔗 링크	
📄 열	
🏷 태그	<span>Baekjoon</span> <span>Gold5</span> <span>game</span> <span>simulation</span>

Drawing



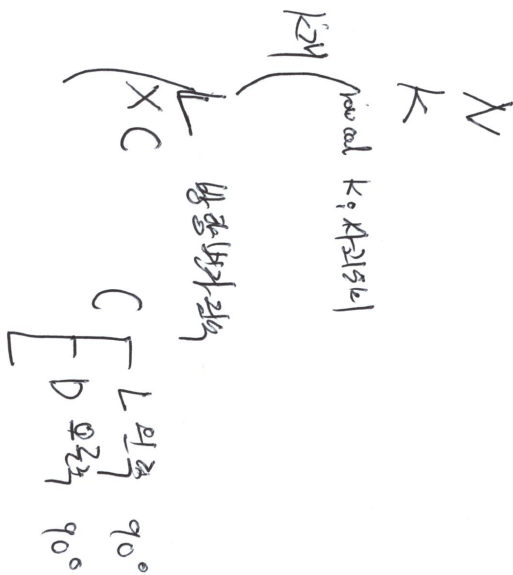
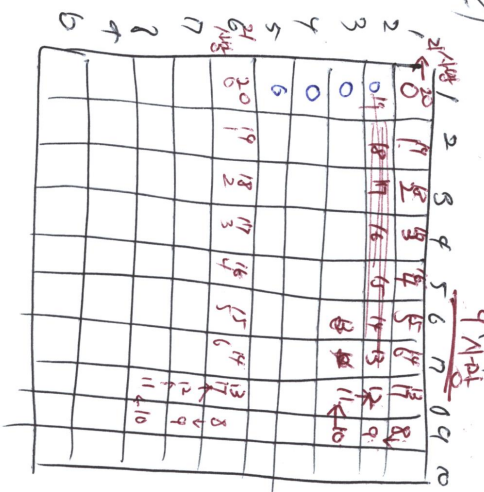
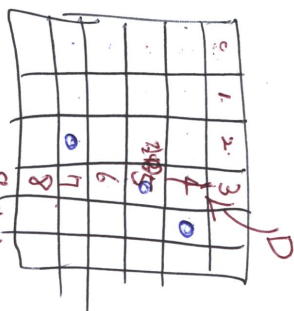
① 머리부터 움직임

② ~~시각~~ 시각이 이동됨  
안정됨

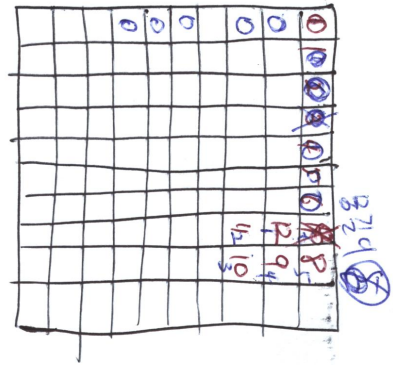
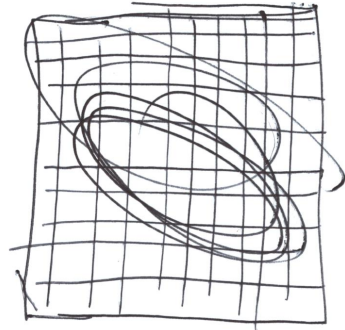
③ K 시각이 가

①

②



③



123 끝난 것  
 75648이 있음  
 132에 있고  
 132에 있음

```
import java.io.BufferedReader;
import java.io.FileInputStream;
```

```

import java.io.IOException;
import java.io.InputStreamReader;
import java.util.LinkedList;
import java.util.StringTokenizer;

public class Baek3190_뱀_1 {
    static boolean[][] map;
    static LinkedList<Snake> s = new LinkedList<>();
    static LinkedList<Apple> a = new LinkedList<>();
    static LinkedList<Command> c = new LinkedList<>();
    static int[] dr = new int[] { 0, 1, 0, -1 }; // 오른쪽으로 가는걸 0으로 시계방향으로 방향 증가
    static int[] dc = new int[] { 1, 0, -1, 0 };
    static int N, K, L, dir;
    static int time = 0;

    public static void main(String[] args) throws NumberFormatException, IOException {
        System.setIn(new FileInputStream("C:/CodingStudy/Baekjoon/Gold5/3190_input"));
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st;
        N = Integer.parseInt(br.readLine());
        K = Integer.parseInt(br.readLine());
        for (int k = 0; k < K; k++) { // 사과 저장
            st = new StringTokenizer(br.readLine());
            a.add(new Apple(Integer.parseInt(st.nextToken()) - 1, Integer.parseInt(st.nextToken()
ken()) - 1)); // 사과 위치는 0 // 기준이 아님
        }

        L = Integer.parseInt(br.readLine());
        for (int l = 0; l < L; l++) { // 명령어 저장
            st = new StringTokenizer(br.readLine());
            c.add(new Command(Integer.parseInt(st.nextToken()), st.nextToken()));
        }
        time = -1;
        s.add(new Snake(0, 0));
        do {
            ++time; // 시간 증가
            if (c.size() != 0 && c.peek().x == time) { // 현재 수행할 명령이 있는가?
                switch (c.poll().com) {
                    case "D": // 오른쪽이면
                        if (++dir >= 4) // 방향 전환
                            dir %= 4;
                        break;
                    case "L": // 왼쪽이면
                        if (--dir < 0) // 방향전환
                            dir = 3;
                        break;
                }
            }
        }
        // showMap();

        } while (!nextState());
        System.out.println(time + 1);

    }

    static void showMap() {
        System.out.println();
        System.out.println("time : " + time);
    }
}

```

```

        for (int row = 0; row < N; row++) {
            for (int col = 0; col < N; col++) {
                boolean who = false;
                for(int i = 0; i<s.size();i++) {
                    if(s.get(i).c == col && s.get(i).r == row)
                        who = true;
                }
                if(who)
                    System.out.print("x ");
                else
                    System.out.print("* ");
            }
            System.out.println();
        }

    }

    static boolean nextState() { // 다음 상황 확인
        int nr = s.get(0).r + dr[dir];
        int nc = s.get(0).c + dc[dir];
        if (nr < 0 || nr >= N || nc < 0 || nc >= N) // 맵을 벗어나면 죽음
            return true;

        int cnt = 0;
        while (++cnt < s.size()) { // 몸이랑 충돌하면 죽음, 머리부터 체크한다.
            if (nr == s.get(cnt).r && nc == s.get(cnt).c)
                return true;
        }

        cnt = -1;
        while (++cnt < a.size()) { // 사과랑 충돌하면 길이가 1 증가함
            if (nr == a.get(cnt).r && nc == a.get(cnt).c) {
                ++(s.get(0).len);
                a.remove(cnt); // !!!!!사과 먹으면 없어지는거 누락했었다...
            }
        }

        s.push(new Snake(nr, nc)); // 뱀을 새로운 위치로 집어 넣고
        if (s.size() > s.get(0).len) // linkedList의 길이가 현재 뱀의 길이보다 길면
            s.pollLast(); // 꼬리부분 제거 (사과먹으면 제거 안함)

        return false; // 아무 이상 없다.
    }

    static class Command {
        int x;
        String com;

        public Command(int x, String com) {
            super();
            this.x = x;
            this.com = com;
        }
    }

    static class Apple {
        int r;

```

```

    int c;

    public Apple(int r, int c) {
        super();
        this.r = r;
        this.c = c;
    }

}

static class Snake {
    int r = 0; // 몸통 하나하나 위치
    int c = 0; // 몸통 하나하나 위치
    static int len = 1; // 길이는 모두가 신경

    public Snake(int r, int c) {
        super();
        this.r = r;
        this.c = c;
    }

}

}

```