

# SWEA 1873

📅 날짜	@2021년 8월 4일
🔗 링크	
📄 열	
🏷 태그	D3 SWEA game 많은 조건 분기 시물

• 평지 전라진입가능

\* 벽들 벽

# 강철 벽

— 물 진입불가

^  
< v > 각 방향 바라보는  
전라

방향 바꾸고  
평지면 이동



바라보는 방향으로

(S) 공격

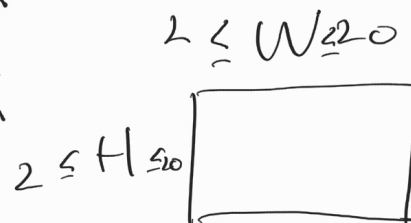
이동 방향이 맨 밖이면 이동 안함

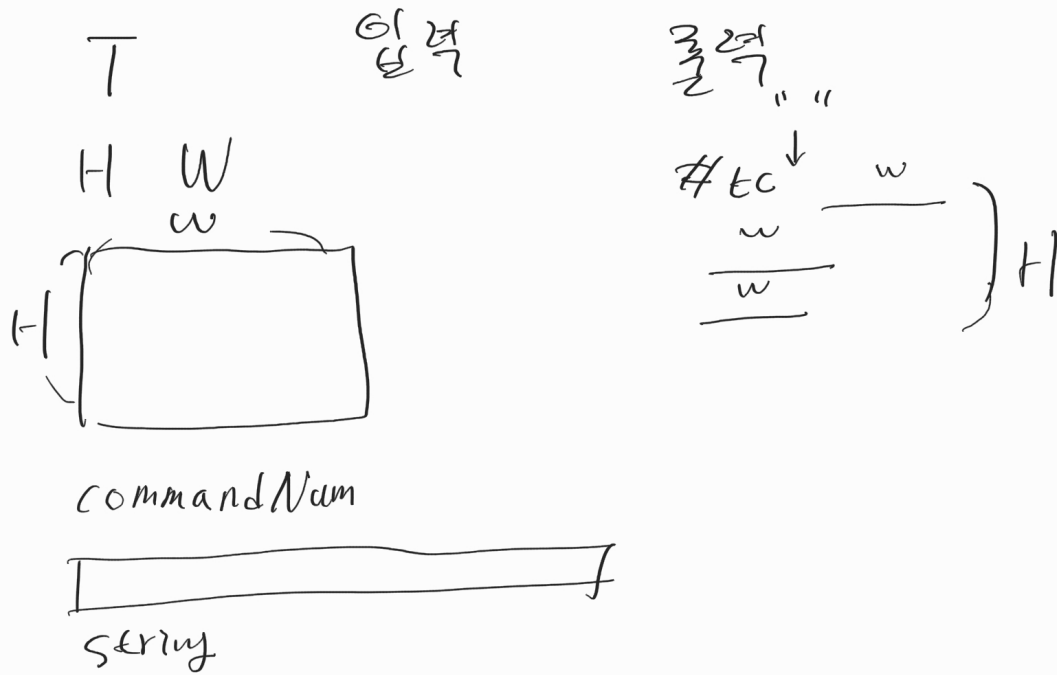
포탄은 직진 벽에 부딪히거나 맨 나감

벽은 벽  $\Rightarrow$  평지

강철벽  $\Rightarrow$  X

맨 밖  $\Rightarrow$  X





① tc 반복문

② 입력 읽기 (이때 행크기 기록)

map, commandLine

③ commandLine 을 입력 받고

명령어 수행

이동 E

상하

좌우

스

③-1 여러 가능하면 이동

max out of index

범위

3-2) 대포 역할 수랭 < 맵 out of index  
범

map, x, y, dir 공유받은 내용

1) 키맨스 종료시

배 클럭

```
package d3;
```

```

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.StringTokenizer;

//public class Solution {
public class Test1873_2 {
    static int H;
    static int W;
    static int[] dx = { 0, 1, 0, -1 };
    static int[] dy = { -1, 0, 1, 0 };

    public static void main(String[] args) throws NumberFormatException, IOException {
// main
        System.setIn(new FileInputStream("C:/CodingStudy/SWEA/D3/Test1873_input.txt"));
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st;
        int T = Integer.valueOf(br.readLine());

        for (int tc = 1; tc <= T; tc++) { // ----- tc 시작
            st = new StringTokenizer(br.readLine());
            H = Integer.valueOf(st.nextToken());
            W = Integer.valueOf(st.nextToken());
            char[][] map = new char[H][W];
            int[] tankPos = new int[3];
//            System.out.println("H:" + H + " " + "W:" + W);
            for (int c = 0; c < H; c++) { // ----- 맵 입력 및 탱크 상태 확인
                String tmpMap = br.readLine();
                for (int r = 0; r < W; r++) {
                    map[c][r] = tmpMap.charAt(r);
                    switch (map[c][r]) {
                        case '^':
                            tankPos[2] = 0;
                            tankPos[0] = c;
                            tankPos[1] = r;
                            map[c][r] = '.';
                            break;
                        case '>':
                            tankPos[2] = 1;
                            tankPos[0] = c;
                            tankPos[1] = r;
                            map[c][r] = '.';
                            break;
                        case 'v':
                            tankPos[2] = 2;
                            tankPos[0] = c;
                            tankPos[1] = r;
                            map[c][r] = '.';
                            break;
                        case '<':
                            tankPos[2] = 3;
                            tankPos[0] = c;
                            tankPos[1] = r;
                            map[c][r] = '.';
                            break;
                    }
                }
            }
        }
    }
}

```

```

    }
} // -----end 맵 입력

int comNum = Integer.valueOf(br.readLine());
String comLine = br.readLine();

for (int comIdx = 0; comIdx < comNum; comIdx++) { // -----command 돌리기

    switch (comLine.charAt(comIdx)) {
    case 'U':
        tankMove(tankPos, 0, map);
        break;
    case 'R':
        tankMove(tankPos, 1, map);
        break;
    case 'D':
        tankMove(tankPos, 2, map);
        break;
    case 'L':
        tankMove(tankPos, 3, map);
        break;
    case 'S':
        tankShot(tankPos, map);
        break;
    }

    //      print(tankPos, map);
    //      System.out.println(comLine.charAt(comIdx));

    } // -----command 종료
    System.out.print("#"+tc+" ");
    print(tankPos, map);

} // -----end tc

} // end main

public static void print(int[] pos, char[][] map) { // 확인하기 위해서 함수로 바꿈
    for (int c = 0; c < H; c++) { // ----- 맵 출력 탱크 추가 해야함 //
        추가 했음
        for (int r = 0; r < W; r++) {
            if(c==pos[0] && r==pos[1]) {
                switch (pos[2]) {
                case 0:
                    System.out.print('^');
                    break;
                case 1:
                    System.out.print('>');
                    break;
                case 2:
                    System.out.print('v');
                    break;
                case 3:
                    System.out.print('<');
                    break;
                }
            }
        }
    }
    else

```

```

        System.out.print(map[c][r]);
    }
    System.out.println();
} // -----end 맵 출력

}

public static void tankShot(int[] pos, char[][] map) {
    int sc = (pos[0] );
    int sr = (pos[1]);
    while (true) {
        sc += dy[pos[2]];
        sr += dx[pos[2]];
//        System.out.println("sc:" + sc + " " + "sr:" + sr);
        if ((sc < 0) || (sc >= H) || (sr < 0) || (sr >= W)) { // 벽 나가는 경우
            return;
        } else if (map[sc][sr] == '*') { // 벽돌벽
            map[sc][sr] = '.'; // 파! 괴!
            return;
        } else if (map[sc][sr] == '#') { // 강철벽
            return;
        }
    }
}

}

public static void tankMove(int[] pos, int dir, char[][] map) { //
    pos[2] = dir;
    int newC = (pos[0] + dy[dir]);
    int newR = (pos[1] + dx[dir]);
    if ((newC >= 0) && (newC < H) && (newR >= 0) && (newR < W) && (map[newC][newR] ==
    '.')) {
        pos[0] += dy[dir];
        pos[1] += dx[dir];
    }
    else {

    }
}

}
}

```

오늘 상태도 안 좋았는데, 어렵다고 생각한 문제가 쉽게 풀렸더니 살짝 정신 나간 하루를 보낸 것 같다.

다시 다음 다잡고 열심히 하자.